

SGP.23 Test Specifcation

Version 1.14

05 July 2023

Security Classification: Non-confidential

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is subject to copyright protection. This document is to be used only for the purposes for which it has been supplied and information contained in it must not be disclosed or in any other way made available, in whole or in part, to persons other than those permitted under the security classification without the prior written approval of the Association.

Copyright Notice

Copyright © 2023 GSM Association

Disclaimer

The GSM Association (“Association”) makes no representation, warranty or undertaking (express or implied) with respect to and does not accept any responsibility for, and hereby disclaims liability for the accuracy or completeness or timeliness of the information contained in this document. The information contained in this document may be subject to change without prior notice.

Compliance Notice

The information contain herein is in full compliance with the GSM Association’s antitrust compliance policy.

This Permanent Reference Document is classified by GSMA as an Industry Specification, as such it has been developed and is maintained by GSMA in accordance with the provisions set out in GSMA AA.35 - Procedures for Industry Specifications.

Table of Contents

[1 Introduction 7](#_Toc139446719)

[1.1 Overview 7](#_Toc139446720)

[1.2 Scope 7](#_Toc139446721)

[1.3 Definition of Terms 8](#_Toc139446722)

[1.4 Abbreviations 8](#_Toc139446723)

[1.5 Document Cross-references 9](#_Toc139446724)

[1.6 Conventions 10](#_Toc139446725)

[2 Testing Rules 10](#_Toc139446726)

[2.1 Applicability 10](#_Toc139446727)

[2.1.1 Format of the Optional Features Table 10](#_Toc139446728)

[2.1.2 Format of the Applicability Table 10](#_Toc139446729)

[2.1.3 Applicability and Notations 11](#_Toc139446730)

[2.1.4 Optional Features Table 11](#_Toc139446731)

[2.1.5 Applicability Table 14](#_Toc139446732)

[2.2 General Consideration 32](#_Toc139446733)

[2.2.1 Test Case Definition 32](#_Toc139446734)

[2.2.2 Test Cases Format 32](#_Toc139446735)

[2.2.3 General Rules for eUICC Testing 37](#_Toc139446736)

[2.2.4 General Rules for Device Testing 38](#_Toc139446737)

[2.2.5 Pass Criteria 41](#_Toc139446738)

[2.2.6 Future Study 41](#_Toc139446739)

[2.2.7 General Rules for SM-DP+ Testing 41](#_Toc139446740)

[3 Testing Architecture 41](#_Toc139446741)

[3.1 Testing Scope 41](#_Toc139446742)

[3.2 Testing Execution 43](#_Toc139446743)

[3.2.1 eUICC - Test Environment 44](#_Toc139446744)

[3.2.2 SM-DP+ and SM-DS - Test Environment 44](#_Toc139446745)

[3.2.3 Device/LPAd - Test Environment 46](#_Toc139446746)

[3.2.4 End-to-End Testing 48](#_Toc139446747)

[3.2.5 Integrated eUICC – Test Environment 48](#_Toc139446748)

[4 Interface Compliance Testing 49](#_Toc139446749)

[4.1 General Overview 49](#_Toc139446750)

[4.2 eUICC Interfaces 49](#_Toc139446751)

[4.2.1 ATR and ISD-R Selection 49](#_Toc139446752)

[4.2.2 ES6 (Operator -- eUICC): UpdateMetadata 50](#_Toc139446753)

[4.2.3 ES8+ (SM-DP+ -- eUICC): InitialiseSecureChannel 58](#_Toc139446754)

[4.2.4 ES8+ (SM-DP+ -- eUICC): ConfigureISDP 62](#_Toc139446755)

[4.2.5 ES8+ (SM-DP+ -- eUICC): StoreMetadata 65](#_Toc139446756)

[4.2.6 ES8+ (SM-DP+ -- eUICC): ReplaceSessionKeys 80](#_Toc139446757)

[4.2.7 ES8+ (SM-DP+ -- eUICC): LoadProfileElements 81](#_Toc139446758)

[4.2.8 ES10a (LPA -- eUICC): GetEuiccConfiguredAddresses 91](#_Toc139446759)

[4.2.9 ES10a (LPA -- eUICC): SetDefaultDPAddress 92](#_Toc139446760)

[4.2.10 ES10b (LPA -- eUICC): PrepareDownload 94](#_Toc139446761)

[4.2.11 ES10b (LPA -- eUICC): LoadBoundProfilePackage 101](#_Toc139446762)

[4.2.12 ES10b (LPA -- eUICC): GetEUICCChallenge 111](#_Toc139446763)

[4.2.13 ES10b (LPA -- eUICC): GetEUICCInfo 112](#_Toc139446764)

[4.2.14 ES10b (LPA -- eUICC): ListNotification 121](#_Toc139446765)

[4.2.15 ES10b (LPA -- eUICC): RetrieveNotificationsList 135](#_Toc139446766)

[4.2.16 ES10b (LPA -- eUICC): RemoveNotificationFromList 163](#_Toc139446767)

[4.2.17 ES10b (LPA -- eUICC): LoadCRL 169](#_Toc139446768)

[4.2.18 ES10b (LPA -- eUICC): AuthenticateServer 169](#_Toc139446769)

[4.2.19 ES10b (LPA -- eUICC): CancelSession 191](#_Toc139446770)

[4.2.20 ES10c (LPA -- eUICC): GetProfilesInfo 199](#_Toc139446771)

[4.2.21 ES10c (LPA -- eUICC): EnableProfile 208](#_Toc139446772)

[4.2.22 ES10c (LPA -- eUICC): DisableProfile 238](#_Toc139446773)

[4.2.23 ES10c (LPA -- eUICC): DeleteProfile 266](#_Toc139446774)

[4.2.24 ES10c (LPA -- eUICC): eUICCMemoryReset 278](#_Toc139446775)

[4.2.25 ES10c (LPA -- eUICC): GetEID 285](#_Toc139446776)

[4.2.26 ES10c (LPA -- eUICC): SetNickname 286](#_Toc139446777)

[4.2.27 ES10b (LPA -- eUICC): GetRAT 290](#_Toc139446778)

[4.3 SM-DP+ Interfaces 291](#_Toc139446779)

[4.3.1 ES2+ (Operator -- SM-DP+): DownloadOrder 291](#_Toc139446780)

[4.3.2 ES2+ (Operator -- SM-DP+): ConfirmOrder 294](#_Toc139446781)

[4.3.3 ES2+ (Operator -- SM-DP+): CancelOrder 305](#_Toc139446790)

[4.3.4 ES2+ (Operator -- SM-DP+): ReleaseProfile 311](#_Toc139446791)

[4.3.5 ES2+ (Operator -- SM-DP+): HandleDownloadProgressInfo 311](#_Toc139446792)

[4.3.6 ES2+ (Operator -- SM-DP+): TLS, Mutual Authentication, Server,Session Establishment 311](#_Toc139446793)

[4.3.7 ES8+ (SM-DP+ -- eUICC): InitialiseSecureChannel 311](#_Toc139446794)

[4.3.8 ES8+ (SM-DP+ -- eUICC): ConfigureISDP 311](#_Toc139446795)

[4.3.9 ES8+ (SM-DP+ -- eUICC): StoreMetadata 312](#_Toc139446796)

[4.3.10 ES8+ (SM-DP+ -- eUICC): ReplaceSessionKeys 312](#_Toc139446797)

[4.3.11 ES8+ (SM-DP+ -- eUICC): LoadProfileElements 312](#_Toc139446798)

[4.3.12 ES9+ (LPA -- SM-DP+): InitiateAuthentication 312](#_Toc139446799)

[4.3.13 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage 321](#_Toc139446800)

[4.3.14 ES9+ (LPA -- SM-DP+): AuthenticateClient 360](#_Toc139446801)

[4.3.15 ES9+ (LPA -- SM-DP+): HandleNotification 400](#_Toc139446802)

[4.3.16 ES9+ (LPA -- SM-DP+): CancelSession 418](#_Toc139446803)

[4.3.17 ES9+ (LPA -- SM-DP+): TLS, Server Authentication, Session Establishment 439](#_Toc139446804)

[4.3.18 ES12 (SM-DP+ -- SM-DS): RegisterEvent 440](#_Toc139446805)

[4.3.19 ES12 (SM-DP+ -- SM-DS): DeleteEvent 440](#_Toc139446806)

[4.3.20 ES12 (SM-DP+ -- SM-DS): TLS, Mutual Authentication, Client, Session Establishment 440](#_Toc139446807)

[4.4 LPAd Interfaces 440](#_Toc139446808)

[4.4.1 ES10a (LPA -- eUICC): GetEuiccConfiguredAddresses 440](#_Toc139446809)

[4.4.2 ES10a (LPA -- eUICC): SetDefaultDPAddress 440](#_Toc139446810)

[4.4.3 ES10b (LPA -- eUICC): PrepareDownload 440](#_Toc139446811)

[4.4.4 ES10b (LPA -- eUICC): LoadBoundProfilePackage 440](#_Toc139446812)

[4.4.5 ES10b (LPA -- eUICC): GetEUICCChallenge 441](#_Toc139446813)

[4.4.6 ES10b (LPA -- eUICC): GetEUICCInfo 441](#_Toc139446814)

[4.4.7 ES10b (LPA -- eUICC): ListNotification 441](#_Toc139446815)

[4.4.8 ES10b (LPA -- eUICC): RetrieveNotificationsList 441](#_Toc139446816)

[4.4.9 ES10b (LPA -- eUICC): RemoveNotificationFromList 441](#_Toc139446817)

[4.4.10 ES10b (LPA -- eUICC): LoadCRL 441](#_Toc139446818)

[4.4.11 ES10b (LPA -- eUICC): AuthenticateServer 441](#_Toc139446819)

[4.4.12 ES10b (LPA -- eUICC): CancelSession 441](#_Toc139446820)

[4.4.13 ES10c (LPA -- eUICC): GetProfilesInfo 441](#_Toc139446821)

[4.4.14 ES10c (LPA -- eUICC): EnableProfile 441](#_Toc139446822)

[4.4.15 ES10c (LPA -- eUICC): DisableProfile 441](#_Toc139446823)

[4.4.16 ES10c (LPA -- eUICC): DeleteProfile 441](#_Toc139446824)

[4.4.17 ES10c (LPA -- eUICC): eUICCMemoryReset 441](#_Toc139446825)

[4.4.18 ES10c (LPA -- eUICC): GetEID 441](#_Toc139446826)

[4.4.19 ES10c (LPA -- eUICC): SetNickname 441](#_Toc139446827)

[4.4.20 ES10b (LPA -- eUICC): GetRAT 442](#_Toc139446828)

[4.4.21 ES9+ (LPA -- SM-DP+): InitiateAuthentication 442](#_Toc139446829)

[4.4.22 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage 447](#_Toc139446830)

[4.4.23 ES9+ (LPA -- SM-DP+): AuthenticateClient 455](#_Toc139446831)

[4.4.24 ES9+ (LPA – SM-DP+): HandleNotification 469](#_Toc139446832)

[4.4.25 ES9+ (LPA – SM-DP+): CancelSession 478](#_Toc139446833)

[4.4.26 ES9+ (LPA – SM-DP+): HTTPS 491](#_Toc139446834)

[4.4.27 ES11 (LPA – SM-DS): InitiateAuthentication 495](#_Toc139446835)

[4.4.28 ES11 (LPA – SM-DS): AuthenticateClient 499](#_Toc139446836)

[4.4.29 ES11 (LPA -- SM-DS): HTTPS 508](#_Toc139446837)

[4.5 SM-DS Interfaces 512](#_Toc139446838)

[4.5.1 ES12 (SM-DP+ -- SM-DS): RegisterEvent 512](#_Toc139446839)

[4.5.2 ES12 (SM-DS -- SM-DP+): DeleteEvent 519](#_Toc139446840)

[4.5.3 ES15 (SM-DS -- SM-DS): RegisterEvent 527](#_Toc139446841)

[4.5.4 ES15 (SM-DS -- SM-DS): DeleteEvent 529](#_Toc139446842)

[4.5.5 ES11 (LPA -- SM-DS): InitiateAuthentication 532](#_Toc139446843)

[4.5.6 ES11 (LPA -- SM-DS): Authenticate Client 533](#_Toc139446844)

[4.5.7 ES15 (SM-DS -- SM-DS): TLS, Mutual Authentication, Client, Session Establishment 554](#_Toc139446845)

[4.5.8 ES12 (SM-DS -- SM-DP+): TLS, Mutual Authentication, Server, Session Establishment 555](#_Toc139446846)

[4.5.9 ES15 (SM-DS -- SM-DS): TLS, Mutual Authentication, Server, Session Establishment 555](#_Toc139446847)

[4.5.10 ES11 (LPA -- SM-DS): TLS, Server Authentication, Session Establishment 556](#_Toc139446848)

[4.6 TLS Interface 556](#_Toc139446849)

[4.6.1 TLS, Mutual Authentication, Client, TLS Establishment 556](#_Toc139446850)

[4.6.2 TLS, Mutual Authentication, Server, TLS Establishment 568](#_Toc139446851)

[4.6.3 TLS, Server Authentication, TLS Establishment 577](#_Toc139446852)

[4.7 LPAe Interfaces 580](#_Toc139446853)

[5 Procedure - Behaviour Testing 580](#_Toc139446854)

[5.1 General Overview 580](#_Toc139446855)

[5.2 eUICC Behaviour 580](#_Toc139446856)

[5.2.1 Retry mechanism 580](#_Toc139446857)

[5.2.2 Forbidden PPRs 586](#_Toc139446858)

[5.2.3 eUICC's RAT 588](#_Toc139446859)

[5.2.4 eUICC File Structure 589](#_Toc139446860)

[5.2.5 eUICC Delete Profile Process 590](#_Toc139446861)

[5.2.6 eUICC Enable Profile Process 592](#_Toc139446862)

[5.2.7 eUICC Disable Profile Process 593](#_Toc139446863)

[5.2.8 eUICC Notifications 594](#_Toc139446864)

[5.3 Platform Procedures 596](#_Toc139446865)

[5.3.1 Profile Download and Installation Procedure 596](#_Toc139446866)

[5.3.2 Common Mutual Authentication Process 596](#_Toc139446867)

[5.3.3 Profile Download and Installation Process 596](#_Toc139446868)

[5.4 Device Procedures 599](#_Toc139446869)

[5.4.1 Local Profile Management - Add Profile 599](#_Toc139446870)

[5.4.2 Local Profile Management – ListProfiles 625](#_Toc139446871)

[5.4.3 Local Profile Management - SetNickname 626](#_Toc139446872)

[5.4.4 Local Profile Management - Delete Profile 629](#_Toc139446873)

[5.4.5 Local Profile Management - Enable Profile 636](#_Toc139446874)

[5.4.6 Local Profile Management- Disable Profile 642](#_Toc139446875)

[5.4.7 Local eUICC Management - Retrieve EID Process 645](#_Toc139446876)

[5.4.8 Local eUICC Management - eUICC Memory Reset Process 645](#_Toc139446877)

[5.4.9 Local eUICC Management–- eUICC Test Memory Reset Process 653](#_Toc139446878)

[5.4.10 Local eUICC Management – Set/Edit Default SM-DP+ Address Process 653](#_Toc139446879)

[5.4.11 Device Power On – Profile Discovery 656](#_Toc139446880)

[6 End-to-End Testing 659](#_Toc139446881)

[7 External Test Specifications 659](#_Toc139446882)

[7.1 TCA eUICC Profile Package Test Specification 659](#_Toc139446883)

[Annex A Constants 660](#_Toc139446884)

[A.1 Generic Constants 660](#_Toc139446885)

[A.2 Test Certificates and Test Keys 673](#_Toc139446886)

[Annex B Dynamic Content 682](#_Toc139446887)

[Annex C Methods And Procedures 693](#_Toc139446888)

[C.1 Methods 693](#_Toc139446889)

[C.2 Procedures 708](#_Toc139446890)

[Annex D Commands And Responses 729](#_Toc139446891)

[D.1 ES8+ Requests And Responses 729](#_Toc139446892)

[D.1.1 ES8+ Requests 729](#_Toc139446893)

[D.2 ES9+ Requests And Responses 748](#_Toc139446894)

[D.2.1 ES9+ Requests 748](#_Toc139446895)

[D.2.2 ES9+ Responses 782](#_Toc139446896)

[D.3 ES10x Requests And Responses 797](#_Toc139446897)

[D.3.1 ES10x Requests 797](#_Toc139446898)

[D.3.2 ES10x Responses 809](#_Toc139446899)

[D.4 APDU 836](#_Toc139446900)

[D.4.1 APDU Commands 836](#_Toc139446901)

[D.4.2 R-APDU Chaining 837](#_Toc139446902)

[D.5 ES6 Requests And Responses 838](#_Toc139446903)

[D.5.1 ES6 Requests 838](#_Toc139446904)

[D.6 ES11 Requests And Responses 839](#_Toc139446905)

[D.6.1 ES11 Requests 839](#_Toc139446906)

[D.6.2 ES11 Responses 846](#_Toc139446907)

[D.7 ES12 Requests And Responses 847](#_Toc139446908)

[D.8 ES15 Requests And Responses 848](#_Toc139446909)

[D.9 Common Server Responses 848](#_Toc139446910)

[D.10 ES2+ Requests And Responses 857](#_Toc139446911)

[Annex E Profiles 858](#_Toc139446912)

[Annex F IUT Settings 873](#_Toc139446913)

[F.1 eUICC Settings 873](#_Toc139446914)

[F.2 Platforms Settings 874](#_Toc139446915)

[F.3 Device Settings 874](#_Toc139446916)

[F.4 Common Settings 875](#_Toc139446917)

[Annex G Initial States 877](#_Toc139446918)

[G.1 Device 877](#_Toc139446919)

[G.1.1 Device (default) 877](#_Toc139446920)

[G.1.2 Companion Device connected to a Primary Device 877](#_Toc139446921)

[G.1.3 Test eUICC Settings 877](#_Toc139446922)

[G.2 eUICC 879](#_Toc139446923)

[G.2.1 Common Initial States 879](#_Toc139446924)

[G.2.2 For eUICC supporting NIST P-256 880](#_Toc139446925)

[G.2.3 For eUICC supporting BrainpoolP256r1 880](#_Toc139446926)

[G.2.4 With default RAT configuration 880](#_Toc139446927)

[G.2.5 With Additional PPARs in the RAT 880](#_Toc139446928)

[G.2.6 Clean-up procedure 880](#_Toc139446929)

[G.3 SM-DP+ and SM-DS 881](#_Toc139446930)

[Annex H Icons and QR Codes 882](#_Toc139446931)

[Annex I Requirements 883](#_Toc139446932)

[Annex J Integrated eUICC Testing (Normative) 884](#_Toc139446933)

[Annex K Document Management 887](#_Toc139446934)

[J.1 Document History 887](#_Toc139446935)

[J.2 Other Information 892](#_Toc139446936)

# Introduction

## 1.1 Overview

The main aim of the GSMA Remote SIM Provisioning specifications [2] & [3] is to provide solution for the Remote SIM Provisioning of Consumer Devices. The adoption of this technical solution will provide the basis for global interoperability between different Operator deployment scenarios, for example network equipment (e.g. Subscription Manager Data Preparation (SM-DP+)) and various eUICC platforms.

This Test Plan provides a set of test cases to be used for testing the implementations of the provisioning system specifications documents [2] & [3]. This document offers to the involved entities an unified test strategy and ensures interoperability between different implementations.

## 1.2 Scope

This document is intended for:

* Parties which develop test tools and platforms
* Vendors (Device and eUICC Manufacturers, SM-DP+ and SM-DS Providers)
* Operators

The Test Plan consists of a set of relevant test cases for testing all entities involved in the eUICC remote provisioning system. The Implementations Under Test (IUT) are:

* the eUICC
* the LPAd for a Standalone and Companion Device
* the SM-DP+
* the SM-DS

The testing scopes developed in this document are:

* Interface compliance testing: Test cases to verify the compliance of the interfaces within the system.
* System behaviour testing: Test cases to verify the functional behaviour of the system.

Each test case specified within this Test Plan refers to one or more requirements.

The Test Plan contains test cases for the following versions of SGP.22:

* GSMA RSP Technical Specification V2.2 [2b]
* GSMA RSP Technical Specification V2.2. x (x≥1) [2c]
* GSMA RSP Technical Specification V2.3 [2d]
* GSMA RSP Technical Specification V2.4 [2e]
* GSMA RSP Technical Specification V2.5 [2]

This document includes an applicability table providing an indication whether test cases are relevant for a specific entity.

## 1.3 Definition of Terms

In addition to the terms which are defined below, the terms defined in SGP.22 [2] also apply

| Term | Description |
| --- | --- |
| End User | The person using the Device. |
| Integrated eUICC Test Interface | An external interface provided by its manufacturer for the purpose of testing eUICC functionality. |
| Standalone Device | A Device which provides all the capabilities to be able to be used in an RSP environment and needs no other Device for the purpose of Remote SIM Provisioning. |
| Test Plan | Current document describing the test cases that allow the RSP ecosystem to be tested. |

## 1.4 Abbreviations

In addition to the abbreviations which are defined below, the abbreviations defined in SGP.22 [2] also apply.

| Abbreviation | Description |
| --- | --- |
| APDU | Application Protocol Data Unit |
| ATR | Answer To Reset |
| C-APDU | Command APDU |
| CCID | (USB) Chip Card Interface Device |
| DER TLV | Distinguished Encoding Rules - Tag Length Value |
| FCP | File Control Parameters |
| HW | Hardware |
| IUT | Implementation Under Test |
| KVN | Key Version Number |
| OCE | Off-Card Entity |
| OS | Operating System |
| PIR | Profile Installation Result |
| POR | Proof Of Receipt |
| R-APDU | Response APDU |
| SoC | System on a Chip |
| SP | Service Provider |
| SSD | Supplemental Security Domain |
| USB | Universal Serial Bus |

## 1.5 Document Cross-references

| Ref | Document Number | Title |
| --- | --- | --- |
| [1] | SGP.02 | GSMA "Remote Provisioning of Embedded UICC Technical specification" V3.1 |
| [2] | SGP.22 | RSP Technical Specification V2.5 |
| [2b] | SGP.22 | RSP Technical Specification V2.2 |
| [2c] | SGP.22 | RSP Technical Specification V2.2.x (x≥1) |
| [2d] | SGP.22 | RSP Technical Specification V2.3 |
| [2e] | SGP.22 | RSP Technical Specification V2.4 |
| [3] | SGP.21 | RSP Architecture V2.5 |
| [3b] | SGP.21 | RSP Architecture V2.2 |
| [3c] | SGP.21 | RSP Architecture V2.3 |
| [3d] | SGP.21 | RSP Architecture V2.4 |
| [4] | eUICC Profile Package | Trusted Connectivity Alliance (formerly SIMalliance) eUICC Profile Package: Interoperable Format Technical Specification V2.1 or later |
| [5] | ETSI TS 102 221 | Smart Cards; UICC-Terminal interface |
| [6] | GPC\_SPE\_034 | GlobalPlatform Card Specification v.2.3 |
| [7] | ISO/IEC 7816-4:2013 | Identification cards – Integrated circuit cards - Part 4: Organization, security and commands for interchange |
| [8] | RFC 5639 | Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation |
| [9] | ANSSI ECC FRP256V1 | Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français. JORF n°0241 du 16 octobre 2011 page 17533. texte n° 30. 2011 |
| [10] | ITU E.118 | The international telecommunication charge card |
| [11] | NIST SP 800-56A | NIST Special Publication SP 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revision 2), May 2013 |
| [12] | 3GPP TS 23.003 | Digital cellular telecommunications system (Phase 2+);  Universal Mobile Telecommunications System (UMTS);  Numbering, addressing and identification |
| [13] | ETSI TS 102 225 | Secured packet structure for UICC based applications; Release 12 |
| [14] | ETSI TS 102 226 | Remote APDU structure for UICC based applications; Release 9 |
| [15] | TS.26 | GSMA NFC Handset Requirements V9.0 |
| [16] | ITU-T X.690 (11/2008) | ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) including Corrigendum 1 and 2 |
| [17] | ETSI TS 102 241 | Smart cards; UICC Application Programming Interface (UICC API) for Java Card™ |
| [18] | 3GPP TS 31.102 | Characteristics of the Universal Subscriber Identity Module (USIM) application |
| [19] | GPC\_SPE\_095 | GlobalPlatform Card - Digital Letter of Approval - Version 1.0 |
| [20] | RFC 2119 | Key words for use in RFCs to Indicate Requirement Levels, S. Bradner  <http://www.ietf.org/rfc/rfc2119.txt> |
| [21] | Void |  |
| [22] | 3GPP TS 23.040 | Technical realization of the Short Message Service (SMS) |
| [23] | TCA Test | Trusted Connectivity Alliance (TCA) eUICC Profile Package: Interoperable Format Test Specification Version 3.2.2 |
| [24] | RFC 4492 | Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) |
| [25] | SGP.26 | RSP Test Certificates Definition v1.5 |
| [26] | 3GPP TS 29.002 | Mobile Application Part (MAP) specification |
| [27] | RFC 5246 | The Transport Layer Security (TLS) Protocol Version 1.2 |
| [28] | GSMA PRD AA.35 | Procedures for Industry Specifications Product |
| [29] | CCID Rev 1.1 | CCID Specification for Integrated Circuit(s) Cards Interface Devices |

## 1.6 Conventions

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document SHALL be interpreted as described in RFC 2119 [[20]](#RFC2119).

# Testing Rules

## Applicability

### Format of the Optional Features Table

The columns in Table 4 have the following meaning:

|  |  |
| --- | --- |
| Column | Meaning |
| Option | The optional feature supported or not by the implementation. |
| Mnemonic | The mnemonic column contains mnemonic identifiers for each item. |

Table 1: Format of the Optional Features Table

### Format of the Applicability Table

The applicability of every test in Table 5 is formally expressed by the use of a Boolean expression defined in the following clause.

The columns in Table 5 have the following meaning:

|  |  |
| --- | --- |
| Column | Meaning |
| Test case | The "Test case" column gives a reference to the test case number detailed in the present document and is required to validate the implementation of the corresponding item in the "Name" column. |
| Name | In the "Name" column, a short non-exhaustive description of the test is found. |
| Roles | SM-DP+, SM-DS, Device, LPAd, LPAe or eUICC Entities under test that take in charge the functions used in the test case. |
| Version | This column specifies which test cases are applicable for the given SGP.22 version. The column for the version declared in #IUT\_RSP\_VERSION shall be used.  See clause 2.1.3 'Applicability and Notations'. |
| Test Env. | Test environment used for executing the test case. |

Table 2: Format of the Applicability Table

### Applicability and Notations

The following notations are used for the Applicability column:

| Applicability code | Meaning |
| --- | --- |
| M | mandatory - the capability is required to be supported. |
| N/A | not applicable - in the given context, it is impossible to use the capability. |
| Ci | conditional - the requirement on the capability depends on the support of other items. "i" is an integer identifying an unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE...) ELSE ..." is to be used to avoid ambiguities. |

Table 3: Applicability and Notations

### Optional Features Table

The supplier of the implementation SHALL state the support of possible options in Table 5.

|  |  |
| --- | --- |
| eUICC Options | Mnemonic |
| The eUICC supports NIST P-256 [11] for signing and for verification (see NOTE 2) | O\_E\_NIST |
| The eUICC supports brainpoolP256r1 [8] for signing and for verification (see NOTE 2) | O\_E\_BRP |
| The eUICC supports FRP256V1 [9] for signing and for verification (see NOTE 2) | O\_E\_FRP |
| The eUICC supports Test Profiles | O\_E\_TEST\_PROF |
| The eUICC supports CRL | O\_E\_CRL |
| The eUICC supports the LPAe | O\_E\_LPAe |
| The eUICC stores the otPK.eUICC.ECKA / otSK.eUICC.ECKA from previous unsuccessful download attempt for future retry | O\_E\_REUSE\_OTPK |
| The eUICC can hold two PIR | O\_E\_2\_PIR |
| The eUICC terminates EnableProfile and DisableProfile with error "catBusy" when a proactive session is ongoing and the refresh Flag is set. | O\_E\_CATBUSY\_EN\_DIS\_REFRESH |
| The eUICC terminates EnableProfile and DisableProfile with error "catBusy" when a proactive session is ongoing and the refresh Flag is not set | O\_E\_CATBUSY\_EN\_DIS\_NOREFRESH |
| The eUICC terminates eUICCMemoryReset with error "catBusy" when a proactive session is ongoing | O\_E\_CATBUSY\_MR |
| The eUICC is based on an integrated TRE | O\_E\_INTEGRATED |
| The eUICC supports Service Specific Data in Profile Metadata | O\_E\_SERVICES\_SPECIFIC\_DATA |
| Device Options | Mnemonic |
| The Device supports LPAd | O\_D\_LPAD |
| The Device supports GSM/GERAN | O\_D\_GSM\_GERAN |
| The Device supports UMTS/UTRAN | O\_D\_UMTS\_UTRAN |
| The Device supports cdma2000 1X | O\_D\_CDMA2000\_1X |
| The Device supports cdma2000 HRPD | O\_D\_CDMA2000\_HRPD |
| The Device supports cdma2000 eHRPD | O\_D\_CDMA2000\_EHRPD |
| The Device supports LTE/E-UTRAN | O\_D\_LTE |
| The Device supports NFC as defined in TS26 | O\_D\_NFC\_TS26 |
| The Device supports eUICC CRL | O\_D\_CRL |
| Initiation of the Enable Profile procedure is allowed on a Profile that is enabled already | O\_D\_ENPROF |
| Initiation of the Enable Profile procedure is allowed even if the currently enabled Profile has PPR1 | O\_D\_ENPREVPPR1 |
| Device supports only cellular connectivity (see NOTE 1) | O\_D\_ONLY\_CELLULAR\_CONNECTIVITY |
| Device offers a user interface to enter a PIN for user authentication | O\_D\_PIN |
| Device allows the End User to initiate the disabling or deletion of an enabled Profile with ppr1 | O\_D\_DISDELPPR1 |
| Device allows the End User to initiate the deletion of a Profile with ppr2 | O\_D\_DELPPR2 |
| Initiation of the Disable Profile procedure is allowed on a Profile that is disabled already | O\_D\_DISPROF |
| Initiation of Disable Profile procedure is allowed even if the currently enabled Profile has PPR1 | O\_D\_DISPPR1 |
| Device retries after unsuccessful CC entry attempt | O\_D\_CC\_RETRY |
| The Device provides the LUI functionality to postpone Profile Download | O\_D\_EU\_POSTPONED |
| Device supports Power-on Profile discovery | O\_D\_POW\_ON\_PROF\_DISCOVERY |
| The Device provides the LUI functionality to reject Profile Download | O\_D\_EU\_REJECT |
| The Device supports Set/Edit Nickname procedure as defined in SGP.22 [2] section 3.2.6 and displaying the profile's Nickname | O\_D\_NICKNAME |
| The Device supports Add Profile and Enable Profile in one combined operation (See NOTE 3) | O\_D\_ADD\_ENABLE\_COMBINED |
| The Device supports Add Profile and Enable Profile as separated operations (See NOTE 3) | O\_D\_ADD\_ENABLE\_SEPARATED |
| Initiation of Add Profile procedure is allowed even if the currently enabled Profile has PPR1 | O\_D\_ADDPREPPR1 |
| The Device supports Set/Edit Default SM-DP+ Address procedure as defined in SGP.22 [2] section 3.3.4 | O\_D\_DEFAULT\_DP\_ADDRESS |
| The Device supports a removable eUICC and downloading a profile containing PPRs to the removable eUICC. | O\_D\_REMOVABLE\_DOWNLOAD\_PPR |
| The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR1 is allowed and End User Consent is required. | O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ |
| The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR1 is allowed and End User Consent is NOT required. | O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_NOT\_REQ |
| The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR2 is allowed and End User Consent is required. | O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_REQ |
| The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR2 is allowed and End User Consent is NOT required. | O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_NOT\_REQ |
| The Device supports SM-DS | O\_D\_SMDS |
| The Device supports Disable Profile as separated operation, without automatically deleting it. | O\_D\_DISABLE\_SEPARATED |
| The Device supports Profile download using Default SM-DP+ | O\_D\_DEFAULT\_DP |
| SM-DP+ Options | Mnemonic |
| SM-DP+ reuses otPK.eUICC.ECKA from previous unsuccessful download attempt | O\_P\_REUSE\_OTPK |
| SM-DP+ supports usage of session keys (S-ENC, S-MAC) for profile protection | O\_P\_SESSION\_KEYS |
| SM-DP+ accepts receiving two identical function call successively via ES2+ | O\_P\_ES2+\_RETRY |
| SM-DP+ supports brainpoolP256r1 for TLS handshake | O\_P\_TLS\_BRP |
| SM-DS Options | Mnemonic |
| SM-DS is an Alternative SM-DS.  NOTE: If an SM-DS is not an Alternative SM-DS then it is a Root SM-DS. | O\_S\_ALT |
| SM-DS supports brainpoolP256r1 for TLS handshake | O\_S\_TLS\_BRP |
| NOTE 1: Devices which supports O\_D\_ONLY\_CELLULAR\_CONNECTIVITY are out of scope of the current version of this document.  NOTE 2: For this version of test specification:   * O\_E\_FRP is not applicable * The eUICC SHALL support either O\_E\_NIST or O\_E\_BRP or both   NOTE 3: The Device SHALL support at least one of O\_D\_ADD\_ENABLE\_COMBINED or O\_D\_ADD\_ENABLE\_SEPARATED. It is valid to support both options. | |

Table 4: Options

### Applicability Table

Table 5 specifies the applicability of each test case. See clause 2.1.2 for the format of this table.

| Test case | Name | | Role | V2.2 or V2.2.1 | V2.2.2 | V2.3 | V2.4 and V2.5 | Test Env. |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | eUICC Interfaces Compliance Testing | | | | | | | | |
| 4.2.1.2.1 | TC\_eUICC\_ATR\_And\_ISDR\_Selection | | eUICC | C006 | C006 | C006 | C006 | TE\_eUICC |
| 4.2.2.2.1 | TC\_eUICC\_ES6.UpdateMetadata | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.3.2.1 | TC\_eUICC\_ES8+.InitialiseSecureChannel | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.4.2.1 | TC\_eUICC\_ES8+.ConfigureISDP | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.5.2.1 | TC\_eUICC\_ES8+.StoreMetadata | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.5.2.2 | TC\_eUICC\_ES8+.StoreMetadata\_Service\_Specific\_Data | | eUICC | N/A | N/A | N/A | C057 | TE\_eUICC |
| 4.2.6.2.1 | TC\_eUICC\_ES8+.ReplaceSessionKeys | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.7.2.1 | TC\_eUICC\_ES8+.LoadProfileElements | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.8.2.1 | TC\_eUICC\_ES10a.GetEuiccConfiguredAddresses | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.9.2.1 | TC\_eUICC\_ES10a.SetDefaultDPAddress | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.10.2.1 | TC\_eUICC\_ES10b.PrepareDownloadNIST | | eUICC | C001 | C001 | C001 | C001 | TE\_eUICC |
| 4.2.10.2.2 | TC\_eUICC\_ES10b.PrepareDownloadBRP | | eUICC | C002 | C002 | C002 | C002 | TE\_eUICC |
| 4.2.10.2.3 | TC\_eUICC\_ES10b.PrepareDownloadFRP | | eUICC | C003 | C003 | C003 | C003 | TE\_eUICC |
| 4.2.10.2.4 | TC\_eUICC\_ES10b.PrepareDownloadErrorCases | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.11.2.1 | TC\_eUICC\_ES10b.LoadBoundProfilePackageNIST | | eUICC | C001 | C001 | C001 | C001 | TE\_eUICC |
| 4.2.11.2.2 | TC\_eUICC\_ES10b.LoadBoundProfilePackageBRP | | eUICC | C002 | C002 | C002 | C002 | TE\_eUICC |
| 4.2.11.2.3 | TC\_eUICC\_ES10b.LoadBoundProfilePackageFRP | | eUICC | C003 | C003 | C003 | C003 | TE\_eUICC |
| 4.2.11.2.4 | TC\_eUICC\_ES10b.LoadBoundProfilePackage\_ErrorCases | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.12.2.1 | TC\_eUICC\_ES10b.GetEUICCChallenge | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.13.2.1 | TC\_eUICC\_ES10b.GetEUICCInfo1 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.13.2.3 | TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_V2.2.x | | eUICC | M | M | N/A | N/A | TE\_eUICC |
| 4.2.13.2.4 | TC\_eUICC\_ES10b.GetEUICCInfo2 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.13.2.5 | TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_V2.3 | | eUICC | N/A | N/A | M | N/A | TE\_eUICC |
| 4.2.13.2.6 | TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_Integrated\_eUICC | | eUICC | N/A | N/A | C040 | C040 | TE\_eUICC |
| 4.2.13.2.7 | TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_V2.4\_or\_Higher | | eUICC | N/A | N/A | N/A | M | TE\_eUICC |
| 4.2.13.2.8 | TC\_eUICC\_ES10b.GetEUICCInfo\_SVN | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.13.2.9 | TC\_eUICC\_ES10b.GetEUICCInfo\_profileVersion | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.13.2.10 | TC\_eUICC\_ES10b.GetEUICCInfo\_additionalEuiccProfilePackageVersions | | eUICC | N/A | N/A | M | M | TE\_eUICC |
| 4.2.14.2.1 | TC\_eUICC\_ES10b.ListNotification  All test sequences except the sequence #5 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.14.2.1 | TC\_eUICC\_ES10b.ListNotification  Only the test sequence #5 | | eUICC | C025 | C025 | C025 | C025 | TE\_eUICC |
| 4.2.15.2.1 | TC\_eUICC\_ES10b.RetrieveNotificationsList  All test sequences except the sequences #5 and #15 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.15.2.1 | TC\_eUICC\_ES10b.RetrieveNotificationsList  Only the test sequences #5 and #15 | | eUICC | C025 | C025 | C025 | C025 | TE\_eUICC |
| 4.2.16.2.1 | TC\_eUICC\_ES10b.RemoveNotificationFromList  All test sequences except the sequence #5 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.16.2.1 | TC\_eUICC\_ES10b.RemoveNotificationFromList  Only the test sequence #5 | | eUICC | C025 | C025 | C025 | C025 | TE\_eUICC |
| 4.2.18.2.1 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_NIST | | eUICC | C001 | C001 | C001 | C001 | TE\_eUICC |
| 4.2.18.2.2 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_BRP | | eUICC | C002 | C002 | C002 | C002 | TE\_eUICC |
| 4.2.18.2.3 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_FRP | | eUICC | C003 | C003 | C003 | C003 | TE\_eUICC |
| 4.2.18.2.4 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_ErrorCases | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.18.2.5 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_BRP | | eUICC | C002 | C002 | C002 | C002 | TE\_eUICC |
| 4.2.18.2.6 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_NIST | | eUICC | C001 | C001 | C001 | C001 | TE\_eUICC |
| 4.2.18.2.7 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_FRP | | eUICC | C003 | C003 | C003 | C003 | TE\_eUICC |
| 4.2.18.2.8 | TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_ErrorCases | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.19.2.1 | TC\_eUICC\_ES10b.CancelSessionNIST | | eUICC | C001 | C001 | C001 | C001 | TE\_eUICC |
| 4.2.19.2.2 | TC\_eUICC\_ES10b.CancelSessionBRP | | eUICC | C002 | C002 | C002 | C002 | TE\_eUICC |
| 4.2.19.2.3 | TC\_eUICC\_ES10b.CancelSessionFRP | | eUICC | C003 | C003 | C003 | C003 | TE\_eUICC |
| 4.2.19.2.4 | TC\_eUICC\_ES10b.CancelSession\_ErrorCase | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.20.2.1 | TC\_eUICC\_ES10c.GetProfilesInfo | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.21.2.1 | TC\_eUICC\_ES10c.EnableProfile\_Case3  All test sequences except the sequence #7, sequence #8, sequence #9 and sequence #10 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.21.2.1 | TC\_eUICC\_ES10c.EnableProfile\_Case3  Only the sequence #7 and sequence #9 | | eUICC | N/A | C033 | C033 | C033 | TE\_eUICC |
| 4.2.21.2.1 | TC\_eUICC\_ES10c.EnableProfile\_Case3  Only the sequence #8 and sequence #10 | | eUICC | N/A | C037 | C037 | C037 | TE\_eUICC |
| 4.2.21.2.2 | TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case3  All test sequences except the sequence #7 and sequence #8 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.21.2.2 | TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case3  Only sequence #7 | | eUICC | M | C036 | C036 | C036 | TE\_eUICC |
| 4.2.21.2.2 | TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case3  Only sequence #8 | | eUICC | M | C032 | C032 | C032 | TE\_eUICC |
| 4.2.21.2.3 | TC\_eUICC\_ES10c.EnableProfile\_Case4  All test sequences except the sequence #7, sequence #8, sequence #9 and sequence #10 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.21.2.3 | TC\_eUICC\_ES10c.EnableProfile\_Case4  Only the sequence #7 and sequence#9 | | eUICC | N/A | C033 | C033 | C033 | TE\_eUICC |
| 4.2.21.2.3 | TC\_eUICC\_ES10c.EnableProfile\_Case4  Only the sequence #8 and sequence#10 | | eUICC | N/A | C037 | C037 | C037 | TE\_eUICC |
| 4.2.21.2.4 | TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case4  All test sequences except the sequence #7 and sequence #8 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.21.2.4 | TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case4  Only the sequence #7 | | eUICC | M | C036 | C036 | C036 | TE\_eUICC |
| 4.2.21.2.4 | TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case4  Only the sequence #8 | | eUICC | M | C032 | C032 | C032 | TE\_eUICC |
| 4.2.22.2.1 | TC\_eUICC\_ES10c.DisableProfile\_Case3  All test sequences except the sequence #7, sequence #8, sequence #9 and sequence #10 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.22.2.1 | TC\_eUICC\_ES10c.DisableProfile\_Case3  Only the sequence #7 and sequence#9 | | eUICC | N/A | C033 | C033 | C033 | TE\_eUICC |
| 4.2.22.2.1 | TC\_eUICC\_ES10c.DisableProfile\_Case3  Only the sequence #8 and sequence#10 | | eUICC | N/A | C037 | C037 | C037 | 4.2.22.2.1 |
| 4.2.22.2.2 | TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case3  All test sequences except the sequence #7 and sequence #8 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.22.2.2 | TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case3  Only the sequence #7 | | eUICC | M | C036 | C036 | C036 | TE\_eUICC |
| 4.2.22.2.2 | TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case3  Only the sequence #8 | | eUICC | M | C032 | C032 | C032 | TE\_eUICC |
| 4.2.22.2.3 | TC\_eUICC\_ES10c.DisableProfile\_Case4  All test sequences except the sequence #7, sequence #8, sequence #9 and sequence #10 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.22.2.3 | TC\_eUICC\_ES10c.DisableProfile\_Case4  Only the sequence #7 and sequence #9 | | eUICC | N/A | C033 | C033 | C033 | TE\_eUICC |
| 4.2.22.2.3 | TC\_eUICC\_ES10c.DisableProfile\_Case4  Only the sequence #8 and sequence #10 | | eUICC | N/A | C037 | C037 | C037 | TE\_eUICC |
| 4.2.22.2.4 | TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case4  All test sequences except the sequence #7 and sequence #8 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.22.2.4 | TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case4  Only the sequence #7 | | eUICC | M | C036 | C036 | C036 | TE\_eUICC |
| 4.2.22.2.4 | TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case4  Only the sequence #8 | | eUICC | M | C032 | C032 | C032 | TE\_eUICC |
| 4.2.23.2.1 | TC\_eUICC\_ES10c.DeleteProfile\_Case3 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.23.2.2 | TC\_eUICC\_ES10c.DeleteProfile\_ErrorCases\_Case3 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.23.2.3 | TC\_eUICC\_ES10c.DeleteProfile\_Case4 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.23.2.4 | TC\_eUICC\_ES10c.DeleteProfile\_ErrorCases\_Case4 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.24.2.1 | TC\_eUICC\_ES10c.eUICCMemoryReset  All test sequences except the sequence #5 and sequence #6 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.24.2.1 | TC\_eUICC\_ES10c.eUICCMemoryReset  Only the sequence #5 and sequence #6 | | eUICC | N/A | C039 | C039 | C039 | TE\_eUICC |
| 4.2.24.2.2 | TC\_eUICC\_ES10c.eUICCMemoryReset\_ErrorCases  Test sequence #1 | | eUICC | M | C038 | C038 | C038 | TE\_eUICC |
| 4.2.24.2.2 | TC\_eUICC\_ES10c.eUICCMemoryReset\_ErrorCases  Test sequence #2 | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.25.2.1 | TC\_eUICC\_ES10c.GetEID | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.26.2.1 | TC\_eUICC\_ES10c.SetNickname | | eUICC | M | M | M | M | TE\_eUICC |
| 4.2.27.2.1 | TC\_eUICC\_ES10b.GetRAT | | eUICC | M | M | M | M | TE\_eUICC |
|  |  |
| 4.3.1.2.1 | TC\_SM-DP+\_ES2+.DownloadOrder | | SM-DP+ | M | M | M | M | TE\_P3 |
| 4.3.1.2.2 | TC\_SM-DP+\_ES2+.DownloadOrder\_RetryCases | | SM-DP+ | N/A | C042 | C042 | C042 | TE\_P3 |
| 4.3.1.2.3 | TC\_SM-DP+\_ES2+.DownloadOrder\_ErrorCases | | SM-DP+ | M | M | M | M | TE\_P3 |
| 4.3.2.2.1 | TC\_SM-DP+\_ES2+.ConfirmOrder | | SM-DP+ | M | M | M | M | TE\_P3 |
| 4.3.2.2.2 | TC\_SM-DP+\_ES2+.ConfirmOrder\_RetryCases | | SM-DP+ | N/A | C042 | C042 | C042 | TE\_P3 |
| 4.3.2.2.3 | TC\_SM-DP+\_ES2+.ConfirmOrder\_ErrorCases | | SM-DP+ | M | M | M | M | TE\_P3 |
| 4.3.3.2.1 | TC\_SM-DP+\_ES2+.CancelOrder | | SM-DP+ | M | M | M | M | TE\_P3 |
| 4.3.3.2.2 | TC\_SM-DP+\_ES2+.CancelOrder\_ErrorCases | | SM-DP+ | M | M | M | M | TE\_P3 |
| 4.3.12.2.1 | TC\_SM-DP+\_ES9+.InitiateAuthenticationNIST | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.12.2.2 | TC\_SM-DP+\_ES9+.InitiateAuthenticationFRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.12.2.3 | TC\_SM-DP+\_ES9+.InitiateAuthenticationBRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.13.2.1 | TC\_SM-DP+\_ES9+.GetBoundProfilePackageNIST  Test sequences #1, #2 and #5 | | SM-DP+ | C028 | C028 | C028 | C028 | TE\_P2 |
| 4.3.13.2.1 | TC\_SM-DP+\_ES9+.GetBoundProfilePackageNIST  Test sequences #3, #4 and #6 | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.13.2.2 | TC\_SM-DP+\_ES9+.GetBoundProfilePackageFRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.13.2.3 | TC\_SM-DP+\_ES9+.GetBoundProfilePackageBRP  Test sequence #1 | | SM-DP+ | C028 | C028 | C028 | C028 | TE\_P2 |
| 4.3.13.2.3 | TC\_SM-DP+\_ES9+.GetBoundProfilePackageBRP  Test sequence #2 | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.13.2.4 | TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_ReuseOTPK\_NIST  Test sequences #1, #2, #5 and #6 | | SM-DP+ | C029 | C029 | C029 | C029 | TE\_P2 |
| 4.3.13.2.4 | TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_ReuseOTPK\_NIST  Test sequences #3, #4, #7, #8 and #9 | | SM-DP+ | C015 | C015 | C015 | C015 | TE\_P2 |
| 4.3.13.2.7 | TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_DifferentOTPK\_NIST  Test sequences #1 and #2 | | SM-DP+ | C030 | C030 | C030 | C030 | TE\_P2 |
| 4.3.13.2.7 | TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_DifferentOTPK\_NIST  Test sequences #3 and #4 | | SM-DP+ | C016 | C016 | C016 | C016 | TE\_P2 |
| 4.3.13.2.10 | TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_ErrorCasesNIST | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.14.2.1 | TC\_SM-DP+\_ES9+.AuthenticateClientNIST | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.14.2.2 | TC\_SM-DP+\_ES9+.AuthenticateClientNIST\_ErrorCases | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.14.2.3 | TC\_SM-DP+\_ES9+.AuthenticateClientFRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.14.2.5 | TC\_SM-DP+\_ES9+.AuthenticateClientBRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.14.2.6 | TC\_SM-DP+\_ES9+.AuthenticateClient\_RetryCases\_Reuse\_OTPK | | SM-DP+ | C015 | C015 | C015 | C015 | TE\_P2 |
| 4.3.15.2.1 | TC\_SM-DP+\_ES9+\_HandleNotificationNIST | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.15.2.2 | TC\_SM\_DP+\_ES9+\_HandleNotificationFRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.15.2.3 | TC\_SM-DP+\_ES9+\_HandleNotificationBRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.16.2.1 | TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientNIST | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.16.2.2 | TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageNIST | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.16.2.3 | TC\_SM\_DP+\_ES9+.CancelSession\_After\_AuthenticateClientFRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.16.2.4 | 4.3.16.2.4 TC\_SM\_DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageFRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.16.2.5 | TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientBRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.16.2.6 | TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageBRP | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.17.1 | TC\_SM-DP+\_ES9+\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST | | SM-DP+ | M | M | M | M | TE\_P2 |
| 4.3.17.2 | TC\_SM-DP+\_ES9+\_Server\_Authentication\_for\_HTTPS\_EstablishmentBRP | | SM-DP+ | M | M | C053 | C053 | TE\_P2 |
| 4.3.20.1 | TC\_SM-DP+\_ES12\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST | | SM-DP+ | M | M | M | M | TE\_P1 |
| 4.3.20.2 | TC\_SM-DP+\_ES12\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP | | SM-DP+ | M | M | C053 | C053 | TE\_P1 |
|  |  |
| 4.4.21.2.1 | TC\_LPAd\_InitiateAuthentication\_Nominal | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.21.2.2 | TC\_LPAd\_InitiateAuthentication\_ErrorCases | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.22.2.1 | TC\_LPAd\_ES9+\_GetBoundProfilePackage\_Nominal | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.22.2.2 | TC\_LPAd\_ES9+\_GetBoundProfilePackage\_Retry | | LPAd | C005 | C005 | C005 | C005 |  |
| 4.4.22.2.3 | TC\_LPAd\_ES9+\_GetBoundProfilePackage\_Error | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.23.2.1 | TC\_LPAd\_AuthenticatClient\_Nominal | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.23.2.2 | TC\_LPAd\_AuthenticateClient\_ErrorCases | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.24.2.1 | TC\_LPAd\_ES9+\_HandleNotification\_Nominal  All test sequences except the sequence #03 | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.24.2.1 | TC\_LPAd\_ES9+\_HandleNotification\_Nominal  Only the test sequence #03 | | LPAd | C060 | C060 | C060 | C060 |  |
| 4.4.25.2.1 | TC\_LPAd\_ES9+\_CancelSession\_Nominal  All test sequences except the sequence #02 | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.25.2.1 | TC\_LPAd\_ES9+\_CancelSession\_Nominal  Only the test sequences #02 | | LPAd | C023 | C023 | C023 | C023 |  |
| 4.4.25.2.2 | TC\_LPAd\_ES9+\_CancelSession\_EndUserPostponed\_Nominal | | LPAd | C008 | C008 | C008 | C008 |  |
| 4.4.25.2.3 | TC\_LPAd\_ES9+\_CancelSession\_Error | | LPAd | C026 | C026 | C026 | C026 |  |
| 4.4.25.2.4 | TC\_LPAd\_ES9+\_CancelSession\_PPRs  Only the test sequence #01 | | LPAd | C045 | C045 | C045 | C045 |  |
| 4.4.25.2.4 | TC\_LPAd\_ES9+\_CancelSession\_PPRs  Only the test sequence #02 | | LPAd | C046 | C046 | C046 | C046 |  |
| 4.4.26.2.1 | TC\_LPAd\_HTTPS\_Nominal | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.26.2.2 | TC\_LPAd\_HTTPS\_ErrorCases | | LPAd | C007 | C007 | C007 | C007 |  |
| 4.4.27.2.1 | TC\_LPAd\_ES11\_InitiateAuthentication\_Nominal | | LPAd | C058 | C058 | C058 | C058 |  |
| 4.4.27.2.2 | TC\_LPAd\_ES11\_InitiateAuthentication\_ErrorCases | | LPAd | C058 | C058 | C058 | C058 |  |
| 4.4.28.2.1 | TC\_LPAd\_ES11\_AuthenticateClient\_Nominal | | LPAd | C058 | C058 | C058 | C058 |  |
| 4.4.28.2.2 | TC\_LPAd\_ES11\_AuthenticateClient\_ErrorCases | | LPAd | C058 | C058 | C058 | C058 |  |
| 4.4.29.2.1 | TC\_LPAd\_HTTPS\_Nominal | | LPAd | C058 | C058 | C058 | C058 |  |
| 4.4.29.2.2 | TC\_LPAd\_HTTPS\_Error | | LPAd | C058 | C058 | C058 | C058 |  |
|  |  |
| 4.5.1.2.1 | TC\_ROOT\_SM\_DS\_ES12.RegisterEvent | | SM-DS | C024 | C024 | C024 | C024 | TE\_S3 |
| 4.5.1.2.2 | TC\_ALT\_SM\_DS\_ES12.RegisterEvent | | SM-DS | C021 | C021 | C021 | C021 | TE\_SA2 |
| 4.5.2.2.1 | TC\_ROOT\_SM\_DS\_ES12.DeleteEvent | | SM-DS | C024 | C024 | C024 | C024 | TE\_S3 |
| 4.5.2.2.2 | TC\_ALT\_SM\_DS\_ES12.DeleteEvent | | SM-DS | C021 | C021 | C021 | C021 | TE\_SA2 |
| 4.5.2.2.3 | TC\_ALT\_SM\_DS\_ES12.DeleteEvent\_Error\_Nonexistant\_EventID | | SM-DS | C021 | C021 | C021 | C021 | TE\_S2 |
| 4.5.3.2.1 | TC\_ROOT\_SM\_DS\_ES15.RegisterEvent | | SM-DS | C024 | C024 | C024 | C024 | TE\_SR2 |
| 4.5.4.2.1 | TC\_ROOT\_SM\_DS\_ES15.DeleteEvent | | SM-DS | C024 | C024 | C024 | C024 | TE\_SR2 |
| 4.5.5.2.1 | TC\_SM\_DS\_ES11.InitiateAuthenticationNIST | | SM-DS | M | M | M | M | TE\_S1 |
| 4.5.5.2.2 | TC\_SM\_DS\_ES11.InitiateAuthenticationBRP | | SM-DS | M | M | M | M | TE\_S1 |
| 4.5.6.2.1 | TC\_SM\_DS\_ES11.AuthenticateClientNIST  All test sequences except sequences #07 and #08 | | SM-DS | M | M | M | M | TE\_S1 |
| 4.5.6.2.1 | TC\_SM\_DS\_ES11.AuthenticateClientNIST  Only test sequences #07 and #08 | | SM-DS | C021 | C021 | C021 | C021 | TE\_S1 |
| 4.5.6.2.2 | TC\_SM\_DS\_ES11.AuthenticateClientBRP  All test sequences except sequences #07 and #08 | | SM-DS | M | M | M | M | TE\_S1 |
| 4.5.6.2.2 | TC\_SM\_DS\_ES11.AuthenticateClientBRP  Only test sequences #07 and #08 | | SM-DS | C021 | C021 | C021 | C021 | TE\_S1 |
| 4.5.7.1 | TC\_ALT\_SM\_DS\_ES15\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST | | SM-DS | C021 | C021 | C021 | C021 | TE\_SA1 |
| 4.5.7.2 | TC\_ALT\_SM\_DS\_ES15\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP | | SM-DS | C021 | C021 | C055 | C055 | TE\_SA1 |
| 4.5.8.1 | TC\_SM\_DS\_ES12\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST | | SM-DS | M | M | M | M | TE\_S2 |
| 4.5.8.2 | TC\_SM\_DS\_ES12\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP | | SM-DS | M | M | C054 | C054 | TE\_S2 |
| 4.5.9.1 | TC\_ROOT\_SM\_DS\_ES15\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST | | SM-DS | C024 | C024 | C024 | C024 | TE\_SR1 |
| 4.5.9.2 | TC\_ROOT\_SM\_DS\_ES15\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP | | SM-DS | C024 | C024 | C056 | C056 | TE\_SR1 |
| 4.5.10.1 | TC\_SM\_DS\_ES11\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST | | SM-DS | M | M | M | M | TE\_S1 |
| 4.5.10.2 | TC\_SM\_DS\_ES11\_Server\_Authentication\_for\_HTTPS\_EstablishmentBRP | | SM-DS | M | M | C054 | C054 | TE\_S1 |
|  |  |
| 5.2.1.2.1 | TC\_eUICC\_PrepareDownload\_Retry\_ReuseOTKeys | | eUICC | C019 | C019 | C019 | C019 | TE\_eUICC |
| 5.2.1.2.2 | TC\_eUICC\_PrepareDownload\_Retry\_NewOTKeys | | eUICC | C020 | C020 | C020 | C020 | TE\_eUICC |
| 5.2.2.2.1 | TC\_eUICC\_ForbiddenPPRs | | eUICC | M | M | M | M | TE\_eUICC |
| 5.2.3.2.1 | TC\_eUICC\_GetProfilesInfo\_GetRAT\_RSPSession | | eUICC | M | M | M | M | TE\_eUICC |
| 5.2.4.2.1 | TC\_eUICC\_Default\_FileSystem | | eUICC | M | M | M | M | TE\_eUICC |
| 5.2.5.2.1 | TC\_eUICC\_DeleteProfile\_ISDP\_And\_Components | | eUICC | M | M | M | M | TE\_eUICC |
| 5.2.6.2.1 | TC\_eUICC\_EnableProfile\_Twice\_Notifications | | eUICC | M | M | M | M | TE\_eUICC |
| 5.2.7.2.1 | TC\_eUICC\_DisableProfile\_ApplicationManagement | | eUICC | M | M | M | M | TE\_eUICC |
| 5.2.8.2.1 | TC\_eUICC\_Enable\_Disable\_Delete\_Notifications | | eUICC | M | M | M | M | TE\_eUICC |
| 5.3.3.2.1 | TC\_SM-DP+\_ProfileMetadata | | SM-DP+ | M | M | M | M |  |
| 5.4.1.2.1 | TC\_LPAd\_AddProfile\_Manual\_Entry | | LPAd | C035 | C035 | C035 | C035 |  |
| 5.4.1.2.2 | TC\_LPAd\_AddProfile\_QRCode\_scanning | | LPAd | C035 | C035 | C035 | C035 |  |
| 5.4.1.2.3 | TC\_LPAd\_AddProfile\_ActivationCode\_InvalidFormat\_QRCode | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.1.2.4 | TC\_LPAd\_AddProfile\_ActivationCode\_InvalidFormat\_ManualEntry | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.1.2.5 | TC\_LPAd\_AddProfile\_ConfirmationCode\_smdpSigned2\_QR | | LPAd | C035 | C035 | C035 | C035 |  |
| 5.4.1.2.6 | TC\_LPAd\_AddProfile\_ConfirmationCode\_smdpSigned2\_Manual\_Entry | | LPAd | C035 | C035 | C035 | C035 |  |
| 5.4.1.2.7 | TC\_LPAd\_AddProfile\_default\_SM-DP+\_address | | LPAd | C063 | C063 | C063 | C063 |  |
| 5.4.1.2.8 | TC\_LPAd\_AddProfile\_QRCode\_with\_ConfirmationCode | | LPAd | C035 | C035 | C035 | C035 |  |
| 5.4.1.2.9 | TC\_LPAd\_AddProfile\_PPRs  Only the test sequence #01 | | LPAd | C047 | C047 | C047 | C047 |  |
| 5.4.1.2.9 | TC\_LPAd\_AddProfile\_PPRs  Only test sequence #2 | | LPAd | C048 | C048 | C048 | C048 |  |
| 5.4.1.2.9 | TC\_LPAd\_AddProfile\_PPRs  Only test sequence #3 | | LPAd | C051 | C051 | C051 | C051 |  |
| 5.4.1.2.11 | TC\_LPAd\_AddProfile\_Security\_Errors | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.1.2.12 | TC\_LPAd\_AddProfile\_Empty\_MatchingID | | LPAd | C035 | C035 | C035 | C035 |  |
| 5.4.1.2.13 | TC\_LPAd\_AddEnabledProfile\_Manual\_Entry | | LPAd | C034 | C034 | C034 | C034 |  |
| 5.4.1.2.14 | TC\_LPAd\_AddEnabledProfile\_QRCode\_scanning | | LPAd | C034 | C034 | C034 | C034 |  |
| 5.4.1.2.15 | TC\_LPAd\_AddEnabledProfile\_ConfirmationCode\_smdpSigned2\_QR | | LPAd | C034 | C034 | C034 | C034 |  |
| 5.4.1.2.16 | TC\_LPAd\_AddEnableProfile\_ConfirmationCode\_smdpSigned2\_Manual\_Entry | | LPAd | C034 | C034 | C034 | C034 |  |
| 5.4.1.2.17 | TC\_LPAd\_AddEnabledProfile\_default\_SM-DP+\_address | | LPAd | C062 | C062 | C062 | C062 |  |
| 5.4.1.2.18 | TC\_LPAd\_AddEnableProfile\_QRCode\_with\_ConfirmationCode | | LPAd | C034 | C034 | C034 | C034 |  |
| 5.4.1.2.19 | TC\_LPAd\_AddEnabledProfile\_PPRs  Only the test sequence #01 | | LPAd | C049 | C049 | C049 | C049 |  |
| 5.4.1.2.19 | TC\_LPAd\_AddEnabledProfile\_PPRs  Only test sequence #2 | | LPAd | C050 | C050 | C050 | C050 |  |
| 5.4.1.2.19 | TC\_LPAd\_AddEnabledProfile\_PPRs  Only test sequence #3 | | LPAd | C052 | C052 | C052 | C052 |  |
| 5.4.1.2.20 | TC\_LPAd\_AddEnableProfile\_Empty\_MatchingID | | LPAd | C034 | C034 | C034 | C034 |  |
| 5.4.2.2.1 | TC\_LPAd\_ListProfiles | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.3.2.1 | TC\_LPAd\_SetNickname | | LPAd | C027 | C027 | C027 | C027 |  |
| 5.4.3.2.2 | TC\_LPAd\_EditNickname | | LPAd | C027 | C027 | C027 | C027 |  |
| 5.4.4.2.1 | TC\_LPAd\_DeleteProfile\_Disabled\_without\_PPR | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.4.2.2 | TC\_LPAd\_DeleteProfile\_Enabled\_without\_PPR  Only the test sequence #01 | | LPAd | C009 | C009 | C009 | C009 |  |
| 5.4.4.2.2 | TC\_LPAd\_DeleteProfile\_Enabled\_without\_PPR  Only the test sequence #02 | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.4.2.3 | TC\_LPAd\_DeleteProfile\_Error\_with\_PPR1 | | LPAd | C012 | C012 | C012 | C012 |  |
| 5.4.4.2.4 | TC\_LPAd\_DeleteProfile\_Error\_Disabled\_with\_PPR2 | | LPAd | C013 | C013 | C013 | C013 |  |
| 5.4.4.2.5 | TC\_LPAd\_DeleteProfile\_Error\_Enabled\_with\_PPR2 | | LPAd | C014 | C014 | C014 | C014 |  |
| 5.4.4.2.6 | TC\_LPAd\_DeleteProfile\_Security\_Errors | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.5.2.1 | TC\_LPAd\_EnableProfile  Only the test sequence #01 | | LPAd | C009 | C009 | C009 | C009 |  |
| 5.4.5.2.1 | TC\_LPAd\_EnableProfile  Only the test sequence #02 | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.5.2.2 | TC\_LPAd\_EnableProfile\_ImplicitDisable  Only the test sequence #01 | | LPAd | C009 | C009 | C009 | C009 |  |
| 5.4.5.2.2 | TC\_LPAd\_EnableProfile\_ImplicitDisable  Only the test sequence #02 | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.5.2.3 | TC\_LPAd\_EnableProfile\_Error\_ProfileAlreadyEnabled | | LPAd | C010 | C010 | C010 | C010 |  |
| 5.4.5.2.4 | TC\_LPAd\_EnableProfile\_Error\_PPR1Set | | LPAd | C011 | C011 | C011 | C011 |  |
| 5.4.6.2.1 | TC\_LPAd\_DisableProfile | | LPAd | C060 | C060 | C060 | C060 |  |
| 5.4.6.2.2 | TC\_LPAd\_DisableProfile\_Error\_ProfileAlreadyDisabled | | LPAd | C017 | C017 | C017 | C017 |  |
| 5.4.6.2.3 | TC\_LPAd\_DisableProfile\_Error\_PPR1Set | | LPAd | C018 | C018 | C018 | C018 |  |
| 5.4.7.2.1 | TC\_LPAd\_RetrieveEID | | LPAd | C004 | C004 | C004 | C004 |  |
| 5.4.8.2.1 | TC\_LPAd\_eUICCMemoryReset  Only the test sequence #01 and test sequence #05 | | LPAd | C007 | C007 | C007 | C007 |  |
| 5.4.8.2.1 | TC\_LPAd\_eUICCMemoryReset  Only the test sequence #02 and test sequence #03 | | LPAd | C044 | C044 | C044 | C044 |  |
| 5.4.8.2.1 | TC\_LPAd\_eUICCMemoryReset  Only the test sequence #04 | | LPAd | C043 | C043 | C043 | C043 |  |
| 5.4.8.2.2 | TC\_LPAd\_eUICCMemoryResetWithPINVerification | | LPAd | C009 | C009 | C009 | C009 |  |
| 5.4.10.2.1 | TC\_LPAd\_Set/Edit Default SM-DP+ Address | | LPAd | C007 | C007 | C041 | C041 |  |
| 5.4.11.2.1 | TC\_LPAd\_DevicePowerOnProfileDiscovery\_SM-DP+\_address | | LPAd | C022 | C022 | C022 | C022 |  |
| 5.4.11.2.2 | TC\_LPAd\_DevicePowerOnProfileDiscovery\_SM-DS | | LPAd | C059 | C059 | C059 | C059 |  |
|  |  |
| 7.1 | TCA eUICC Profile Package Test Specification | | eUICC | M | M | M | M | See section 7.1 |

Table 5: Applicability of Tests

| Conditional item | Condition |
| --- | --- |
| C001 | IF (O\_E\_NIST) THEN M ELSE N/A |
| C002 | IF (O\_E\_BRP) THEN M ELSE N/A |
| C003 | IF (O\_E\_FRP) THEN M ELSE N/A |
| C004 | IF (O\_D\_LPAD) THEN M ELSE N/A |
| C005 | IF (O\_D\_LPAD AND O\_D\_CC\_RETRY AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C006 | IF (NOT O\_E\_LPAe) THEN M ELSE N/A |
| C007 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C008 | IF (O\_D\_LPAD AND O\_D\_EU\_POSTPONED AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C009 | IF (O\_D\_LPAD AND O\_D\_PIN AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C010 | IF (O\_D\_LPAD AND O\_D\_ENPROF) THEN M ELSE N/A |
| C011 | IF (O\_D\_LPAD AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_ALLOWS\_PPR1\_EUC\_REQ OR O\_D\_ALLOWS\_PPR1\_EUC\_NOT\_REQ) AND O\_D\_ENPREVPPR1 AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEM M ELSE N/A |
| C012 | IF (O\_D\_LPAD AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_ALLOWS\_PPR1\_EUC\_REQ OR O\_D\_ALLOWS\_PPR1\_EUC\_NOT\_REQ) AND O\_D\_DISDELPPR1 AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C013 | IF (O\_D\_LPAD AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_ALLOWS\_PPR2\_EUC\_REQ OR O\_D\_ALLOWS\_PPR2\_EUC\_NOT\_REQ) AND O\_D\_DELPPR2 AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C014 | IF (O\_D\_LPAD AND O\_D\_PIN AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_ALLOWS\_PPR2\_EUC\_REQ OR O\_D\_ALLOWS\_PPR2\_EUC\_NOT\_REQ) AND O\_D\_DELPPR2 AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C015 | IF (O\_P\_REUSE\_OTPK) THEN M ELSE N/A |
| C016 | IF (NOT O\_P\_REUSE\_OTPK) THEN M ELSE N/A |
| C017 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_DISPROF AND O\_D\_DISABLED\_SEPARATED) THEN M ELSE N/A |
| C018 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_ALLOWS\_PPR1\_EUC\_REQ OR O\_D\_ALLOWS\_PPR1\_EUC\_NOT\_REQ) AND O\_D\_DISPPR1 AND O\_D\_DISABLED\_SEPARATED) THEN M ELSE N/A |
| C019 | IF (O\_E\_REUSE\_OTPK) THEN M ELSE N/A |
| C020 | IF (NOT O\_E\_REUSE\_OTPK) THEN M ELSE N/A |
| C021 | IF (O\_S\_ALT) THEN M ELSE N/A |
| C022 | IF (O\_D\_LPAD AND O\_D\_POW\_ON\_PROF\_DISCOVERY AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND\_O\_D\_DEFAULT\_DP) THEN M ELSE N/A |
| C023 | IF (O\_D\_LPAD AND O\_D\_EU\_REJECT AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) THEN M ELSE N/A |
| C024 | IF (NOT O\_S\_ALT) THEN M ELSE N/A |
| C025 | IF (O\_E\_2\_PIR) THEN M ELSE N/A |
| C026 | IF ((O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) AND (O\_D\_EU\_POSTPONED OR O\_D\_EU\_REJECT)) THEN M ELSE N/A |
| C027 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_NICKNAME) THEN M ELSE N/A |
| C028 | IF (O\_P\_SESSION\_KEYS) THEN M ELSE N/A |
| C029 | IF (O\_P\_SESSION\_KEYS AND O\_P\_REUSE\_OTPK) THEN M ELSE N/A |
| C030 | IF (O\_P\_SESSION\_KEYS AND NOT O\_P\_REUSE\_OTPK) THEN M ELSE N/A |
| C031 | VOID |
| C032 | IF (O\_E\_CATBUSY\_EN\_DIS\_REFRESH) THEN M ELSE N/A |
| C033 | IF (NOT O\_E\_CATBUSY\_EN\_DIS\_REFRESH) THEN M ELSE N/A |
| C034 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_COMBINED) THEN M ELSE N/A |
| C035 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_SEPARATED) THEN M ELSE N/A |
| C036 | IF (O\_E\_CATBUSY\_EN\_DIS\_NOREFRESH) THEN M ELSE N/A |
| C037 | IF (NOT O\_E\_CATBUSY\_EN\_DIS\_NOREFRESH) THEN M ELSE N/A |
| C038 | IF (O\_E\_CATBUSY\_MR) THEN M ELSE N/A |
| C039 | IF (NOT O\_E\_CATBUSY\_MR) THEN M ELSE N/A |
| C040 | IF (O\_E\_INTEGRATED THEN M ELSE N/A |
| C041 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_DEFAULT\_DP\_ADDRESS) THEN M ELSE N/A |
| C042 | IF (O\_P\_ES2+\_RETRY) THEN M ELSE N/A |
| C043 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_NOT\_REQ)) THEN M ELSE N/A |
| C044 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_REQ OR O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_NOT\_REQ)) THEN M ELSE N/A |
| C045 | IF ((O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) AND (O\_D\_EU\_POSTPONED OR O\_D\_EU\_REJECT) AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ)) THEN M ELSE N/A |
| C046 | IF ((O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY) AND (O\_D\_EU\_POSTPONED OR O\_D\_EU\_REJECT) AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_REQ)) THEN M ELSE N/A |
| C047 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_SEPARATED AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ)) THEN M ELSE N/A |
| C048 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_SEPARATED AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_REQ)) THEN M ELSE N/A |
| C049 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_COMBINED AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ)) THEN M ELSE N/A |
| C050 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_COMBINED AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_REQ)) THEN M ELSE N/A |
| C051 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_SEPARATED AND O\_D\_ADDPREPPR1 AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_NOT\_REQ) THEN M ELSE N/A |
| C052 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_COMBINED AND O\_D\_ADDPREPPR1 AND (O\_D\_REMOVABLE\_DOWNLOAD\_PPR OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ OR O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_NOT\_REQ) THEN M ELSE N/A |
| C053 | IF (O\_P\_TLS BRP) THEN M ELSE N/A |
| C054 | IF (O\_S\_TLS BRP) THEN M ELSE N/A |
| C055 | IF (O\_S\_ALT AND O\_S\_TLS\_BRP) THEN M ELSE N/A |
| C056 | IF (NOT O\_S\_ALT AND O\_S\_TLS\_BRP) THEN M ELSE N/A |
| C057 | IF (O\_E\_SERVICES\_SPECIFIC\_DATA) THEN M ELSE N/A |
| C058 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_SMDS) THEN M ELSE N/A |
| C059 | IF (O\_D\_LPAD AND O\_D\_POW\_ON\_PROF\_DISCOVERY AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_SMDS) THEN M ELSE N/A |
| C060 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_DISABLE\_SEPARATED) THEN M ELSE N/A |
| C061 | VOID |
| C062 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_COMBINED AND O\_D\_DEFAULT\_DP) THEN M ELSE N/A |
| C063 | IF (O\_D\_LPAD AND NOT O\_D\_ONLY\_CELLULAR\_CONNECTIVITY AND O\_D\_ADD\_ENABLE\_SEPARATED AND\_O\_D\_DEFAULT\_DP) THEN M ELSE N/A |

Table 6: Conditional Items Referenced by Table 5

## General Consideration

This section contains some general considerations about the test cases defined in this document. Note that some external test specifications are referred to in chapter 7. Consequently, the following sub sections SHALL only apply for test cases defined in sections 4 and 5 and 6.

### Test Case Definition

Test descriptions are independent.

For each test described in this document, a chapter provides a general description of the initial conditions applicable for the whole test. This description is completed by specific configurations to each individual sub-case.

It is implicitly assumed that all entities under test SHALL be compliant with the initial states described in Annex G. An initial state SHALL be considered as a pre-requisite to execute all the test cases described in this Test Plan.

After completing the test, the configuration is reset before the execution of the following test.

### Test Cases Format

Here is an explanation of the way to define the test cases in chapters 4, 5 and 6.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **4.X.Y.Z Test Cases**  **4.X.Y.Z.1 TC\_IUT\_TestName1**   |  |  | | --- | --- | | General Initial Conditions | | | Entity | Description of the general initial condition | | Entity1 | Test case - general condition 1 | | Entity2 | Test case - general condition 2 |   **Test Sequence #01: Short Description**  Description of the aim of the test sequence N°1   |  |  | | --- | --- | | Initial Conditions |  | | Entity | Description of the initial condition | | Entity1 | Test sequence N°1 - initial condition 1 | | Entity2 | Test sequence N°1 - initial condition 2 |  | Step | Direction | Sequence / Description | Expected result | REQ | | --- | --- | --- | --- | --- | | IC1 | Entity1 → Entity2 | Command or Message to send from Entity1 to Entity2 | Expected result N°1.1 |  | | 1 | Entity1 → Entity2 | Command or Message to send from Entity1 to Entity2 | 1- expected result N°1.2  2- expected result N°1.3 | REQ1 | | 2 | Entity2 → Entity3 | Command or Message to send from Entity2 to Entity3 |  |  |   **Test Sequence #02**  Description of the aim of the test sequence N°2   | Step | Direction | Sequence / Description | Expected result | REQ | | --- | --- | --- | --- | --- | | 1 | Entity1 → Entity2 | Command or Message to send from Entity1 to Entity2 |  |  | | 2 | Entity2 → Entity3 | Command or Message to send from Entity2 to Entity3 | 1- expected result N°2.1  2- expected result N°2.2 | REQ2 |   **4.X.Y.Z.2 TC\_IUT\_TestName2**  … |

The test cases TC\_IUT\_TestName1 and TC\_IUT\_TestName2 are referenced in Table 5 that allows indicating the applicability of the tests.

In the test case TC\_IUT\_TestName1, the requirements REQ1 and REQ2 are respectively covered by the test sequences #01 and #02.

Note: For some test cases, requirements to be covered are not listed in the test sequences. In that case, references to sections in GSMA RSP Technical Specification [2] covered by the test sequences are indicated in the Conformance Requirements References section of the test case.

The test sequence #01 SHALL be executed if and only if these conditions are met:

* Test case - general condition 1
* Test case - general condition 2
* Test sequence N°1 - initial condition 1
* Test sequence N°1 - initial condition 2

The test sequence #02 SHALL be executed if and only if these conditions are met:

* Test case - general condition 1
* Test case - general condition 2

The tables defining the different initial conditions are optional.

Initial Conditions are intended to be reached dynamically using the Test Tool when possible.

No additional operation SHALL be done prior to the test sequence besides those indicated in the Initial Conditions (e.g. no other Profiles SHALL be present on the eUICC besides those defined in the Initial Conditions).

In the test sequence #01:

* the step IC1 corresponds to an additional Initial Condition
* in the step N°1, if the expected results N°1 and N°2 are validated, the requirement REQ1 (or a part of the REQ1) SHALL be considered as implemented

Note that all initial states (described in Annex G) SHALL be implemented by the entity under test whatever the test cases to execute.

In addition, following 2.2.1 sub sections present all information (e.g. Methods, Constants…) that MAY be referenced in test sequences.

After execution of each test sequence a clean-up procedure (CU) SHALL be executed to restore the IUT to the Common Initial State as defined in Annex G.

#### Methods and Procedures

A method is referenced as follow:

 MTD\_NAME\_OF\_THE\_METHOD(PARAM1, PARAM2…)

The key word “NO\_PARAM” SHALL be set in method call if the related optional parameter is not used.

All methods and their related parameters are described in Annex C.1.

A procedure is a generic sub-sequence and is referenced as follow:

* PROC\_NAME\_OF\_THE\_PROCEDURE

All procedures are described in Annex C.2.

The implementation of these methods and procedures is under the responsibility of the test tool providers.

#### Constants and Dynamic Content

A constant (e.g. text, ASN.1 structure, hexadecimal string, icon, URI, integer, EID, AID…) is referenced as follow:

 #NAME\_OF\_THE\_CONSTANT

All constants are defined in Annex A.

When provided as an ASN.1 value notation, a constant SHALL be encoded in DER TLV (as specified in ITU-T X.690 [16]) by the test tool.

A dynamic content (e.g. TLV, ASN.1 structure, signature, integer, AID, one-time key pair…) is referenced as follow:

 <NAME\_OF\_THE\_VARIABLE>

All dynamic contents are defined in Annex B.

A dynamic content is either generated by an IUT or by a test tool provider.

#### Requests and Responses

An ASN.1 or a JSON request is referenced as follow:

 #NAME\_OF\_THE\_REQUEST

An ASN.1 or a JSON response is referenced as follows:

 #R\_NAME\_OF\_THE\_RESPONSE

Each ASN.1 or JSON request and response MAY refer to a constant or a dynamic content. All these structures are defined in Annex D.

When provided as an ASN.1 value notation, a request or a response SHALL be encoded in DER TLV (as specified in ITU-T X.690 [16]) by the test tool.

When an ASN.1 element definition contains three points (i.e. “…”), it means that fields MAY be present but SHALL not be checked by the test tool.

In the following example, several fields MAY be part of the ProfileInfoListResponse but only the profileNickname SHALL be verified.

resp ProfileInfoListResponse ::=

profileInfoListOk :{

{   
 ...  
 profileNickname #NICKNAME  
 ...

}

}

This rule applies also for Constants definition.

Some ASN.1 SEQUENCE components have a DEFAULT value (for example, profileClass in StoreMetadataRequest). In this specification, when values are specified in ASN.1 syntax and the DEFAULT value is intended, two different formulations (both of which are valid) may be used:

* the relevant component is specified with the DEFAULT value;
* the relevant component is missing entirely.

These are logically equivalent and lead to the same DER encoding. In both cases, the following rules apply:

* When the test tool is sending the DER value, it SHALL NOT include the component (as per DER rules).
* When the test tool is checking a received DER value from the entity under test, it SHALL check that the component is NOT present.

Test tools SHALL consider two BIT STRINGs to be equivalent if the BIT STRINGs have the same DER encoding. For example, '0101'B shall be considered to be equivalent to '010100'B.

NOTE: this is equivalent to removing any trailing zero bits from the BIT STRINGs in "bstring" notation (e.g. '010100'B 🡪 '0101'B) and then comparing the strings textually.

NOTE: according to the DER format, the encoding of transmitted values will remove the trailing zeroes. The definition above allows for values which are specified using ASN.1 value notation and are not transmitted, such as values specified in the Annexes of the current document, including IUT settings which might be specified by a user of the current document and may contain trailing zeroes in the ASN.1 value notation.

#### APDUs

A C-APDU is referenced as follow:

 [NAME\_OF\_THE\_CAPDU]

All C-APDUs are defined in Annex D.4.

An R-APDU is referenced as follow:

 [R\_NAME\_OF\_THE\_RAPDU]

All R-APDUs are defined in Annex D.4.

Each APDU MAY refer to a constant or a dynamic content.

The APDU TERMINAL RESPONSE SHALL be dynamically generated by the test tool according to the related proactive command. Therefore, this particular command is not referenced with brackets in this specification. If not explicitly defined in the step, the general result SHALL be set by default to “Command performed successfully” (i.e. 0x83 01 00).

#### Profiles

In order to execute the test cases described in this document, Operational, Test and Provisioning Profiles are necessary. All these Profiles are defined in Annex E with the Profile Metadata content and the corresponding Profile Package as defined in the eUICC Profile Package Specification [4].

A Profile is referenced as follow:

 PROFILE\_OPERATIONALx with x the identifier of the Operational Profile

or

 PROFILE\_TESTx with x the identifier of the Test Profile

or

 PROFILE\_PROVISIONINGx with x the identifier of the Provisioning Profile

NOTE: Test Profiles and Provisioning Profiles are out of the scope of this version of test specification.

#### IUT Settings

For the purpose of some test cases, Device and eUICC manufacturers and Platforms (i.e. SM-DP+, SM-DS) providers need to give some information related to their products to the test tools providers (e.g. supported Java Card version).

An IUT setting is referenced as follow:

 #IUT\_NAME\_OF\_SETTING

All these settings are defined in Annex F.

#### Referenced Requirements

All requirements referenced in this document by their identifiers are present and described in Annex I. These requirements have been extracted from the specifications:

 GSMA RSP Technical Specification [2]

 GSMA RSP Architecture [3]

### General Rules for eUICC Testing

#### Default Profile Downloading process

By default, when an Operational Profile needs to be downloaded on the eUICC (e.g. As mentioned in an initial condition), the following rules apply except if it is differently defined in the Test Case.

The highest priority CI in euiccCiPKIdListForSigning SHALL be used.

In order to execute the Common Mutual Authentication procedure and the Sub-procedure Profile Download and Installation (End User Confirmation), the following requests SHALL be sent by the Test Tool:

 #GET\_EUICC\_INFO1 and #GET\_EUICC\_CHALLENGE

 #AUTH\_SMDP\_MATCH\_ID

* + - * with the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> set to the CI for signing indicated as highest priority in the #R\_EUICC\_INFO1
      * with the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same CI as the one chosen for signing
      * with the SM-DP+ address #TEST\_DP\_ADDRESS1

 #PREP\_DOWNLOAD\_NO\_CC

* + - * with the #CERT\_S\_SM\_DPpb\_ECDSA leading to the same CI as the one chosen for signing

 Neither ES10b.GetRAT nor ES10b.GetProfilesInfo requests SHALL be executed

During the Profile Installation, the following SCP03t TLVs SHALL be used by default:

 #S\_INIT\_SC\_PROF1

 #CONF\_ISDP\_EMPTY

 no TLV for "ES8+.ReplaceSessionKeys" function SHALL be used (i.e. the Profile SHALL be downloaded by using the session keys <S\_ENC> and <S\_MAC>)

#### Default Local Profile Management process

By default, when an Operational Profile needs to be enabled, disabled or deleted on the eUICC (e.g. As mentioned in an initial condition), the following rules apply except if it is differently defined in the Test Case.

The EnableProfileRequest and the DisableProfileRequest SHALL contain the following parameters:

 ICCID of the Profile to Enable or to Disable

 RefreshFlag set to TRUE

The eUICC SHALL send the REFRESH command in "UICC Reset" mode (i.e. the APDU[TERMINAL\_PROFILE] indicating the support "UICC Reset" SHALL be used by the Test Tool).

The DeleteProfileRequest SHALL contain the following parameter:

 ICCID of the Profile to Delete

#### ASN.1 elements verifications

Each time the eUICC returns an ASN.1 structure containing a SEQUENCE OF elements, the order of elements SHALL be checked by the Test Tool except for the particular responses:

 notificationMetadataList of ListNotificationResponse

 profileInfoListOk of ProfileInfoListResponse

 notificationList of RetrieveNotificationsListResponse

When an Operational Profile class is expected to be indicated in a ProfileInfoListResponse, the Test Tool SHALL accept only one DER encoding if the eUICC supports SGP.22 v2.2.x [2] or SGP.22 V2.2 [2b]: a tag 0x95 containing the integer value 2.

### General Rules for Device Testing

#### Default Profile Download and LPM Process on the Device Under Test

By default, when an Operational Profile needs to be downloaded, installed (and if necessary enabled) on the (Test) eUICC resided in the Device Under Test (e.g. As mentioned in an initial condition), the following rules apply except if it is defined differently in the Test Case.

The default way to execute the Profile download SHALL be the Add Profile procedure with Activation Code #ACTIVATION\_CODE\_1. The way to apply the Activation Code (manual typing or QR code scanning) depends on the Device/LPAd implementation. In order to execute the Common Mutual Authentication procedure and the Sub-procedure Profile Download and Installation (End User Confirmation), the following responses SHALL be sent by the S\_SM-DP+:

 #INITIATE\_AUTH\_OK

* + - * with the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> set to the CI for signing indicated as highest priority in euiccCiPKIdListForSigning in the #R\_EUICC\_INFO1
      * with the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same CI as the one chosen for signing
      * with the SM-DP+ address #TEST\_DP\_ADDRESS1

 #AUTH\_CLIENT\_OK

* + - * with the #CERT\_S\_SM\_DPpb\_ECDSA leading to the same CI as the one chosen for signing
      * Metadata of the downloaded Profile instead of #METADATA\_OP\_PROF1

 #GET\_BPP\_OK with the content of the installed Profile (no session keys used)

Before running a test sequence, and after establishing the Initial conditions, all pending Notifications (sent on the best-effort basis as soon as connectivity is available as defined in section 3.5 of SGP.22 [2]) SHALL have been acknowledged by the simulated SM-DP+(s). S\_SM-DP+(s) SHALL be run with suitable addresses in order to receive and acknowledge all pending Notifications (including install, enable, disable and delete). The addresses which are required depend on the server address used for recent profile downloads (typically #TEST\_DP\_ADDRESS1 to receive and acknowledge PIR), and the notificationAddress values in the Metadata of recently downloaded Profiles (for otherSignedNotification). Each S\_SM\_DP+ SHALL use the TLS certificate corresponding to its address (CERT\_S\_SM\_DP\_TLS, CERT\_S\_SM\_DP2\_TLS, etc).

If only O\_D\_ADD\_ENABLE\_COMBINED (or any other combined operation, like combined “disable and delete”) is supported by the DUT, the user might have to perform actions in a particular manner in order to achieve the initial conditions related to enabled/disabled state of profiles (for example: disable a profile after installing, install profiles in a particular order, re-enable an initial profile after installing a subsequent profile).

Some devices may always combine the “disable” procedure with a “delete” procedure. For such devices, further actions might be required to achieve the initial condition that a particular profile is disabled; in particular, this might be the case when the device supports only the combine “add and enable” procedure, and not the “add only” procedure. In this case, if neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, one of the following procedures is required (where the profile which needs to be disabled is denoted as Profile A):

* Install (and enable) Profile A; install another “helper” profile; enable the “helper” profile (this should automatically disable Profile A).
* Install (and enable) another “helper” profile; install Profile A (Profile A should remain disabled).

In some cases, the “helper” profile has to be deleted before the start of the actual test sequence to achieve the required state of the initial conditions.

If the test case requires a Profile Download to be initiated via SM-DS:

* The mechanism used to initiate this is device-specific.
* If the device is using Power-on Profile Discovery the following applies:
  + - * when it is supported, the value of the configuration parameter for Device Power-on Profile discovery is 'Enabled'.
      * the Device has to be powered-off and then powered-on before each test sequence.

#### LUI Settings and Result Verification Criteria

Some Initial Conditions require the “The protection of access to the LUI is disabled” setting. It means that no protection mechanism is enforced upon entry to the LUI as defined in SGP.22 [2].

The way to perform Authenticated or Strong Confirmation SHALL be executed by the S\_EndUser according to the description provided by the Device Vendor in #IUT\_LPAd\_Confirmation.

For operations for which SGP.21 [3] and SGP.22 [2] do not require Confirmation – i.e. only User Intent is required (for example, Enable Profile, Disable Profile, Set/Edit Nickname): if the Device requests Confirmation from the User, the Test Tool SHALL NOT treat this as a failure.

For operations for which SGP.21 [3] and SGP.22 [2] require Simple Confirmation: if the Device requests Authenticated or Strong Confirmation from the User, the Test Tool SHALL NOT treat this as a failure.

Some of the Expected Results on the IUT side expect “No Error”. In this case the Test Tool SHALL verify that there is no error message appears on the UI of the DUT.

The End User SHALL follow the LUI requests to successfully complete the Profile Download process. Any combined confirmation for consecutive Local Profile Management Operations SHALL be avoided by the End User unless it is explicitly required by the test procedure. E.g.: upon installation of a new Profile, the LPA could propose ‘add Profile’ and ‘enable’ into one single step with a single confirmation only (e.g. “Do you want to install Profile ‘ProfileName’ on your Device and enable it? Yes / No / Install only”) In this case the End User will select the confirmation only for the single actual operation (i.e. select “Install only”).

NOTE: When combined Add and Enable Profile operations are to be initiated, various device implementations are possible. Examples (non-exhaustive):

* The user initiates the Add Profile operation first, with the Enable operation being incorporated later in the process, for example, at the confirmation stage.
* The user initiates a composite "Add and Enable Profile" operation at the start of the process.

If a test sequence requires Add Profile initiation and only O\_D\_ADD\_ENABLE\_COMBINED is supported by the DUT, then Add Profile initiation SHALL be interpreted to mean that the combined Add and Enable Profile operations are to be initiated, taking into account the note above regarding various device implementations.

#### TLS Testing Recommendations

The TLS connection may be rejected either:

* by sending a TLS alert, or
* by closing of the TCP connection, though TLS handshake completed, or
* TLS handshake not completed without sending a TLS alert, or
* No further RSP communication has been initiated by LPAd on ES9+/ES11 within the #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT

Please note that this is not an exhaustive list, and acting as guidelines for the test tools.

### Pass Criteria

A test execution is considered as successful only if the test procedure was fully carried out successfully.

A test execution is considered as failed if the tested feature provides an unexpected behaviour.

A test execution is considered as inconclusive when the pass criteria cannot be evaluated due to issues during the setup of the initial conditions (including the ICx steps) or during the execution of steps in which no requirement is referenced.

### Future Study

Some of the test cases or test sequences described in this Test Plan are FFS (For Future Study). This MAY mean that some clarifications are expected at the requirement level to conclude on a test method. As consequence, the corresponding test SHALL not be executed.

### General Rules for SM-DP+ Testing

#### Default Profile Processing

By default, for ES2+ testing, the SM-DP+ SHALL use random keys to protect profiles.

# 3 Testing Architecture

## 3.1 Testing Scope

All the interfaces, intended to be tested in the scope of this document, are presented hereafter:



| Interface | Between | | Description |
| --- | --- | --- | --- |
| ES2+ | Operator | SM-DP+ | Used by the Operator to order Profiles for specific eUICCs as well as other administrative functions.  NOTE: this interface is out of scope of this specification. |
| ES6 | Operator | eUICC | Used by the Operator for the management of Operator services via OTA services. |
| ES8+ | SM-DP+ | eUICC | Provides a secure end-to-end channel between the SM-DP+ and the eUICC for the administration of the ISD-P and the associated Profile during download and installation. It provides Perfect Forward Secrecy. |
| ES9+ | SM-DP+ | LPD | Used to provide a secure transport between the SM-DP+ and the LPA (LPD) for the delivery of the Bound Profile Package and the delivery of Remote Profile Management Commands. |
| ES10a | LDSd | eUICC | Used between the LDSd and the LPA Services to handle a Profile discovery. |
| ES10b | LPDd | eUICC | Used between the LPDd and the LPA services to transfer a Bound Profile Package to the eUICC. This interface plays no role in the decryption of Profile Packages. |
| ES10c | LUId | eUICC | Used between the LUId and the LPA services for Local Profile Management by the End User. |
| ES11 | LDS | SM-DS | Used by the LDS to retrieve Event Records for the respective eUICC. |
| ES12 | SM-DP+ | SM-DS | Used by the SM-DP+ to issue or remove Event Registrations on the SM-DS. |
| ES15 | SM-DS | SM-DS | Used in the case of deployments of cascaded SM-DSs to connect those SM-DSs. |

Table 7: Interfaces Descriptions

## 3.2 Testing Execution

This chapter aims to describe the different testing environments and equipments to allow the test cases to be executed.

To permit the execution of the different test cases described in this Test Plan, specifics simulators SHALL be used. The simulators that have been defined are listed hereafter:

 S\_Device: the Device Simulator used to send some commands to the eUICC under test using ISO/IEC 7816-4 [7] on the contact interface

 S\_SM-DP+: the SM-DP+ Simulator

 S\_SM-DS: the SM-DS Simulator

 S\_MNO: the MNO Simulator

 S\_LPAd: the LPAd Simulator

 S\_LPAe: the LPAe Simulator

 S\_EndUser: the End User Simulator that acts as an End User. This simulator MAY be either a person (i.e. a Tester) or a software that simulates the End User interactions.

 S\_CLIENT: the HTTPs client Simulator for the purpose of TLS testing. The S\_CLIENT MAY be S\_SM-DP+, S\_SM-DS depending on the component under test.

 S\_SERVER: the HTTPs server Simulator for the purpose of TLS testing. The S\_SERVER MAY be S\_SM-DP+ or S\_SM-DS depending on the component under test.

 Implementation of these simulators remains under the responsibility of the test tool providers.

 The aim of all the test cases is to verify the compliance of an Actor/Component (i.e. eUICC, SM-DP+, Alternative SM‑DS, Root SM‑DS, LPAe, LPAd, Device).

Following notations are used:

 S\_ComponentName for a simulated component

 ComponentName for the Implementation Under Test (IUT)

 Where ComponentName is indicated by CLIENT, SERVER

 Depending on the component under test, the CLIENT MAY be the SM-DP+ or the SM-DS. The Operator component is currently out of scope.

* Depending on the component under test, the SERVER MAY be the SM-DP+ or the SM-DS. The Operator component is currently out of scope.
* The use of "-- optional" in any ASN.1 elements defined within this document indicate that the test tool SHALL allow for the value either being present with that value, or being absent.

### 3.2.1 eUICC - Test Environment

The following test environment is used for all eUICC test cases as defined in chapter 4.2 and 5.2 (unless it is specified differently in the specific test case). Following conditions apply:

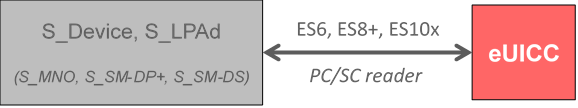
 Removable eUICC is used

 In the scope of this Test Plan, the eUICC SHALL support Java cardTM

 EUM SHALL provide products with one of the form factors specified in ETSI TS 102 221 [5]

 EUM SHALL provide products compliant with Annex G.2 – eUICC Initial States

 LPAd / MNO / SM-DP+ / SM-DS / Device Simulators SHALL be implemented by the test tools



The reference of this Test Environment is TE\_eUICC.

### 3.2.2 SM-DP+ and SM-DS - Test Environment

The following test environment is used for all SM-DP+ and SM-DS Interfaces related test cases as defined in chapter 4.3 and 4.5 (unless it is specified differently in the specific test case). Following conditions apply:

 SM-DS / SM-DP+ / LPA Simulators SHALL be implemented by the test tools

 Simulators act as a RSP server or a RSP client

 Definition of the TLS parameters/configuration is provided

 JSON (and ASN.1) input data are used (NOTE: ASN.1 format is out of scope of this specification)

#### 3.2.2.1 Test environment for SM-DP+ under test

Test Environment reference:

 TE\_P1 (SM-DP+ on ES12)

**SM-DP+**

***S\_SM-DS***

ES12

Test Environment reference:

 TE\_P2 (SM-DP+ on ES9+)

***S\_LPAd***

ES9+

**SM-DP+**

#### Test Environment reference:

 TE\_P3 (SM-DP+ on ES2+)

**SM-DP+**

***S\_MNO***

ES2+

***S\_LPAd***

ES9+

***S\_SM-DS***

ES12

#### 3.2.2.2 Test environment for SM-DS under test

Test Environment reference:

 TE\_S1 (SM-DS on ES11)

ES11

***S\_LPAd***

**SM-DS**

Test Environment reference:

 TE\_S2 (SM-DS on ES12)

ES12

**SM-DS**

***S\_SM-DP+***

Test Environment reference:

 TE\_S3 (SM-DS on ES12 and ES11)

ES11

ES12

***S\_LPAd***

**SM-DS**

***S\_SM-DP+***

Test Environment reference:

 TE\_SA1 (Alternative SM-DS on ES12 and ES15)

**SM-DS  
(alt)**

**(**

***S\_SM-DP+***

ES12

***S\_SM-DS  
(root)***

ES15

Test Environment reference:

* TE\_SA2 (Alternative SM-DS on ES12, ES15 and ES11)

**SM-DS  
(alt)**

**(**

***S\_SM-DP+***

ES12

***S\_SM-DS  
(root)***

ES15

***S\_LPAd***

ES11

Test Environment reference:

 TE\_SR1 (Root SM-DS on ES15)

ES15

***S\_SM-DS  
(alt)***

**SM-DS  
(root)**

**(**

Test Environment reference:

 TE\_SR2 (Root SM-DS on ES15 and ES11)

ES11

ES15

***S\_SM-DS  
(alt)***

***S\_LPAd***

**SM-DS  
(root)**

**(**

### 3.2.3 Device/LPAd - Test Environment

The following test environment is used for all LPAd Interfaces related test cases as defined in chapter 4.4 and 5.4 (unless it is specified differently in the specific test case). Following conditions apply:

 The Device contains an eUICC configured with Test Certificates and Test Keys

 The Test eUICC is either soldered or removable. In case the eUICC is removable, it SHALL NOT be removed during testing

 The Test eUICC is only used for LPAd testing and SHALL not be considered as an IUT

 The Test eUICC SHALL not support LPAe

 The Test eUICC SHOULD be compliant with the GSMA RSP Technical Specification [2]

 SM-DP+ Simulator(s) SHALL be implemented by the test tools

 SM-DS Simulator(s) SHALL be implemented by the test tools

 End User Simulator SHALL be used (S\_EndUser)

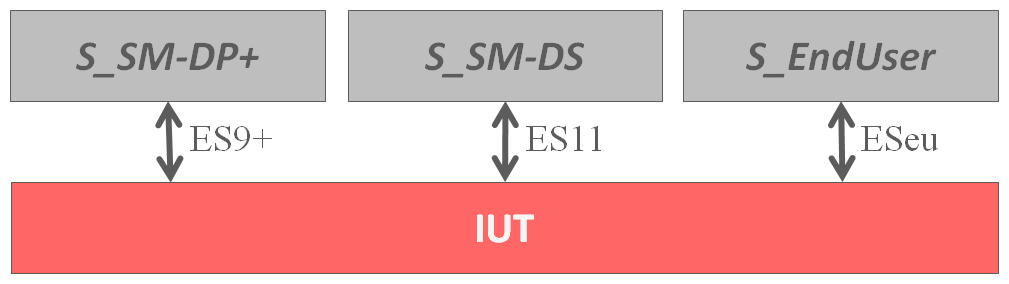
 No modification of the Device HW is required

 If the IUT is a Companion Device it has to be connected to a Primary Device as defined by the Device Vendor. The Device Vendor SHALL provide detailed information about which RSP functionality requires a Primary Device.

 No modification of the Device OS is required for the usage of S\_EndUser

 Test Root Certificate SHALL be configured in the Device

#### 3.2.3.1 General (Device/LPAd) Test Environment



The Test Environment consists of:

 IUT: Device, or Companion Device supporting the LPAd with a Test eUICC connected to a Primary Device

 S\_SM-DP+: a simulated SM-DP+ supporting a connection used by the Device to establish ES9+, (ES8+)

 S\_SM-DS: a simulated SM-DS supporting a connection used by the Device to establish ES11

 S\_EndUser

In case the Device supports a connection method different from Cellular Network it is expected that this connection method is used.

NOTE: Device that supports only Cellular Networks is out of scope for this specification.

#### 3.2.3.2 Device – Test Environment

If the IUT is a Device as defined in SGP21/SGP.22 [2] it SHALL provide at least one method to establish the IP connection to the S\_SM-DP+, or S\_SM-DS.

When executing a test case with an IUT matching this definition, default Initial States as defined in G.1.1 apply unless it is specified differently in the specific test case.

#### 3.2.3.3 Companion Device connected to a Primary Device – Test Environment

The Companion Device is connected to a Primary Device.

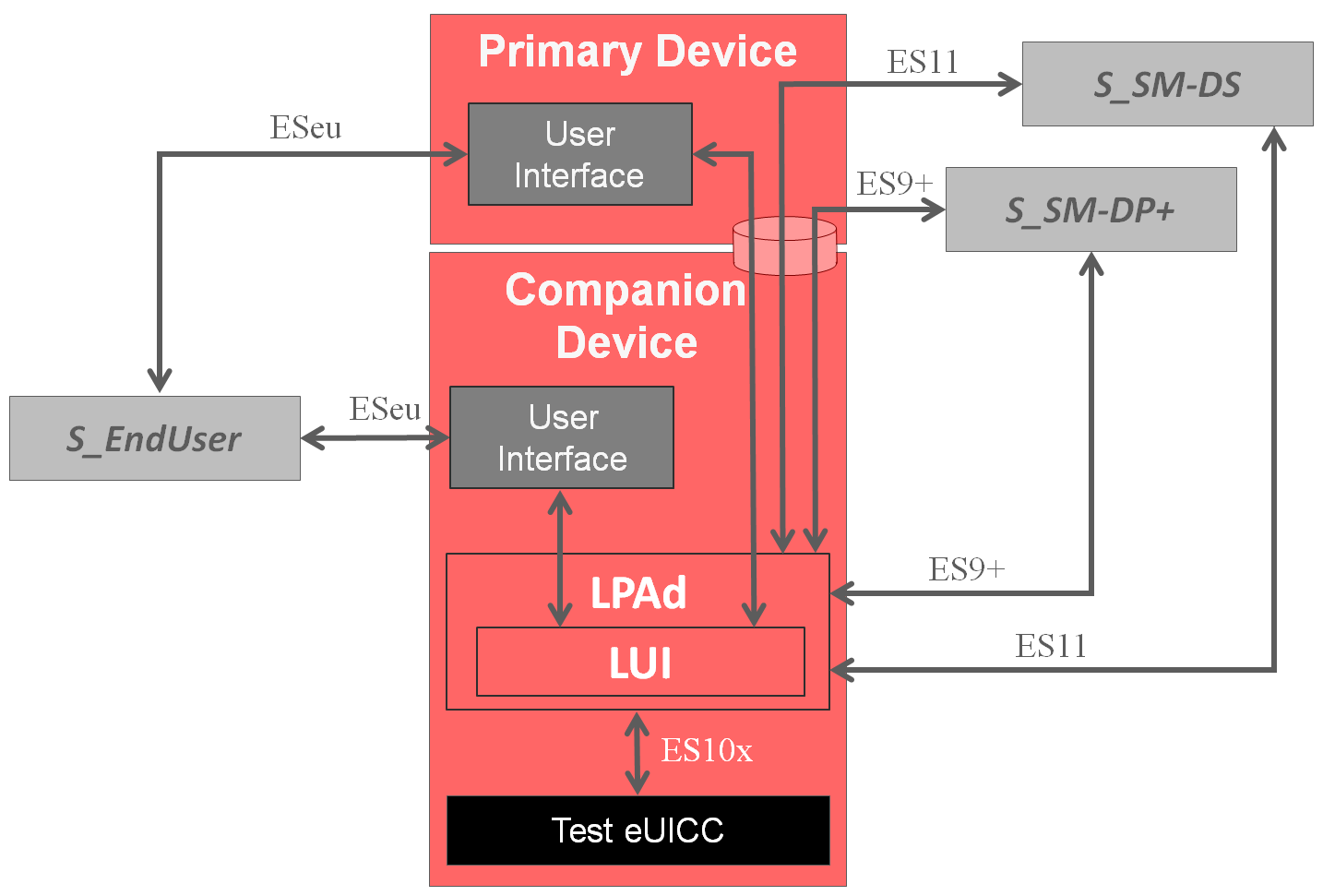
Device Vendors SHALL provide the mechanism to connect the Primary Device to the Companion Device.

User interaction and the verification of User Intents can be performed on the User Interface of the Primary Device or the companion Device.

The Companion Device MAY connect to the S\_SM-DP+, or S\_SM-DS directly, or MAY use a connection offered by the Primary Device.

To connect to the SM-DP+ or the SM-DS the Companion Device uses a connection offered by the Primary Device.

Initial State as defined in G.1.2 applies unless otherwise stated in the test case.



### 3.2.4 End-to-End Testing

The aim of all the test cases related to the system behaviour sections is to verify the functional behaviour of the RSP ecosystem composed of the following Actors:

 eUICC

 SM-DP+

 Device

 LPA

 SM-DS

This test environment is defined as FFS.

### 3.2.5 Integrated eUICC – Test Environment

The following test environment is used for all eUICC test cases as defined in chapter 4.2 and 5.2 (unless it is specified differently in the specific test case). Following conditions apply:

* EUM SHALL provide products compliant with Annex G.2 – eUICC Initial States
* LPAd / MNO / SM-DP+ / SM-DS / Device Simulators SHALL be implemented by the test tools
* Integrated eUICC shall provide a test interface which includes one of the following:
  + ISO/IEC 7816-4 [7]
  + USB CCID [29]
* For Integrated eUICC providing a USB CCID [29] test interface, the provisions of Annex J SHALL apply
* For Integrated eUICC providing ISO/IEC 7816-4 [7], the requirements of 3.2.1 eUICC – Test Environment with implementing shall apply



The reference of this [29] USB CCID based Test Environment is TE\_Integrated eUICC.

# 4 Interface Compliance Testing

## 4.1 General Overview

This section focuses on the implementation of the different interfaces according to the GSMA RSP Technical Specification [2]. The aim is to verify the compliance of all interfaces within the system.

## 4.2 eUICC Interfaces

### 4.2.1 ATR and ISD-R Selection

#### 4.2.1.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ34\_001

 RQ57\_001, RQ57\_003, RQ57\_005

 RQD0\_001

#### 4.2.1.2 Test Cases

##### 4.2.1.2.1 TC\_eUICC\_ATR\_And\_ISDR\_Selection

Test Sequence #01 Nominal: ATR and Select ISD-R

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_Device → eUICC | RESET | ATR present with the first tBi (i>2) after T = 15 containing b2=1 | RQ34\_001 |
| 2 | S\_Device → eUICC | [SELECT\_MF] | FCP Template present  SW=0x9000 |  |
| 3 | S\_Device → eUICC | [TERMINAL\_CAPABILITY\_LPAd] | SW=0x9000 |  |
| 4 | S\_Device → eUICC | [TERMINAL\_PROFILE] | Toolkit initialization THEN SW=0x9000 |  |
| 5 | S\_LPAd → eUICC | [MANAGE\_CHANNEL\_OPEN] | Extract the <CHANNEL\_NUMBER> from response data  SW=0x9000 | RQ57\_001 |
| 6 | S\_LPAd → eUICC | MTD\_SELECT(#ISD\_R\_AID) | The response data:  0x6F <L>   84 <L> #ISD\_R\_AID   A5 <L> <PROPRIETARY\_DATA>  #R\_ISDR\_SELECTION  SW=0x9000 | RQ57\_003 RQ57\_005 RQD0\_001 |

### 4.2.2 ES6 (Operator -- eUICC): UpdateMetadata

#### 4.2.2.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

3GPP TS 23.040 - Technical realization of the Short Message Service (SMS) [22]

**Requirements**

 RQ24\_021, RQ24\_024

 RQ29\_001, RQ29\_021

 RQ54\_001, RQ54\_002, RQ54\_003, RQ54\_004, RQ54\_005, RQ54\_006, RQ54\_007, RQ54\_008, RQ54\_009, RQ54\_010, RQ54\_011, RQ54\_012, RQ54\_013, RQ54\_014, RQ54\_013\_1, RQ54\_015, RQ54\_016

 RQ57\_120, RQ57\_122, RQ57\_123, RQ57\_126

#### 4.2.2.2 Test Cases

##### 4.2.2.2.1 TC\_eUICC\_ES6.UpdateMetadata

Throughout all the ES6.UpdateMetadata test cases, SMS is used as the secure OTA channel.

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 with #METADATA\_WITH\_PPRS\_AND\_ICON is loaded on the eUICC. |

Test Sequence #01 Nominal: Unset PPR1

The purpose of this test is to verify that the MNO can unset PPR1 from a Profile and that the eUICC can handle an Update Metadata request with only one field present.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #REMOVE\_PPR1,  FALSE)) | SW=0x91XX | RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_007 RQ54\_009 RQ54\_010 RQ54\_013\_1 RQ29\_021 RQ24\_021 RQ54\_011 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(0x9000) | RQ54\_015 RQ54\_011 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_GET\_UPDATE\_N1  SW=0x9000 | RQ54\_013\_1 RQ54\_009 RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #02 Nominal: Unset PPR2 and update icon

The purpose of this test is to verify that the MNO can unset PPR2 and update the icon and icon type values from a Profile.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #UPD\_ICON\_REM\_PPR2,  FALSE)) | SW=0x91XX | RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_007 RQ54\_009RQ54\_010 RQ54\_011 RQ54\_012 RQ54\_013\_1 RQ29\_021 RQ24\_021 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(0x9000) | RQ54\_015 RQ54\_011 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_GET\_UPDATE\_N2  SW=0x9000 | RQ54\_009RQ54\_012 RQ54\_013\_1 RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #03 Nominal: Unset PPR1 and PPR2 and update Profile name and Service Provider name

The purpose of this test is to verify that MNO can unset PPR1 and PPR2 from a Profile and can update the Service Provider Name and Profile Name values.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #UPD\_NAMES\_REM\_PPRS,  FALSE)) | SW=0x91XX | RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_007 RQ54\_009RQ54\_010 RQ54\_011 RQ54\_012 RQ54\_013\_1 RQ29\_021 RQ24\_021 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(0x9000) | RQ54\_015 RQ54\_011 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_GET\_UPDATE\_N3  SW=0x9000 | RQ54\_009 RQ54\_012 RQ54\_013\_1RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #04 Nominal: Delete PPRs, Service Provider and Profile names

The purpose of this test is to verify that the MNO can delete all PPRs, the Service Provider name and the Profile name from a Profile.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #REMOVE\_NAMES\_PPRS,  FALSE)) | SW=0x91XX | RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_007 RQ54\_009RQ54\_010 RQ54\_011 RQ54\_013 RQ54\_013\_1 RQ29\_021 RQ24\_021 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(0x9000) | RQ54\_015 RQ54\_011 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_GET\_UPDATE\_N4  SW=0x9000 | RQ54\_013 RQ54\_013\_1 RQ54\_009RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #05 Nominal: Delete icon

The purpose of this test is to verify that the MNO can delete the icon and icon type from a Profile.

This test case is defined as FFS and not applicable for this version of test specification.

Test Sequence #06 Nominal: Delete Unset PPRs

The purpose of this test is to verify that the MNO can delete already unset PPRs using the Update Metadata request.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #REMOVE\_NAMES\_PPRS,  FALSE)) | SW=0x91XX |  |
| IC3 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(0x9000) |  |
| IC4 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #UPD\_NAMES\_REM\_PPRS,  FALSE)) | SW=0x91XX | RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_007 RQ54\_009 RQ54\_010 RQ54\_011 RQ54\_013 RQ54\_015 RQ54\_013\_1 RQ29\_021 RQ24\_021 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(0x9000) |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_GET\_UPDATE\_N6  SW=0x9000 | RQ54\_013 RQ54\_013\_1 RQ54\_009 RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #07 Error: Set a pprUpdateControl value to one

The purpose of this test is to verify that the eUICC is correctly handling a pprUpdateControl value error from the MNO request, and return the expected error code status.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #UPD\_PPR\_CONTROL,  FALSE)) | SW=0x91XX | RQ24\_021 RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_010 RQ54\_011 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(0x6A81) | RQ54\_008 RQ54\_014 RQ54\_015 RQ54\_016 RQ54\_011 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_METADATA\_UNCHANGED  SW=0x9000 | RQ54\_014 RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #08 Error: Update Metadata on a Disable Profile

The purpose of this test is to verify that the eUICC is correctly rejecting an Update Metadata request from the MNO when the targeted Profile is Disabled.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #REMOVE\_PPR1,  FALSE)) | SW=0x91XX  or SW=0x9000 (i.e. envelope rejected, see NOTE)  or any error SW (i.e. envelope rejected, see NOTE) | RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_010 RQ54\_011 RQ24\_024 RQ24\_021 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | SMS POR received  SCP80 response status code equal to 0x06 (Unidentified security error) or 0x09 (TAR unknown) | RQ54\_011 RQ54\_014 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_METADATA\_UNCHANGED  SW=0x9000 | RQ54\_014 RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |
| NOTE: Depending on the implementation, the eUICC MAY decide to not send back a POR (i.e. SW=0x9000 on the ENVELOPE command). Therefore, the steps 2 and 3 SHALL only be executed in case SW=0x91XX. | | | | |

Test Sequence #09 Error: Empty request

The purpose of this test is to verify that the eUICC is correctly rejecting an Update Metadata request from the MNO when no field is present.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #UPD\_NO\_METADATA,  FALSE)) | SW=0x91XX | RQ24\_021 RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_010 RQ54\_011 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(<ANY\_SW\_IN\_ERROR>) | RQ54\_011 RQ54\_014 RQ54\_015 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_METADATA\_UNCHANGED  SW=0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 RQ54\_014 |

Test Sequence #10 Error: Update Icon without Icon Type field

The purpose of this test is to verify that the eUICC is correctly rejecting an Update Metadata request from the MNO when the icon field is present but not the icon type field.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #UPD\_ICON\_NO\_TYPE,  FALSE)) | SW=0x91XX | RQ24\_021 RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_010 RQ54\_011 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(<ANY\_SW\_IN\_ERROR>) | RQ54\_011 RQ54\_014 RQ54\_015 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_METADATA\_UNCHANGED  SW=0x9000 | RQ54\_014 RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #11 Error: Update Icon Type without Icon field

The purpose of this test is to verify that the eUICC is correctly rejecting an Update Metadata request from the MNO when the Icon Type field is present but not the Icon field.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [INSTALL\_PERSO\_RES\_ISDP];  MTD\_STORE\_DATA\_SCRIPT(  #UPD\_ICON\_TYPE\_ONLY,  FALSE)) | SW=0x91XX | RQ24\_021 RQ54\_001 RQ54\_002 RQ54\_003 RQ54\_004 RQ54\_005 RQ54\_006 RQ54\_010 RQ54\_011 |
| 2 | S\_Device 🡪eUICC | FETCH “XX” | MTD\_CHECK\_SMS\_POR(<ANY\_SW\_IN\_ERROR>) | RQ54\_011 RQ54\_014 RQ54\_015 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NEW\_METADATA) | #R\_METADATA\_UNCHANGED  SW=0x9000 | RQ54\_014 RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

### 4.2.3 ES8+ (SM-DP+ -- eUICC): InitialiseSecureChannel

#### 4.2.3.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ25\_024, RQ25\_025, RQ25\_026

 RQ31\_162, RQ31\_163

 RQ35\_003\_1

 RQ55\_011, RQ55\_012, RQ55\_013, RQ55\_014, RQ55\_015, RQ55\_019, RQ55\_023

 RQ57\_041\_1, RQ57\_013, RQ57\_016

#### 4.2.3.2 Test Cases

##### 4.2.3.2.1 TC\_eUICC\_ES8+.InitialiseSecureChannel

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Error: Invalid Remote Operation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #INIT\_SC\_INVALID\_OP\_ID,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands except the last one  SW=0x9000 with the response data #R\_PIR\_INVALID\_OP\_ID for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_162 RQ31\_163 RQ55\_012 RQ55\_015 RQ55\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ35\_003\_1 |

Test Sequence #02 Error: Invalid SM-DP+ Signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #INIT\_SC\_INVALID\_SIGN,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_INVALID\_SIGN for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_162 RQ31\_163 RQ55\_011 RQ55\_015 RQ25\_024 RQ25\_025 RQ25\_026 RQ35\_003\_1 |

Test Sequence #03 Error: Invalid Transaction Identifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #INIT\_SC\_INVALID\_TRANS\_ID,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | | SW=0x9000 without response data for all STORE DATA commands except the last one  SW=0x9000 with the response data #R\_PIR\_INVALID\_TRANS\_ID for the last STORE DATA command  The transactionId returned in the response SHALL not be checked (any value SHALL be accepted)  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_162 RQ31\_163 RQ55\_013 RQ55\_015 RQ25\_024 RQ25\_025 RQ25\_026 RQ35\_003\_1 |

Test Sequence #04 Error: Invalid CRT Values

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ | |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #INIT\_SC\_INVALID\_CRT,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for the intermediate STORE DATA commands (if any)  SW=0x9000 with the response data #R\_PIR\_INVALID\_CRT  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_162 RQ31\_163 RQ55\_014 RQ55\_015 RQ55\_019 RQ25\_024 RQ25\_025 RQ25\_026 RQ35\_003\_1 | |

Test Sequence #05 Error: InitialiseSecureChannel request while Secure Channel Session is ongoing

The purpose of this test is to ensure that the eUICC rejects an InitialiseSecureChannel request if a secure channel session is already ongoing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x6A88 or 0x6985  or  SW=0x9000 with a ProfileInstallationResult containing an ErrorResult | RQ55\_010 RQ57\_041\_1 RQ57\_013RQ57\_016 |

### 4.2.4 ES8+ (SM-DP+ -- eUICC): ConfigureISDP

#### 4.2.4.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_010

 RQ25\_023, RQ25\_024, RQ25\_025, RQ25\_026

 RQ31\_165

 RQ35\_003\_1

 RQ55\_025, RQ55\_026, RQ55\_027, RQ55\_028

#### 4.2.4.2 Test Cases

##### 4.2.4.2.1 TC\_eUICC\_ES8+.ConfigureISDP

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Nominal: Empty Proprietary Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_EMPTY,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   * <BPP\_SEG\_INIT> * <BPP\_SEG\_A0> * <BPP\_SEG\_A1> * <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_165 RQ55\_028 RQ24\_010 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA. | RQ25\_023 RQ25\_024 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_CONF\_OP\_PROF1) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  isdpAid <ISD\_P\_AID>  *-- dpProprietaryData SHALL not be*  *-- present*  }  }  SW=0x9000 | RQ55\_025 RQ24\_010 |

Test Sequence #02 Nominal: Proprietary Data with the maximum length authorized (i.e. 128 bytes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_MAX\_LENGTH,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_165 RQ55\_028 RQ24\_010 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA. | RQ25\_023 RQ25\_024 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_CONF\_OP\_PROF1) | #R\_CONF\_OP\_PROF1  SW=0x9000 | RQ55\_027 RQ24\_010 |

Test Sequence #03 Error: Proprietary Data with the maximum length exceeded (i.e. 129 bytes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_SIZE\_EXCEEDED,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands except the last one  SW=0x9000 with the response data #R\_PIR\_INVALID\_DATA for the last STORE DATA command | RQ55\_028 RQ31\_165 RQ55\_026 RQ25\_025 RQ25\_026 RQ35\_003\_1 |

### 4.2.5 ES8+ (SM-DP+ -- eUICC): StoreMetadata

#### 4.2.5.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_028

 RQ25\_017, RQ25\_023, RQ25\_024, RQ25\_025, RQ25\_026

 RQ29\_001, RQ29\_002

 RQ31\_166, RQ31\_167

 RQ32\_071

 RQ55\_029, RQ55\_030, RQ55\_031, RQ55\_032, RQ55\_033, RQ55\_034, RQ55\_035, RQ55\_036, RQ55\_037

 RQ57\_040

#### 4.2.5.2 Test Cases

##### 4.2.5.2.1 TC\_eUICC\_ES8+.StoreMetadata

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Nominal: All Metadata fields present (PNG icon used and PPR1 set)

The purpose of this test is to download the PROFILE\_OPERATIONAL1 by setting all Metadata fields. In this sequence, a PNG icon is used and PPR1 is set.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Operational Profile is present on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #FULL\_METADATA,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands expect the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_METADATA\_OP\_PROF1) | #R\_GET\_METADATA\_OP\_PROF1  SW=0x9000 | RQ32\_071 RQ29\_001 RQ29\_002 |

Test Sequence #02 Nominal: With JPG icon

The purpose of this case is to verify the ability to download JPG icon. The icon size does not allow for the command to fit into one data sequence.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_WITH\_JPG,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  iccid #ICCID\_OP\_PROF1,  iconType jpg,  icon #ICON\_JPG,  …  }  }  SW=0x9000 | RQ32\_071 |

Test Sequence #03 Nominal: Without providing Profile Class

The purpose of this test is to download the PROFILE\_OPERATIONAL1 by not indicating the Profile Class in the Metadata. In such a case, the default Profile Class 'Operational' SHALL be set by the eUICC.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_NO\_CLASS,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  iccid #ICCID\_OP\_PROF1,  profileClass operational  …  }  }  SW=0x9000 | RQ32\_071 RQ29\_001 RQ29\_002 |

Test Sequence #04 Nominal: With PPR2 set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_WITH\_PPR2,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PPR\_OP\_PROF1) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   iccid #ICCID\_OP\_PROF1,  profilePolicyRules {ppr2}  }  }  SW=0x9000 | RQ32\_071 RQ29\_001 RQ29\_002 |

Test Sequence #05 Nominal: With PPR1 and PPR2 set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Operational Profile is present on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_WITH\_PPR1\_PPR2,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PPR\_OP\_PROF1) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   iccid #ICCID\_OP\_PROF1,  profilePolicyRules {ppr1,ppr2}  }  }  SW=0x9000 | RQ32\_071 RQ29\_001 RQ29\_002 |

Test Sequence #06 Nominal: With several Notification events configured

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_WITH\_NOTIFS,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_NOTIF\_CONF\_OP\_PROF1) | #R\_GET\_PROF\_NOTIF\_CONF  SW=0x9000 | RQ32\_071 |

Test Sequence #07 Error: ICCID already present in the eUICC

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | General Initial Conditions do not apply. |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC | | | |
| IC2 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC3 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC4 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_ICCID\_ALREADY\_EXIST  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ25\_017 RQ31\_166 RQ55\_030 RQ55\_032 RQ25\_024 RQ25\_025 RQ25\_026 |

Test Sequence #08 Error: Profile Policy Rule is set but Profile Owner is not

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_PPR\_NO\_OWNER,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_METADATA\_INVALID (See NOTE)  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ31\_166 RQ55\_030 RQ55\_032 RQ25\_024 RQ25\_025 RQ25\_026 RQ25\_017 |
| NOTE: The errorReason "pprNotAllowed" or "installFailedDueToUnknownError" MAY be also returned by the eUICC. | | | | |

Test Sequence #09 Error: Profile Owner is set with a wildcard ('E') digits

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_WILDCARD,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_METADATA\_INVALID (See NOTE)  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ31\_166 RQ55\_030 RQ55\_032 RQ25\_024 RQ25\_025 RQ25\_026 RQ25\_017 |
| NOTE: The errorReason "pprNotAllowed" MAY be also returned by the eUICC. | | | | |

Test Sequence #10 Error: Icon Type is set but icon is not

The purpose of this test is to check ASN.1 conditional requirement for icon presence. If icon type is present then icon SHALL also be present.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the Euicc. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_WITHOUT\_ICON,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_METADATA\_INVALID  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ31\_166 RQ55\_030 RQ55\_032 RQ25\_024 RQ25\_025 RQ25\_026 RQ25\_017 |

##### 4.2.5.2.2 TC\_eUICC\_ES8+.StoreMetadata\_Service\_Specific\_Data

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Nominal: Metadata include service-specific data, stored

The purpose of this test is to download the PROFILE\_OPERATIONAL1 with service-specific metadata stored in the eUICC.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Operational Profile is present on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_SERVICE\_SPECIFIC\_STORED,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands expect the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_METADATA\_OP\_PROF1\_SERVICE\_SPECIFIC) | #R\_GET\_METADATA\_OP\_PROF1\_SERVICE\_SPECIFIC  SW=0x9000 | RQ32\_071 RQ29\_001 RQ29\_002 |

Test Sequence #02 Nominal: Metadata include service-specific data, not stored

The purpose of this test is to download the PROFILE\_OPERATIONAL1 with service-specific metadata *not* stored in the eUICC.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Operational Profile is present on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_SERVICE\_SPECIFIC\_NOT\_STORED,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands expect the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_METADATA\_OP\_PROF1\_SERVICE\_SPECIFIC) | #R\_GET\_METADATA\_OP\_PROF1  SW=0x9000 | RQ32\_071 RQ29\_001 RQ29\_002 |

Test Sequence #03 Nominal: Metadata include service-specific data, stored and not stored

The purpose of this test is to download the PROFILE\_OPERATIONAL1 with service-specific metadata stored in the eUICC and other service-specific metadata *not* stored.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Operational Profile is present on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_SERVICE\_SPECIFIC\_STORED\_AND\_NOT\_STORED,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ31\_166 RQ31\_167 RQ55\_029 RQ55\_031 RQ55\_033 RQ55\_035 RQ24\_028 RQ57\_040 RQ29\_001 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands expect the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_METADATA\_OP\_PROF1\_SERVICE\_SPECIFIC) | #R\_GET\_METADATA\_OP\_PROF1\_SERVICE\_SPECIFIC  SW=0x9000 | RQ32\_071 RQ29\_001 RQ29\_002 |

### 4.2.6 ES8+ (SM-DP+ -- eUICC): ReplaceSessionKeys

#### 4.2.6.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ25\_024, RQ25\_025, RQ25\_026

 RQ26\_021, RQ26\_022

 RQ31\_168

 RQ55\_038, RQ55\_041

#### 4.2.6.2 Test Cases

##### 4.2.6.2.1 TC\_eUICC\_ES8+.ReplaceSessionKeys

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Error: Incorrect PPK size

The purpose of this test is to verify that the eUICC checks that all PPK sizes are the same as session keys.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  #REPLACE\_S\_KEYS\_REQ\_INV\_SIZE,  #UPP\_OP\_PROF1)  MTD\_GENERATE\_BPP overriding:  For this test sequence, session keys SHALL be used for UPP SCP03t protection. Therefore:  Encrypt all <UPP\_SEG> with <S\_ENC>  Calculate and add a MAC to all tags 0x86 of sequenceOf86 by using <S\_MAC> | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A2>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A2>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_PPK\_INV  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ55\_038 RQ55\_041 RQ31\_168 RQ26\_021 RQ26\_022 RQ25\_024 RQ25\_025 RQ25\_026 |

### 4.2.7 ES8+ (SM-DP+ -- eUICC): LoadProfileElements

#### 4.2.7.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ25\_023, RQ25\_024, RQ25\_025, RQ25\_026

 RQ31\_173

 RQ32\_071

 RQ55\_045, RQ55\_045\_2, RQ55\_045\_3, RQ55\_047, RQ55\_048

 RQ57\_071, RQ57\_074

#### 4.2.7.2 Test Cases

##### 4.2.7.2.1 TC\_eUICC\_ES8+.LoadProfileElements

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Error: EFICCID different from the ICCID provided in the Profile Metadata

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL2 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_ICCID\_MISMATCH,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 with the response data #R\_PIR\_DATA\_MISMATCH  for one of the STORE DATA commands  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 RQ55\_045 RQ55\_048 RQ25\_025 RQ25\_026 RQ31\_173 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF2,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |

Test Sequence #02 Error: MCC / MNC of EFIMSI different from MCC / MNC of Profile Owner present in Metadata

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_MCCMNC\_MISMATCH,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 with the response data #R\_PIR\_DATA\_MISMATCH  for one of the STORE DATA commands  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 RQ55\_043 RQ55\_047 RQ55\_048 RQ25\_025 RQ25\_026 RQ31\_173 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_043 RQ55\_048 |

Test Sequence #03 Error: Session MAC chaining used instead of new Initial MAC chaining

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP (  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  #REPLACE\_S\_KEYS\_REQ,  #UPP\_OP\_PROF1)  MTD\_GENERATE\_BPP overriding:  For this test sequence, <S\_MAC\_CHAIN> SHALL be used instead of <PPK\_INIT\_MAC> for UPP SCP03t protection. | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A2>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC7 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A2>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 with the response data #R\_PIR\_SECU\_INVALID for one of the STORE DATA commands  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ55\_048 RQ31\_173 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |

Test Sequence #04 Error: S-MAC used instead of PPK-MAC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP (  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  #REPLACE\_S\_KEYS\_REQ,  #UPP\_OP\_PROF1)  MTD\_GENERATE\_BPP overriding:  For this test sequence <S\_MAC> SHALL be used instead of <PPK\_MAC> for UPP SCP03t protection. | | | |
| IC3 | Execute the step IC3 of the Test Sequence #03 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC7 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A2>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 with the response data #R\_PIR\_SECU\_INVALID for one of the STORE DATA commands  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ55\_048 RQ31\_173 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |

Test Sequence #05 Error: S-ENC used instead of PPK-ENC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP (  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  #REPLACE\_S\_KEYS\_REQ,  #UPP\_OP\_PROF1)  MTD\_GENERATE\_BPP overriding:  For this test sequence <S\_ENC> SHALL be used instead of <PPK\_ENC> for UPP SCP03t protection. | | | |
| IC3 | Execute the step IC3 of the Test Sequence #03 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC7 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A2>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 with the response data #R\_PIR\_SECU\_INVALID for one of the STORE DATA commands  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ55\_048 RQ31\_173 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |

Test Sequence #06 Error: Profile Downloading stopped by a Reset

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No pending Notification is present on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP (  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Execute the step IC3 of the Test Sequence #01 defined in this section | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except the last one.  Step 2 SHALL be triggered before sending the last STORE DATA | RQ25\_023 |
| 2 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 3 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |

Test Sequence #07 Nominal: ICCID in the 'ProfileHeader' PE is ignored by the eUICC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1  NO\_PARAM,  #UPP\_OP\_PROF1)  #UPP\_OP\_PROF1 overriding:  For this sequence, the *iccid* field SHALL be set to #ICCID\_OP\_PROF2 in the *ProfileHeader* element | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA. <ISD\_P\_AID> SHALL be in the range as defined SGP.02 [1]. | RQ25\_023 RQ25\_024 RQ55\_045 RQ55\_048 RQ25\_025 RQ25\_026 RQ55\_044 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  iccid #ICCID\_OP\_PROF1,  isdpAid <ISD\_P\_AID>,  profileState disabled,  …  }  }  SW=0x9000 | RQ32\_071 RQ55\_048 |

Test Sequence #08 Nominal: With gid1 and gid2 set

The purpose of this test is to verify that an Operational Profile configured with gid1 and gid2 can be downloaded on the eUICC.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL9 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF9,  NO\_PARAM,  #UPP\_OP\_PROF9) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK\_PROF9 for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023RQ25\_024RQ55\_045\_2 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_OWNERS) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  profileOwner {  mccMnc #MCC\_MNC9,  gid1 #GID1,  gid2 #GID2  }  }  }  SW=0x9000 | RQ32\_071 |

Test Sequence #09 Error: gid1 and gid2 provided in the Profile Metadata but not in the Profile Package

The purpose of this test is to verify that if gid1 and gid2 are provided in the Profile Metadata but ef-gid1 and ef-gid2 are not present and the related services (17 and 18) in ef-ust are not available, the eUICC returns an error during the LoadProfileElements process.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP1\_GID1GID2\_PRESENT,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 with the response data #R\_PIR\_DATA\_MISMATCH for one of the STORE DATA commands  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 RQ55\_045 RQ55\_048 RQ25\_025 RQ25\_026 RQ55\_045\_2 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |

Test Sequence #10 Error: gid1 and gid2 not provided in the Profile Metadata but present in Profile Package

The purpose of this test is to verify that if gid1 and gid2 are not provided in the Profile Metadata but ef-gid1 and ef-gid2 are present and the related services (17 and 18) in ef-ust are available, the eUICC returns an error during the LoadProfileElements process.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL9 is not loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP9\_GID1GID2\_MISSING,  NO\_PARAM,  #UPP\_OP\_PROF9) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 with the response data #R\_PIR\_DATA\_MISMATCH for one of the STORE DATA commands  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA | RQ25\_023 RQ25\_024 RQ55\_045 RQ55\_048 RQ25\_025 RQ25\_026 RQ55\_045\_3 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF9,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ32\_071 RQ55\_048 |

### 4.2.8 ES10a (LPA -- eUICC): GetEuiccConfiguredAddresses

#### 4.2.8.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_066

 RQ33\_021\_1

 RQ57\_017, RQ57\_018, RQ57\_019

#### 4.2.8.2 Test Cases

##### 4.2.8.2.1 TC\_eUICC\_ES10a.GetEuiccConfiguredAddresses

Test Sequence #01 Nominal: Only Root SM-DS Address

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Default SM-DP+ address has been set on the ISD-R. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS SW = 0x9000 | RQ57\_017 RQ57\_018 RQ57\_019 RQ33\_021\_1 |

Test Sequence #02 Nominal: Root SM-DS and Default SM-DP+ Addresses

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The ISD-R is provisioned with the Default SM-DP+ Address #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS\_DP\_1 SW = 0x9000 | RQ57\_017 RQ57\_018 RQ57\_019 RQ31\_066 RQ33\_021\_1 |

### 4.2.9 ES10a (LPA -- eUICC): SetDefaultDPAddress

#### 4.2.9.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ33\_021\_4, RQ33\_021\_5

 RQ57\_020, RQ57\_021, RQ57\_022, RQ57\_023, RQ57\_024

#### 4.2.9.2 Test Cases

##### 4.2.9.2.1 TC\_eUICC\_ES10a.SetDefaultDPAddress

Test Sequence #01 Nominal: Set SM-DP+ Address with Address Empty in eUICC

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No value is assigned to the Default SM-DP+ Address field. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #SET\_EUICC\_CONFIGURED\_ADDRESS\_1) | #R\_ES10a\_SD\_DP\_A\_OK SW = 0x9000 | RQ57\_020 RQ57\_021 RQ57\_023 RQ57\_024 RQ33\_021\_4 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS\_DP\_1 SW = 0x9000 | RQ57\_020 RQ57\_021 RQ57\_023 RQ57\_024 RQ33\_021\_5 |

Test Sequence #02 Nominal: Set SM-DP+ Address with SM-DP+ Address already in eUICC

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The SM-DP+ address #TEST\_DP\_ADDRESS1 is provisioned. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #SET\_EUICC\_CONFIGURED\_ADDRESS\_2) | #R\_ES10a\_SD\_DP\_A\_OK SW = 0x9000 | RQ57\_020 RQ57\_021 RQ57\_023 RQ57\_024 RQ33\_021\_4 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS\_DP\_2 SW = 0x9000 | RQ57\_020 RQ57\_021 RQ57\_023 RQ57\_024 RQ33\_021\_5 |

Test Sequence #03 Nominal: Set Empty SM-DP+ Address with SM-DP+ Address already in eUICC

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The SM-DP+ address #TEST\_DP\_ADDRESS1 is provisioned. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #SET\_EUICC\_CONFIGURED\_ADDRESS\_EMPTY) | #R\_ES10a\_SD\_DP\_A\_OK SW = 0x9000 | RQ57\_022 RQ57\_023 RQ57\_024 RQ33\_021\_4 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS SW = 0x9000 | RQ57\_022 RQ57\_023 RQ57\_024 RQ33\_021\_5 |

Test Sequence #04 Nominal: Set Empty SM-DP+ Address with Empty SM-DP+ Address in eUICC

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No value is assigned to the Default SM-DP+ Address field. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #SET\_EUICC\_CONFIGURED\_ADDRESS\_EMPTY) | #R\_ES10a\_SD\_DP\_A\_OK SW = 0x9000 | RQ57\_022 RQ57\_023 RQ57\_024 RQ33\_021\_4 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DSSW = 0x9000 | RQ57\_022 RQ57\_023 RQ57\_024 RQ33\_021\_5 |

### 4.2.10 ES10b (LPA -- eUICC): PrepareDownload

#### 4.2.10.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_011, RQ26\_029, RQ26\_030, RQ26\_034, RQ26\_035

 RQ31\_062, RQ31\_130, RQ31\_131, RQ31\_132, RQ31\_133, RQ31\_134, RQ31\_135, RQ31\_136, RQ31\_137, RQ31\_138, RQ31\_139, RQ31\_140, RQ31\_141

 RQ45\_006, RQ45\_026\_1, RQ45\_026, RQ45\_028, RQ45\_030

 RQ57\_025, RQ57\_026, RQ57\_027, RQ57\_028, RQ57\_029, RQ57\_030, RQ57\_031, RQ57\_033, RQ57\_034, RQ57\_035, RQ57\_036, RQ57\_037, RQ57\_038, RQ57\_039

#### 4.2.10.2 Test Cases

##### 4.2.10.2.1 TC\_eUICC\_ES10b.PrepareDownloadNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI based on NIST P-256 curve has been chosen for signing and for verification |

Test Sequence #01 Nominal: Without Confirmation Code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_NO\_CC) | #R\_PREP\_DOWNLOAD\_NO\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_NO\_CC. | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ26\_029 RQ26\_030 RQ26\_011 RQ26\_034 RQ26\_035 RQ31\_062 |

Test Sequence #02 Nominal: With Confirmation Code

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_WITH\_CC) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC. | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ26\_029 RQ26\_030 RQ26\_011 RQ26\_034 RQ26\_035 RQ31\_062 | |

Test Sequence #03 Nominal: With an unknown otPK.EUICC.ECKA

The purpose of this test is to verify that the eUICC does not use the one-time key pair given by the SM-DP+ when its value does not correspond to a stored one-time key pair. In this case, the eUICC SHALL generate a new set of key.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | | | |
| IC2 | S\_SM-DP+ generates a random <OTPK\_EUICC\_ECKA> | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT( #PREP\_DOWNLOAD\_RETRY\_CC) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <OTPK\_EUICC\_ECKA> present in the euiccSigned2 is not the same as in #PREP\_DOWNLOAD\_RETRY\_CC. | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ31\_138 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ57\_033 RQ26\_029 RQ26\_030 RQ26\_011 RQ26\_034 RQ26\_035 |

##### 4.2.10.2.2 TC\_eUICC\_ES10b.PrepareDownloadBRP

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI based on BrainpoolP256r1 curve has been chosen for signing and for verification |

Test Sequence #01 Nominal: Without Confirmation Code

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.2.10.2.1 – TC\_eUICC\_ES10b.PrepareDownloadNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #02 Nominal: With Confirmation Code

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.2.10.2.1 – TC\_eUICC\_ES10b.PrepareDownloadNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #03 Nominal: With an unknown otPK.EUICC.ECKA

This test sequence SHALL be the same as the Test Sequence #03 defined in section 4.2.10.2.1 – TC\_eUICC\_ES10b.PrepareDownloadNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

##### 4.2.10.2.3 TC\_eUICC\_ES10b.PrepareDownloadFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.2.10.2.4 TC\_eUICC\_ES10b.PrepareDownloadErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification |

Test Sequence #01 Error: No Hashed Confirmation Code but with Confirmation Code Required Flag set to TRUE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(  #CONFIRMATION\_CODE1,  <S\_TRANSACTION\_ID>) | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_INVALID\_CC) | SW different from 0x9000 without response data  or  SW=0x9000 with a response data containing a downloadResponseError | RQ31\_130 RQ31\_135 RQ31\_136 RQ57\_031 RQ57\_036 RQ57\_037 RQ57\_038 RQ31\_136 |

Test Sequence #02 Error: With incorrect CERT.DPpb.ECDSA (i.e. invalid signature)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_INV\_CERT) | #R\_PREP\_DOWNLOAD\_INV\_CERT  SW=0x9000  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #PREP\_DOWNLOAD\_INV\_CERT. | RQ31\_130 RQ31\_131 RQ31\_136 RQ57\_027 RQ57\_030 RQ57\_031 RQ57\_036 RQ57\_037 RQ57\_038 RQ31\_136 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 |

Test Sequence #03 Error: CERT.DPpb.ECDSA and CERT.DPauth.ECDSA not belonging to the same entity

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_CERT\_SMDP2) | #R\_PREP\_DOWNLOAD\_INV\_CERT  SW=0x9000  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #PREP\_DOWNLOAD\_CERT\_SMDP2. | RQ31\_130 RQ31\_132 RQ31\_136 RQ57\_029 RQ57\_031 RQ57\_036 RQ57\_037 RQ57\_038 RQ31\_136 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 |

Test Sequence #04 Error: With invalid SM-DP+ signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_INV\_SIGN) | #R\_PREP\_DOWNLOAD\_INV\_SIGN  SW=0x9000  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #PREP\_DOWNLOAD\_INV\_SIGN. | RQ31\_130 RQ31\_133 RQ31\_136 RQ57\_028 RQ57\_031 RQ57\_036 RQ57\_038 RQ31\_136 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 |

Test Sequence #05 Error: With invalid Transaction ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_INV\_TRANS\_ID) | #R\_PREP\_DOWN\_INV\_TRANS\_ID  SW=0x9000  The transactionId returned in the response SHALL not be checked (any value SHALL be accepted) | RQ31\_130 RQ31\_134 RQ31\_136 RQ57\_025 RQ57\_031 RQ57\_036 RQ57\_037 RQ57\_038 RQ31\_136 |

Test Sequence #06 Error: SM-DP+ has not been previously authenticated

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Common Mutual Authentication procedure has been executed between the eUICC and the S\_SM-DP+  (this condition overrides the last general initial condition defined in this test case). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the highest priority CI from <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> and choose #CERT\_S\_SM\_DPpb\_ECDSA according to this CI curve. |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT( #PREP\_DOWNLOAD\_NO\_AUTH) | #R\_PREP\_DOWN\_NO\_SESSION  SW=0x9000  The transactionId returned in the response SHALL not be checked (any value SHALL be accepted) | RQ31\_130 RQ31\_136 RQ57\_031 RQ57\_026 RQ57\_036 RQ57\_037 RQ57\_038 |

Test Sequence #07 Error: Unsupported curve

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT( #PREP\_DOWN\_INV\_CURVE) | #R\_PREP\_DOWN\_INV\_CURVE  SW=0x9000  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #PREP\_DOWN\_INV\_CURVE. | RQ31\_130 RQ31\_134 RQ31\_136 RQ57\_025 RQ57\_031 RQ57\_036 RQ57\_037 RQ57\_038 RQ31\_136 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 |

Test Sequence #08 Error: Invalid Certificate Role OID

The purpose of this sequence is to make sure that the eUICC refuses any SM-DP+ Certificate for Profile Package Binding that does not indicate “id-rspRole-dp-pb” in its extension for Certificate Policies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT( #PREP\_DOWNLOAD\_INV\_OID) | #R\_PREP\_DOWNLOAD\_INV\_CERT  SW=0x9000  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #PREP\_DOWNLOAD\_INV\_OID. | RQ31\_130 RQ31\_131 RQ31\_136 RQ57\_027 RQ57\_030 RQ57\_031 RQ57\_036 RQ57\_037 RQ57\_038 RQ31\_136 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 RQ45\_030 |

### 4.2.11 ES10b (LPA -- eUICC): LoadBoundProfilePackage

#### 4.2.11.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_010, RQ24\_028

 RQ25\_003, RQ25\_007, RQ25\_016, RQ25\_018, RQ25\_019, RQ25\_023, RQ25\_024

 RQ26\_011, RQ26\_012, RQ26\_013, RQ26\_016, RQ26\_018, RQ26\_019, RQ26\_020, RQ26\_021, RQ26\_022, RQ26\_029, RQ26\_034, RQ26\_035, RQ26\_036

 RQ31\_161, RQ31\_162, RQ31\_163, RQ31\_164, RQ31\_165, RQ31\_166, RQ31\_168, RQ31\_169, RQ31\_170, RQ31\_171, RQ31\_185, RQ31\_186\_1, RQ31\_188\_1

 RQ32\_070

 RQ35\_003\_1

 RQ44\_003

 RQ55\_001, RQ55\_002, RQ55\_003, RQ55\_006, RQ55\_007, RQ55\_008, RQ55\_016, RQ55\_017, RQ55\_018, RQ55\_020, RQ55\_021, RQ55\_022, RQ55\_024, RQ55\_025, RQ55\_028, RQ55\_033, RQ55\_036, RQ55\_037, RQ55\_039, RQ55\_040, RQ55\_041

 RQ57\_010, RQ57\_011, RQ57\_012, RQ57\_013, RQ57\_014, RQ57\_016, RQ57\_040, RQ57\_042, RQ57\_043, RQ57\_044, RQ57\_045

 RQD0\_001

 RQG0\_001, RQG0\_002, RQG0\_003, RQG0\_004, RQG0\_005, RQG0\_006

#### 4.2.11.2 Test Cases

##### 4.2.11.2.1 TC\_eUICC\_ES10b.LoadBoundProfilePackageNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI based on NIST P-256 curve has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Nominal: By using S-ENC and S-MAC

The purpose of this test is to download the PROFILE\_OPERATIONAL1 by using only the session S-ENC and S-MAC keys resulting from key agreement.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3>  NOTE: In this test sequence, the data resulting of this operation SHALL be composed of the following TLV arrays:   <BPP\_SEG\_INIT> contains the tag and length fields of the BoundProfilePackage TLV plus the #S\_INIT\_SC\_PROF1 command   <BPP\_SEG\_A0> contains the tag and length fields of the firstSequenceOf87 TLV plus the first 0x87 TLV containing #CONF\_ISDP\_PROF1 command   <BPP\_SEG\_A1> contains the tag and length fields of the sequenceOf88 TLV and each of the '88' TLVs containing #METADATA\_OP\_PROF1 command   <BPP\_SEG\_A3> contains the tag and length fields of the sequenceOf86 TLV and each of the '86' TLVs containing #UPP\_OP\_PROF1 protected with <S\_ENC> and <S\_MAC> | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_162 RQ31\_163 RQ31\_164 RQ55\_003 RQ55\_016 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_024 RQ26\_011 RQ26\_013 RQ26\_016 RQ26\_034 RQ26\_035 RQ31\_161 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_165 RQ55\_028 RQ26\_012 RQ26\_013 RQ26\_016 RQ26\_018 RQ26\_019 RQ26\_036 RQ31\_161 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_166 RQ55\_033 RQ55\_036 RQ55\_037 RQ24\_028 RQ26\_012 RQ26\_013 RQ26\_016 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_036 RQ31\_161 RQG0\_005 RQG0\_006 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA. <ISD\_P\_AID> SHALL be in the range as defined SGP.02 [1]. | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_170 RQ31\_171 RQ57\_045 RQ55\_008 RQ25\_003 RQ25\_007 RQ25\_018 RQ25\_019 RQ25\_023 RQ25\_024 RQ55\_025 RQ25\_016 RQ26\_012 RQ26\_013 RQ26\_016 RQ26\_018 RQ26\_019 RQ26\_034 RQ26\_035 RQ26\_036 RQ31\_161 RQ35\_003\_1 RQ44\_003 RQD0\_001 RQG0\_005 RQG0\_006 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  iccid #ICCID\_OP\_PROF1,  isdpAid <ISD\_P\_AID>,  profileState disabled,  …  }  }  SW=0x9000 | RQ32\_070 RQ55\_025 RQ24\_010 RQ26\_020 RQ31\_161 RQD0\_001 |

Test Sequence #02 Nominal: By using PPK-ENC and PPK-MAC

The purpose of this test is to download the PROFILE\_OPERATIONAL1 by using a new set of random session keys: PPK-ENC, PPK-MAC and Initial MAC chaining value.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  #REPLACE\_S\_KEYS\_REQ,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A2>   <BPP\_SEG\_A3>  NOTE: In this test sequence, the data resulting of this operation SHALL be composed of the following TLV arrays:   <BPP\_SEG\_INIT> contains the tag and length fields of the BoundProfilePackage TLV plus the #S\_INIT\_SC\_PROF1 command   <BPP\_SEG\_A0> contains the tag and length fields of the firstSequenceOf87 TLV plus the first 0x87 TLV containing #CONF\_ISDP\_PROF1 command   <BPP\_SEG\_A1> contains the tag and length fields of the sequenceOf88 TLV and each of the '88' TLVs containing #METADATA\_OP\_PROF1 command   <BPP\_SEG\_A2> contains the tag and length fields of the secondSequenceOf87 TLV plus the first '87' TLV, containing the #REPLACE\_S\_KEYS\_REQ command   <BPP\_SEG\_A3> contains the tag and length fields of the sequenceOf86 TLV and each of the '86' TLVs containing #UPP\_OP\_PROF1 protected with PPK-ENC and PPK-MAC | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_162 RQ31\_163 RQ31\_164 RQ55\_003 RQ55\_016 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_024 RQ26\_011 RQ26\_013 RQ26\_016 RQ26\_034 RQ26\_035 RQ31\_161 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_165 RQ55\_028 RQ26\_012 RQ26\_013 RQ26\_016 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_036 RQ31\_161 RQ26\_011 RQ26\_013 RQ26\_016 RQ26\_034 RQ26\_035 RQ31\_161 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_166 RQ55\_033 RQ55\_036 RQ55\_037 RQ26\_012 RQ26\_013 RQ26\_016 RQ26\_018 RQ26\_019 RQ26\_036 RQ31\_161 RQG0\_005 RQG0\_006 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A2>) | SW=0x9000 without response data for all STORE DATA commands | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_168 RQ31\_169 RQ55\_039 RQ55\_040 RQ55\_041 RQ26\_021 RQ26\_022 RQ26\_012 RQ26\_013 RQ26\_016 RQ26\_018 RQ26\_019 RQ26\_036 RQ31\_161 RQG0\_005 RQG0\_006 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK  for the last STORE DATA command  The euiccSignPIR SHALL be verified with the #PK\_EUICC\_ECDSA. <ISD\_P\_AID> SHALL be in the range as defined SGP.02 [1]. | RQ57\_040 RQ57\_042 RQ57\_043 RQ57\_044 RQ55\_001 RQ55\_002 RQ55\_006 RQ55\_007 RQ57\_010 RQ57\_011 RQ57\_012 RQ57\_014 RQ26\_029 RQ31\_170 RQ31\_171 RQ57\_045 RQ55\_008 RQ25\_003 RQ25\_007 RQ25\_018 RQ25\_019 RQ25\_023 RQ25\_024 RQ55\_025 RQ25\_016 RQ26\_012 RQ26\_013 RQ26\_034 RQ26\_035 RQ31\_161 RQ35\_003\_1 RQ44\_003 RQD0\_001 RQG0\_005 RQG0\_006 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  iccid #ICCID\_OP\_PROF1,  isdpAid <ISD\_P\_AID>,  profileState disabled,  …  }  }  SW=0x9000 | RQ55\_025 RQ32\_070 RQ24\_010 RQ26\_020 RQ31\_161 RQD0\_001 |

##### 4.2.11.2.2 TC\_eUICC\_ES10b.LoadBoundProfilePackageBRP

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI based on BrainpoolP256r1 curve has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Nominal: By using S-ENC and S-MAC

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.2.11.2.1 – TC\_eUICC\_ES10b.LoadBoundProfilePackageNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #02 Nominal: By using PPK-ENC and PPK-MAC

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.2.11.2.1 – TC\_eUICC\_ES10b. LoadBoundProfilePackageNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

##### 4.2.11.2.3 TC\_eUICC\_ES10b.LoadBoundProfilePackageFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.2.11.2.4 TC\_eUICC\_ES10b.LoadBoundProfilePackage\_ErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification   Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

Test Sequence #01 Error: Unrecognized leading tag in BPP

The purpose of this test is to ensure that the eUICC rejects any BPP segment with an unrecognized leading tag during Profile download. In such case, the eUICC SHALL return a SW of 0x6A88 and SHALL not discard the download session state.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | | Direction | | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | | | |
| IC4 | S\_LPAd → eUICC | | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | | SW=0x9000 without response data for all STORE DATA commands | RQ31\_186\_1 |
| 1 | S\_LPAd → eUICC | | MTD\_STORE\_DATA\_SCRIPT(  #UNKNOWN\_BPP\_SEGMENT) | | SW=0x6A88 | RQ31\_186\_1 RQ57\_013 RQ57\_016 |
| 2 | S\_LPAd → eUICC | | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | | SW=0x9000 without response data for all STORE DATA commands | RQ31\_186\_1 |
| 3 | S\_LPAd → eUICC | | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | | SW=0x9000 without response data for all STORE DATA commands except for the last one  SW=0x9000 with the response data #R\_PIR\_OK for the last STORE DATA command | RQ31\_186\_1 |
| 4 | S\_LPAd → eUICC | | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  iccid #ICCID\_OP\_PROF1,  isdpAid <ISD\_P\_AID>,  profileState disabled,  …  }  }  SW=0x9000 |  |

Test Sequence #02 Error: GetEUICCChallenge during BPP loading

The purpose of this test is to ensure that the eUICC accepts an ES10b.GetEUICCChallenge request indicating the start of a new RSP session while a BPP is loaded.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Notification is stored in the eUICC's Pending Notifications List. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW=0x9000 | RQ31\_188\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A3>) | SW=0x6A88 or 0x6985 | RQ31\_185 RQ57\_013 RQ57\_016 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{}  SW=0x9000 | RQ31\_185 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ31\_185 |

### 4.2.12 ES10b (LPA -- eUICC): GetEUICCChallenge

#### 4.2.12.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_029, RQ31\_030, RQ31\_031

 RQ57\_048, RQ57\_049, RQ57\_050

#### 4.2.12.2 Test Cases

##### 4.2.12.2.1 TC\_eUICC\_ES10b.GetEUICCChallenge

Test Sequence #01 Nominal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000 | RQ31\_029 RQ31\_030 RQ31\_031 RQ57\_049 RQ57\_050 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  <EUICC\_CHALLENGE> received in this step is different to the <EUICC\_CHALLENGE> in Step 1 | RQ57\_048 |

### 4.2.13 ES10b (LPA -- eUICC): GetEUICCInfo

#### 4.2.13.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_027, RQ31\_028, RQ31\_053, RQ31\_060

 RQ43\_001, RQ43\_002, RQ43\_003, RQ43\_004, RQ43\_005, RQ43\_006, RQ43\_007, RQ43\_008, RQ43\_009, RQ43\_010, RQ43\_011, RQ43\_012, RQ43\_013

 RQ57\_051, RQ57\_052, RQ57\_053, RQ57\_054, RQ57\_057\_1, RQ57\_058, RQ57\_059, RQ57\_060, RQ57\_062, RQ57\_061, RQ57\_063, RQ57\_064, RQ57\_066

#### 4.2.13.2 Test Cases

##### 4.2.13.2.1 TC\_eUICC\_ES10b.GetEUICCInfo1

Test Sequence #01 Nominal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 | RQ31\_027 RQ31\_028 RQ57\_051 RQ57\_052 RQ57\_054 |

Test Sequence #02 Nominal: GetEUICCInfo call after GetEUICCChallenge

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 | RQ31\_027 RQ31\_028 RQ57\_051 RQ57\_052 RQ57\_054 |

Test Sequence #03 Nominal: GetEUICCInfo1 call after AuthenticateServer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTHENTICATE\_SMDP) | #R\_AUTHENTICATE\_SMDP  SW = 0x9000 |  |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 | RQ57\_051 |

##### 4.2.13.2.2 VOID

##### 4.2.13.2.3 TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_V2.2.x

Test Sequence #01 Nominal – RSP Version 2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | #R\_EUICC\_INFO2  Verify if:   <EXT\_CARD\_RESOURCE> contains a “number of installed application” value field set to '00'   #IUT\_TS102241\_VERSION is equal to 0x090000 or higher  #IUT\_GLOBALPLATFORM\_VERSION is equal to 0x020300 or higher  <EUICC\_RSP\_CAPABILITY> contains  o crlSupport set to '0' if O\_E\_CRL is not supported  (otherwise, it SHALL be set to '1')  o testProfileSupport set to '0' if O\_E\_TEST\_PROF is not supported  (otherwise, it SHALL be set to '1')  o rpmSupport set to '0'  o additionalProfile set to '1'   #IUT\_UICC\_CAPABILITY contains  o javacard and akaMilenage set to '1'  o Either akaTuak128 or akaTuak256 set to '1'  SW = 0x9000 | RQ43\_001 RQ43\_002 RQ43\_005 RQ43\_006 RQ43\_007 RQ43\_008 RQ43\_009 RQ43\_010 RQ43\_011 RQ43\_012 RQ43\_013 RQ57\_057\_1 RQ57\_060 RQ57\_061 RQ57\_063 RQ57\_064 RQ57\_066 |

##### 4.2.13.2.4 TC\_eUICC\_ES10b.GetEUICCInfo2

Test Sequence #01 Nominal: GetEUICCInfo2 call after AuthenticateServer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | | |
| IC3 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID | |  | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> | |  | |
| IC5 | The following inputs are required for Step IC6 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | | | |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTHENTICATE\_SMDP) | #R\_AUTHENTICATE\_SMDP  SW = 0x9000 |  | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | same EUICCInfo2 data object as in Step IC6 (the extCardResource field SHALL be excluded from the comparison)  SW = 0x9000 | RQ57\_051 RQ57\_053 RQ57\_054 RQ57\_058 RQ57\_059 RQ57\_062 RQ31\_053 RQ31\_060 RQ43\_001 RQ43\_002 | |

##### 4.2.13.2.5 TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_V2.3

Test Sequence #01 Nominal – RSP Version 2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | #R\_EUICC\_INFO2  Verify if:   <EXT\_CARD\_RESOURCE> contains a “number of installed application” value field set to ‘00’   #IUT\_TS102241\_VERSION is equal to 0x090000 or higher  #IUT\_GLOBALPLATFORM\_VERSION is equal to 0x020300 or higher  <EUICC\_RSP\_CAPABILITY> contains  o crlSupport set to ‘0’ if O\_E\_CRL is not supported  (otherwise, it SHALL be set to ‘1’)  o testProfileSupport set to ‘0’ if O\_E\_TEST\_PROF is not supported  (otherwise, it SHALL be set to ‘1’)  o rpmSupport set to ‘0’  o additionalProfile set to ‘1’   #IUT\_UICC\_CAPABILITY contains  o  javacard and akaMilenage set to ‘1’  o  Either akaTuak128 or akaTuak256 set to ‘1’   * If the treProperties field is present, the value is equal to one of the following values:   o { isDiscrete }  o { isIntegrated }  o { isIntegrated, usesRemoteMemory }  SW = 0x9000 | RQ43\_001 RQ43\_002 RQ43\_005 RQ43\_006 RQ43\_007 RQ43\_008 RQ43\_009 RQ43\_010 RQ43\_011 RQ43\_012 RQ43\_013 RQ57\_057\_1 RQ57\_060 RQ57\_061 RQ57\_063 RQ57\_064 RQ57\_066 |

##### 4.2.13.2.6 TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_Integrated\_eUICC

Test Sequence #01 Nominal

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | #R\_EUICC\_INFO2  Verify that:   * The treProperties field contains   + isIntegrated set to ‘1’   + isDiscrete set to ‘0’ * The treProductReference field is present and not empty   SW = 0x9000  NOTE: usesRemoteMemory can be set to either '0' or '1'. |

##### 4.2.13.2.7 TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_V2.4\_or\_Higher

Test Sequence #01 Nominal – RSP Version 2.4 or Higher

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | #R\_EUICC\_INFO2  Verify if:   <EXT\_CARD\_RESOURCE> contains a “number of installed application” value field set to ‘00’   #IUT\_TS102241\_VERSION is equal to 0x090000 or higher  #IUT\_GLOBALPLATFORM\_VERSION is equal to 0x020300 or higher   * <EUICC\_RSP\_CAPABILITY> contains   o crlSupport set to ‘0’ if O\_E\_CRL is not supported  (otherwise, it SHALL be set to ‘1’)  o testProfileSupport set to ‘0’ if O\_E\_TEST\_PROF is not supported  (otherwise, it SHALL be set to ‘1’)  o rpmSupport set to ‘0’  o additionalProfile set to ‘1’   * serviceSpecificDataSupport set to '0' if O\_E\_SERVICES\_SPECIFIC\_DATA is not supported  (otherwise, it SHALL be set to '1')    #IUT\_UICC\_CAPABILITY contains  o  javacard and akaMilenage set to ‘1’  o  Either akaTuak128 or akaTuak256 set to ‘1’   * If the treProperties field is present, the value is equal to one of the following values:   o { isDiscrete }  o { isIntegrated }  o { isIntegrated, usesRemoteMemory }  SW = 0x9000 | RQ43\_001 RQ43\_002 RQ43\_005 RQ43\_006 RQ43\_007 RQ43\_008 RQ43\_009 RQ43\_010 RQ43\_011 RQ43\_012 RQ43\_013 RQ57\_057\_1 RQ57\_060 RQ57\_061 RQ57\_063 RQ57\_064 RQ57\_066 |

##### 4.2.13.2.8 TC\_eUICC\_ES10b.GetEUICCInfo\_SVN

The purpose of this test is to verify the SVN values returned by the eUICC.

Test Sequence #01 Nominal – SVN as per declared IUT setting

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Verify that <ANY\_SVN> is equal to #RSP\_SVN\_H.Verify that <ANY\_SVN> is one of the versions of SGP.22 for which this specification contains test cases, as specified in section 1.2. | RQ43\_004 |

Test Sequence #02 Nominal – SVN value consistent across ES10b calls

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 1 | S\_LPAd → eUICC | [MANAGE\_CHANNEL\_OPEN] | Extract the <CHANNEL\_NUMBER> from response data  SW=0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_SELECT(#ISD\_R\_AID) | The response data:  0x6F <L>   84 <L> #ISD\_R\_AID   A5 <L> <PROPRIETARY\_DATA>  #R\_ISDR\_SELECTION  SW=0x9000  Extract the value <ANY\_SVN> of the svn field in the response. | RQ57\_003 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Verify that <ANY\_SVN> in the response is equal to the value <ANY\_SVN> extracted in step 2. | RQ43\_004 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | #R\_EUICC\_INFO2  SW = 0x9000  Verify that <ANY\_SVN> in the response is equal to the value <ANY\_SVN> extracted in step 2 | RQ43\_004 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 6 | The following inputs are required for Step 7 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   * Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTHENTICATE\_SMDP) | #R\_AUTHENTICATE\_SMDP  SW = 0x9000  Verify that <ANY\_SVN> is equal to the value <ANY\_SVN> extracted in step 2. | RQ43\_004 |

##### 4.2.13.2.9 TC\_eUICC\_ES10b.GetEUICCInfo\_profileVersion

The purpose of this test is to verify the profileVersion values returned by the eUICC.Test Sequence #01 Nominal – profileVersion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | #R\_EUICC\_INFO2  Verify that <ANY\_PROFILE\_VERSION> is equal to #IUT\_SIMA\_VERSION.  Verify that <ANY\_PROFILE\_VERSION> encodes one of the values listed in section 7.1 in the “eUICC Profile Package Specification versions required for the given SGP.22 version” column for the SGP.22 version supported by the eUICC.  If the SGP.22 version supported by the eUICC is not listed in section 7.1, the test SHALL fail.  SW = 0x9000 | RQ43\_003 |

##### 4.2.13.2.10 TC\_eUICC\_ES10b.GetEUICCInfo\_additionalEuiccProfilePackageVersions

The purpose of this test is to verify the profileVersion values returned by the eUICC.

Test Sequence #01 Nominal – additionalEuiccProfilePackageVersions field

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO2) | #R\_EUICC\_INFO2  If #IUT\_EUICC\_ADD\_PP\_VERSIONS is empty:   * If the SGP.22 version supported by the eUICC is v2.3 or v2.4: verify that <ANY\_ADD\_PP\_VERSIONS> is absent, or is present with empty content * If the SGP.22 version supported by the eUICC is v2.5 or later: verify that <ANY\_ADD\_PP\_VERSIONS> is absent   If #IUT\_EUICC\_ADD\_PP\_VERSIONS is not empty:   * Verify that <ANY\_ADD\_PP\_VERSIONS> is equal to #IUT\_EUICC\_ADD\_PP\_VERSIONS * Verify that <ANY\_ADD\_PP\_VERSIONS> encodes exactly one of the values listed in section 7.1 in the “Allowed values for #IUT\_EUICC\_ADD\_PP\_VERSIONS” column for the SGP.22 version supported by the eUICC. * If the SGP.22 version supported by the eUICC is not listed in section 7.1, the test SHALL fail   SW = 0x9000 | RQ43\_002 |

### 4.2.14 ES10b (LPA -- eUICC): ListNotification

#### 4.2.14.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ25\_020

 RQ31\_172

 RQ35\_016

 RQ57\_068, RQ57\_068\_1, RQ57\_068\_2, RQ57\_068\_3, RQ57\_068\_4, RQ57\_069, RQ57\_070

#### 4.2.14.2 Test Cases

Throughout all the ListNotification test cases the maximum number of Notifications simultaneously tested has been set as to two as there is not minimum defined in SGP.21 [3] or SGP.22 [2] for the number of Notifications that can be stored by the eUICC.

##### 4.2.14.2.1 TC\_eUICC\_ES10b.ListNotification

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | No Operational Profile is installed on the eUICC. |
| eUICC | No Notifications are stored in the eUICC's Pending Notifications List. |

Test Sequence #01 Nominal: Install Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Do not remove both the Notifications. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 RQ25\_020 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |

Test Sequence #02 Nominal: Enable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the Notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |

Test Sequence #03 Nominal: Disable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the Notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |

Test Sequence #04 Nominal: Delete Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the Notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |

Test Sequence #05 Nominal: Two Install Notifications (PIR) with different Notification Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Install PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 RQ25\_020 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |

Test Sequence #06 Nominal: Install Notification (PIR) and Enable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_IN1\_PIR\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 RQ25\_020 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_IN1\_PIR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_IN1\_PIR\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_IN1\_PIR\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2RQ57\_069 RQ57\_070 RQ25\_020 |

Test Sequence #07 Nominal: Disable and Delete Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the notification | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the notification | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DI1\_DE1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_DI1\_DE1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_DI1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_DE1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_DI1\_DE1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_DI1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_DI1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |

Test Sequence #08 Nominal: Install (OtherSignedNotification) and Enable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove the PIR notification, but do not remove the OtherSignedNotification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_IN1\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_IN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_IN1\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_IN1\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |

Test Sequence #09 Nominal: Enable and Install (PIR) Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | Install PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1\_IN2\_PIR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 RQ31\_172 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_EN1\_IN2\_PIR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_3 RQ57\_069 RQ57\_070 RQ25\_020 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_IN2\_PIR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_EN1\_IN2\_PIR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_EN1  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_069 RQ57\_070 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_EN1\_IN2\_PIR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_2 RQ57\_069 RQ57\_070 RQ25\_020 |

Test Sequence #10 Nominal: No Notifications available

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_OMITTED) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_3 RQ57\_068\_4 RQ57\_069 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_NONE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000  OR  #R\_LIST\_NOTIF\_UNDEFINED\_ERROR  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ENABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_DELETE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_DISABLE\_ENABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 | RQ35\_016 RQ57\_068 RQ57\_068\_1 RQ57\_068\_4 RQ57\_069 |

### 4.2.15 ES10b (LPA -- eUICC): RetrieveNotificationsList

#### 4.2.15.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ25\_020, RQ25\_021

 RQ26\_034, RQ26\_035

 RQ31\_174

 RQ35\_001\_1, RQ35\_001\_2, RQ35\_003\_1

 RQ57\_071, RQ57\_071\_1, Q57\_071\_2, RQ57\_071\_3, RQ57\_071\_4

 RQ57\_072, RQ57\_072\_1, RQ57\_072\_2, RQ57\_073, RQ57\_074, RQ57\_075, RQ57\_076

#### 4.2.15.2 Test Cases

Throughout all the RetrieveNotificationsList test cases the maximum number of Notifications simultaneously tested has been set to two as there is no minimum defined in SGP.21 [3] or SGP.22 [2] for the number of Notifications that can be stored by the eUICC.

In some test sequences defined below, it is expected to retrieve especially either a ProfileInstallationResult or an OtherSignedNotification for installation. When both are present in the eUICC, the only way to distinguish these two notifications is to compare their sequence numbers in the ListNotificationResponse. The sequence number related to the ProfileInstallationResult SHALL be lower than the one linked to the OtherSignedNotification.

This assumption is based on the requirement defined in section 5.5.5 of SGP.22 [2] stating that the eUICC SHALL first generate the Profile Installation Result and then as many Notifications as configured in its metadata in the format of OtherSignedNotification.

##### 4.2.15.2.1 TC\_eUICC\_ES10b.RetrieveNotificationsList

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | No Operational Profile is installed on the eUICC. |
| eUICC | No Notifications are stored in the eUICC's Pending Notifications List. |

Test Sequence #01 Nominal: Retrieve by Sequence Number for Install Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Do not remove both the notifications. | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_IN1\_PIR  SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN1>)) | #R\_RETRIEVE\_NOTIF\_IN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 RQ35\_001\_2 RQ35\_003\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 RQ35\_001\_2 RQ35\_003\_1 |

Test Sequence #02 Nominal: Retrieve by Sequence Number for Enable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_EN1>)) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_2 RQ35\_003\_1 |

Test Sequence #03 Nominal: Retrieve by Sequence Number for Disable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_DI1>)) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_2 RQ35\_003\_1 |

Test Sequence #04 Nominal: Retrieve by Sequence Number for Delete Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_DE1>)) | #R\_RETRIEVE\_NOTIF\_DE1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_2 RQ35\_003\_1 |

Test Sequence #05 Nominal: Retrieve by Sequence Number for Two Install (PIR) Notifications with different Notification Addresses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Install PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN2\_PIR>)) | #R\_RETRIEVE\_NOTIF\_IN2\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |

Test Sequence #06 Nominal: Retrieve by Sequence Number for Install (PIR) and Enable Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR\_EN1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_EN1>)) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |

Test Sequence #07 Nominal: Retrieve by Sequence Number for Disable and Delete Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the notification | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the notification | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification | | | |
| IC7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DI1\_DE1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_DI1>)) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_DE1>)) | #R\_RETRIEVE\_NOTIF\_DE1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |

Test Sequence #08 Nominal: Retrieve by Sequence Number for Install (OtherSignedNotification) and Enable Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove the PIR notification, but do not remove the OtherSignedNotification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_EN1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN1>)) | #R\_RETRIEVE\_NOTIF\_IN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_EN1>)) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |

Test Sequence #09 Nominal: Retrieve by Sequence Number for Enable and Install (PIR) notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | Install PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1\_IN2\_PIR  SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN2\_PIR>)) | #R\_RETRIEVE\_NOTIF\_IN2\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_EN1>)) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |

Test Sequence #10 Nominal: Retrieve Sequence Numbers that are not present

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ26\_034 RQ26\_035 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM(<NOTIF\_SEQ\_NO\_IN1\_PIR> +1)) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |

Test Sequence #11 Nominal: Retrieve by Notification Type for Install Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Do not remove both the notifications. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_3 RQ57\_071\_4 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_IN1\_IN1\_PIR  SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL \_ENABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_071\_3 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |

Test Sequence #12 Nominal: Retrieve by Notification Type for Enable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_4 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |

Test Sequence #13 Nominal: Retrieve by Notification Type for Disable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_4 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |

Test Sequence #14 Nominal: Retrieve by Notification Type for Delete Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_DE1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_DE1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_4 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_DE1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_DE1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |

Test Sequence #15 Nominal: Retrieve by Notification Type for Two Install (PIR) Notifications with different Notification Addresses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Install PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000  • Verify both the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_3 RQ57\_073\_1 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000  • Verify both the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_3 RQ57\_072\_4 RQ57\_073\_1 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_IN2\_PIR  SW = 0x9000  • Verify both the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_3 RQ57\_073\_1 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE  SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000  • Verify both the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_3 RQ57\_073\_1 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000  • Verify both the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_3 RQ57\_073\_1 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |

Test Sequence #16 Nominal: Retrieve by Notification Type for Install (PIR) and Enable Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_3 RQ57\_073\_1 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1  SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_PIR\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3 RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |

Test Sequence #17 Nominal: Retrieve by Notification Type for Disable and Delete Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the notification | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the notification | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_DI1\_DE1 SW = 0x9000  • Verify both the euiccNotificationSignatures <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_DI1\_DE1 SW = 0x9000  • Verify both the euiccNotificationSignatures <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_4  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_074 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_DE1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_DI1\_DE1 SW = 0x9000  • Verify both the euiccNotificationSignatures <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_DI1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |

Test Sequence #18 Nominal: Retrieve by Notification Type for Install (OtherSignedNotification) and Enable Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove the PIR notification, but do not remove the OtherSignedNotification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_IN1\_EN1 SW = 0x9000  • Verify both the euiccNotificationSignatures <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_IN1\_EN1 SW = 0x9000  • Verify both the euiccNotificationSignatures <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_1 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_IN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_1 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_EN1 SW = 0x9000  • Verify both the euiccNotificationSignatures <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_1 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1  SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_IN1\_EN1 SW = 0x9000  • Verify both the euiccNotificationSignatures <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 RQ35\_001\_1 |

Test Sequence #19 Nominal: Retrieve by Notification Type for Enable and Install (PIR) notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | Install PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_EN1\_IN2\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_EN1\_IN2\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3 RQ57\_072\_4  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_IN2\_PIR SW = 0x9000  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_072\_3 RQ57\_073\_1 RQ57\_074 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1 SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1\_IN2\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1  RQ57\_073 RQ57\_074 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_EN1  SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ26\_034 RQ26\_035 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_EN1\_IN2\_PIR SW = 0x9000  • Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2  RQ57\_072\_3  RQ57\_073\_1 RQ57\_074 RQ57\_075 RQ57\_076 RQ25\_020 RQ25\_021 RQ26\_034 RQ26\_035 RQ31\_174 RQ35\_001\_1 |

Test Sequence #20 Nominal: Retrieve by Notification Type for No Notifications available

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ALL) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_OMITTED) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_072\_4 RQ57\_073 RQ57\_074 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_NONE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_DELETE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_DISABLE\_ENABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE) | #R\_RETRIEVE\_NOTIF\_NONE SW = 0x9000 | RQ57\_071 RQ57\_072 RQ57\_072\_1 RQ57\_073 RQ57\_074 |

### 4.2.16 ES10b (LPA -- eUICC): RemoveNotificationFromList

#### 4.2.16.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ25\_020

 RQ31\_182

 RQ35\_021

 RQ57\_077, RQ57\_078, RQ57\_079

#### 4.2.16.2 Test Cases

Throughout all the RemoveNotificationFromList test cases the maximum number of Notifications simultaneously tested has been set as to two as there is no minimum defined in SGP.21 [3] or SGP.22 [2] for the number of Notifications that can be stored by the eUICC.

The rule specified in section 4.2.15.2 explaining the way to distinguish a ProfileInstallationResult from an OtherSignedNotification for installation also applies for the test cases defined below.

##### 4.2.16.2.1 TC\_eUICC\_ES10b.RemoveNotificationFromList

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | No Operational Profile is installed on the eUICC. |
| eUICC | No Notifications are stored in the eUICC's Pending Notifications List. |

Test Sequence #01 Nominal: Install Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Do not remove both the notifications. | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_IN1\_PIR SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR SW = 0x9000 |  |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 RQ31\_182 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ25\_020 RQ31\_182 |

Test Sequence #02 Nominal: Enable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_EN1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

Test Sequence #03 Nominal: Disable Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DI1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_DI1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

Test Sequence #04 Nominal: Delete Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Remove the Notification. | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC7 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_DE1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

Test Sequence #05 Nominal: Two Install (PIR) Notifications with different Notification Addresses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Install PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 RQ31\_182 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN2\_PIR SW = 0x9000 | RQ25\_020 RQ31\_182 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN2\_PIR>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 RQ31\_182 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 | RQ25\_020 RQ31\_182 |

Test Sequence #06 Nominal: Install (PIR) and Enable Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR\_EN1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 RQ31\_182 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ25\_020 RQ31\_182 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_EN1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

Test Sequence #07 Nominal: Disable and Delete Notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both the Notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Remove the Notification | | | |
| IC5 | Disable PROFILE\_OPERATIONAL1. Do not remove the Notification | | | |
| IC6 | Delete PROFILE\_OPERATIONAL1. Do not remove the Notification | | | |
| IC7 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DI1\_DE1 SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_DI1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1  SW = 0x9000 |  |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_DE1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

Test Sequence #08 Nominal: Install (OtherSignedNotification) and Enable Notifications

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | | Direction | Sequence / Description | Expected result | REQ |
| IC1 | | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | | Install PROFILE\_OPERATIONAL1. Remove the PIR notification, but do not remove the OtherSignedNotification. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_EN1 SW = 0x9000 |  |
| 1 | | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 2 | | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 |  |
| 3 | | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_EN1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 4 | | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

Test Sequence #09 Nominal: Enable and Install (PIR) notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1. Remove both notifications. | | | |
| IC4 | Enable PROFILE\_OPERATIONAL1. Do not remove the Notification. | | | |
| IC5 | Install PROFILE\_OPERATIONAL2 with METADATA\_OP\_PROF2\_NO\_INSTALL.  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA  Do not remove the Notification. | | | |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1\_IN2\_PIR SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN2\_PIR>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 RQ31\_182 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1 SW = 0x9000 | RQ25\_020 RQ31\_182 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_EN1>)) | #R\_REMOVE\_NOTIF\_OK  SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

Test Sequence #10 Nominal: Removing Sequence Numbers that are not present

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_INSTALL. Do not remove the Notification. | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR> - 1)) | #R\_REMOVE\_NOTIF\_NOTHING\_TO\_DELETE SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR SW = 0x9000 |  |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR> + 1)) | #R\_REMOVE\_NOTIF\_NOTHING\_TO\_DELETE SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN1\_PIR SW = 0x9000 |  |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_REMOVE\_NOTIF\_OK SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE  SW = 0x9000 |  |
| 7 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR>)) | #R\_REMOVE\_NOTIF\_NOTHING\_TO\_DELETE SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_REMOVE\_NOTIF(  <NOTIF\_SEQ\_NO\_IN1\_PIR> + 1)) | #R\_REMOVE\_NOTIF\_NOTHING\_TO\_DELETE SW = 0x9000 | RQ35\_021 RQ57\_077 RQ57\_078 RQ57\_079 |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_NONE SW = 0x9000 |  |

### 4.2.17 ES10b (LPA -- eUICC): LoadCRL

This section is defined as FFS.

### 4.2.18 ES10b (LPA -- eUICC): AuthenticateServer

#### 4.2.18.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_008

 RQ26\_005, RQ26\_006, RQ26\_007, RQ26\_008, RQ26\_010, RQ26\_012, RQ26\_013, RQ26\_029, RQ26\_033, RQ26\_034, RQ26\_035

 RQ31\_025, RQ31\_046, RQ31\_047, RQ31\_048, RQ31\_049, RQ31\_050, RQ31\_051, RQ31\_052, RQ31\_053, RQ31\_054, RQ31\_055, RQ31\_076, RQ31\_077, RQ31\_078, RQ31\_079

 RQ36\_017

 RQ42\_001

 RQ43\_001, RQ43\_002

 RQ45\_002, RQ45\_006, RQ45\_026, RQ45\_026\_1, RQ45\_028, RQ45\_030, RQ45\_032

 RQ55\_004, RQ55\_005

 RQ57\_093, RQ57\_094, RQ57\_095, RQ57\_096, RQ57\_097, RQ57\_098, RQ57\_099, RQ57\_100, RQ57\_101, RQ57\_102, RQ57\_103, RQ57\_104, RQ57\_105, RQ57\_106, RQ57\_107, RQ57\_108

#### 4.2.18.2 Test Cases

##### 4.2.18.2.1 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_NIST

Test Sequence #01 Nominal: Without MatchingID in CtxParams1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on NIST P-256 curve |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on NIST P-256 curve   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTHENTICATE\_SMDP) | #R\_AUTHENTICATE\_SMDP  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTHENTICATE\_SMDP. • Verify that the <S\_SMDP\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTHENTICATE\_SMDP | RQ26\_029 RQ26\_005 RQ26\_006 RQ26\_007 RQ26\_008 RQ26\_034 RQ26\_035 RQ31\_025 RQ31\_046 RQ31\_047 RQ31\_048 RQ31\_049 RQ31\_050 RQ31\_051 RQ31\_053 RQ31\_054 RQ31\_055 RQ31\_076 RQ31\_079 RQ42\_001 RQ43\_001 RQ43\_002 RQ45\_002 RQ55\_004 RQ55\_005 RQ57\_094 RQ57\_095 RQ57\_096 RQ57\_097 RQ57\_098 RQ57\_099 RQ57\_101 RQ57\_102 RQ57\_103 RQ57\_104 RQ57\_105 RQ57\_106 RQ57\_107 RQ57\_108 |

Test Sequence #02 Nominal: With MatchingID in CtxParams1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on NIST P-256 curve |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on NIST P-256 curve   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTH\_SMDP\_MATCH\_ID) | #R\_AUTH\_SMDP\_MATCH\_ID  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_MATCH\_ID • Verify that the <S\_SMDP\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_MATCH\_ID | RQ26\_029 RQ26\_005 RQ26\_006 RQ26\_007 RQ26\_008 RQ26\_034 RQ26\_035 RQ31\_025 RQ31\_046 RQ31\_047 RQ31\_048 RQ31\_049 RQ31\_050 RQ31\_051 RQ31\_053 RQ31\_054 RQ31\_055 RQ31\_076 RQ31\_077 RQ42\_001 RQ43\_001 RQ43\_002 RQ45\_002 RQ55\_004 RQ55\_005 RQ57\_094 RQ57\_095 RQ57\_096 RQ57\_097 RQ57\_098 RQ57\_099 RQ57\_101 RQ57\_102 RQ57\_103 RQ57\_104 RQ57\_105 RQ57\_106 RQ57\_107 RQ57\_108 |

Test Sequence #03 Nominal: With IMEI in Device Capabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on NIST P-256 curve |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on NIST P-256 curve   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTH\_SMDP\_IMEI) | #R\_AUTH\_SMDP\_IMEI  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_IMEI • Verify that the <S\_SMDP\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_IMEI | RQ26\_029 RQ26\_005 RQ26\_006 RQ26\_007 RQ26\_008 RQ26\_034 RQ26\_035 RQ31\_025 RQ31\_046 RQ31\_047 RQ31\_048 RQ31\_049 RQ31\_050 RQ31\_051 RQ31\_053 RQ31\_054 RQ31\_055 RQ31\_076 RQ42\_001 RQ43\_001 RQ43\_002 RQ45\_002 RQ55\_004 RQ55\_005 RQ57\_094 RQ57\_095 RQ57\_096 RQ57\_097 RQ57\_098 RQ57\_099 RQ57\_101 RQ57\_102 RQ57\_103 RQ57\_104 RQ57\_105 RQ57\_106 RQ57\_107 RQ57\_108 |

##### 4.2.18.2.2 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_BRP

Test Sequence #01 Nominal: Without MatchingID in CtxParams1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on BrainpoolP256r1 curve |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on BrainpoolP256r1 curve   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTHENTICATE\_SMDP) | #R\_AUTHENTICATE\_SMDP  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in **#**AUTHENTICATE\_SMDP. • Verify that the <S\_SMDP\_CHALLENGE> present in the euiccSigned1 is the same as in **#**AUTHENTICATE\_SMDP | RQ26\_029 RQ26\_005 RQ26\_006 RQ26\_007 RQ26\_008 RQ26\_034 RQ26\_035 RQ31\_025 RQ31\_046 RQ31\_047 RQ31\_048 RQ31\_049 RQ31\_050 RQ31\_051 RQ31\_053 RQ31\_054 RQ31\_055 RQ31\_076 RQ31\_079 RQ42\_001 RQ43\_001 RQ43\_002 RQ45\_002 RQ55\_004 RQ55\_005 RQ57\_094 RQ57\_095 RQ57\_096 RQ57\_097 RQ57\_098 RQ57\_099 RQ57\_101 RQ57\_102 RQ57\_103 RQ57\_104 RQ57\_105 RQ57\_106 RQ57\_107 RQ57\_108 |

Test Sequence #02 Nominal: With MatchingID in CtxParams1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on BrainpoolP256r1 curve |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on BrainpoolP256r1 curve   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTH\_SMDP\_MATCH\_ID) | #R\_AUTH\_SMDP\_MATCH\_ID  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_MATCH\_ID • Verify that the <S\_SMDP\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_MATCH\_ID | RQ26\_029 RQ26\_005 RQ26\_006 RQ26\_007 RQ26\_008 RQ26\_034 RQ26\_035 RQ31\_025 RQ31\_046 RQ31\_047 RQ31\_048 RQ31\_049 RQ31\_050 RQ31\_051 RQ31\_053 RQ31\_054 RQ31\_055 RQ31\_076 RQ31\_079 RQ42\_001 RQ43\_001 RQ43\_002 RQ45\_002 RQ55\_004 RQ55\_005 RQ57\_094 RQ57\_095 RQ57\_096 RQ57\_097 RQ57\_098 RQ57\_099 RQ57\_101 RQ57\_102 RQ57\_103 RQ57\_104 RQ57\_105 RQ57\_106 RQ57\_107 RQ57\_108 |

Test Sequence #03 Nominal: With IMEI in Device Capabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on BrainpoolP256r1 curve |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   * <S\_TRANSACTION\_ID> * <EUICC\_CHALLENGE> * <S\_SMDP\_CHALLENGE> * <S\_SMDP\_SIGNATURE1> * Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on BrainpoolP256r1 curve * Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTH\_SMDP\_IMEI) | #R\_AUTH\_SMDP\_IMEI  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_IMEI • Verify that the <S\_SMDP\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTH\_SMDP\_IMEI | RQ26\_029 RQ26\_005 RQ26\_006 RQ26\_007 RQ26\_008 RQ26\_034 RQ26\_035 RQ31\_025 RQ31\_046 RQ31\_047 RQ31\_048 RQ31\_049 RQ31\_050 RQ31\_051 RQ31\_053 RQ31\_054 RQ31\_055 RQ31\_076 RQ31\_079 RQ42\_001 RQ43\_001 RQ43\_002 RQ45\_002 RQ55\_004 RQ55\_005 RQ57\_094 RQ57\_095 RQ57\_096 RQ57\_097 RQ57\_098 RQ57\_099 RQ57\_101 RQ57\_102 RQ57\_103 RQ57\_104 RQ57\_105 RQ57\_106 RQ57\_107 RQ57\_108 |

##### 4.2.18.2.3 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_FRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.2.18.2.4 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DP+\_ErrorCases

Test Sequence #01 Error: With Incorrect SM-DPauth certificate (i.e. invalid signature)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DPauth\_INV\_SIGN leading to the same Root CI certificate apart from the signature | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTH\_SMDP\_INV\_CERT) | #R\_AUTH\_SERVER\_INV\_CERT  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTH\_SMDP\_INV\_CERT. | RQ26\_005 RQ26\_006 RQ31\_052 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 RQ55\_005 RQ57\_100 RQ57\_095 RQ57\_100 RQ57\_105 RQ57\_107  RQ26\_010 |

Test Sequence #02 Error: With Invalid SM-DP+ Signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1> NOT computed with the #SK\_S\_SM\_DPauth\_ECDSA but SHALL have the same length as for a valid signature   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(#AUTHENTICATE\_SMDP) | #R\_AUTH\_SERVER\_INV\_SIGN  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTHENTICATE\_SMDP | RQ31\_052 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 RQ55\_005 RQ57\_100 RQ57\_097 RQ57\_100 RQ57\_105 RQ57\_107 RQ26\_010 |

Test Sequence #03 Error: Unsupported Curve

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  | |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  | |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <RANDOM\_SM\_DP+\_SIGN>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   #CERT\_S\_SM\_DPauth\_INV\_CURVE | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTH\_SMDP\_INV\_CURV) | #R\_AUTH\_SERVER\_INV\_CURV  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTH\_SMDP\_INV\_CURV. | RQ26\_005 RQ26\_006 RQ31\_049 RQ31\_052 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 RQ55\_005 RQ57\_097 RQ57\_100 RQ57\_105 RQ57\_107  RQ26\_010 | |

Test Sequence #04 Error: eUICC Challenge Mismatch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000 |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   #S\_EUICC\_CHALLENGE considered as different from <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTH\_SMDP\_INV\_CHALLENGE) | #R\_AUTH\_SERVER\_INV\_CHALLENGE  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTH\_SMDP\_INV\_CHALLENGE. | RQ26\_005 RQ26\_006 RQ31\_050 RQ31\_052 RQ57\_098 RQ57\_100 RQ57\_105 RQ57\_107 RQ26\_010 |

Test Sequence #05 Error: Unknown CI PK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to a CI Key ID not present in the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> (a random SubjectKeyIdentifier can be used)   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTHENTICATE\_SMDP) | #R\_AUTH\_SERVER\_INV\_CI  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTHENTICATE\_SMDP. | RQ26\_005 RQ26\_006 RQ26\_033 RQ31\_048 RQ31\_051 RQ31\_052 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 RQ57\_099 RQ57\_100 RQ57\_105 RQ57\_107 RQ26\_010 |

Test Sequence #06 Error: Invalid Certificate Role OID

The purpose of this sequence is to make sure that the eUICC refuses any SM-DP+ Certificate for authentication that does not indicate “id-rspRole-dp-auth” in its extension for Certificate Policies.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DPpb\_ECDSA (instead of #CERT\_S\_SM\_DPauth\_ECDSA) leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTH\_SMDP\_INV\_OID) | #R\_AUTH\_SERVER\_INV\_OID  SW = 0x9000  OR  #R\_AUTH\_SERVER\_INV\_CERT  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTH\_SMDP\_INV\_OID. | RQ26\_005 RQ26\_006 RQ31\_052 RQ45\_006 RQ45\_026\_1 RQ45\_026 RQ45\_028 RQ45\_030 RQ57\_096 RQ57\_100 RQ57\_105 RQ57\_107 RQ26\_010 |

Test Sequence #07 Error: No RSP session on-going

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial state |
| eUICC | No RSP session is on-going (i.e. no ES10b.getEUICCChallenge has been sent to the eUICC). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | The following inputs are required for Step 3 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   #S\_EUICC\_CHALLENGE   <S\_SMDP\_CHALLENGE>   <S\_SMDP\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTH\_SMDP\_INV\_CHALLENGE) | #R\_AUTH\_SERVER\_NO\_SESSION  SW = 0x9000  The transactionId returned in the response SHALL not be checked (any value SHALL be accepted) | RQ26\_005 RQ26\_006RQ31\_052 RQ57\_094 RQ57\_100 RQ57\_105 RQ57\_107 |

##### 4.2.18.2.5 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_BRP

Test Sequence #01 Nominal: With EventID in CtxParams1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on BrainpoolP256r1 curve | |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> | |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on BrainpoolP256r1 curve   Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTHENTICATE\_SMDS) | #R\_AUTHENTICATE\_SMDS  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTHENTICATE\_SMDS. • Verify that the <S\_SMDS\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTHENTICATE\_SMDS | RQ24\_008RQ26\_005 RQ26\_006 RQ26\_008 RQ26\_012 RQ26\_013 RQ26\_029 RQ26\_034 RQ31\_025 RQ31\_078 RQ43\_002 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ57\_094RQ57\_095RQ57\_096RQ57\_097RQ57\_098RQ57\_099 RQ57\_101RQ57\_102RQ57\_103RQ57\_104RQ57\_105RQ57\_106RQ57\_107RQ57\_108 RQ31\_046RQ31\_047RQ31\_048RQ31\_049RQ31\_050RQ31\_051RQ31\_053RQ31\_054RQ31\_055RQ26\_029 | |

Test Sequence #02 Nominal: With IMEI and EventID in Device Capabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on BrainpoolP256r1 curve |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on BrainpoolP256r1 curve   Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTH\_SMDS\_IMEI) | #R\_AUTH\_SMDS\_IMEI  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTH\_SMDS\_IMEI • Verify that the <S\_SMDS\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTH\_SMDS\_IMEI | RQ24\_008 RQ26\_005 RQ26\_006 RQ26\_008 RQ26\_012 RQ26\_013 RQ26\_029 RQ26\_034 RQ31\_025 RQ31\_078 RQ43\_002 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ57\_094 RQ57\_095 RQ57\_096 RQ57\_097 RQ57\_098 RQ57\_099 RQ57\_101 RQ57\_102 RQ57\_103 RQ57\_104 RQ57\_105 RQ57\_106 RQ57\_107 RQ57\_108 RQ31\_046 RQ31\_047 RQ31\_048 RQ31\_049 RQ31\_050 RQ31\_051 RQ31\_053 RQ31\_054 RQ31\_055 RQ26\_029 |

##### 4.2.18.2.6 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_NIST

Test Sequence #01 Nominal: With EventID in CtxParams1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on NIST P-256 curve | | RQ36\_017 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> | | RQ36\_017 |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on NIST P-256 curve   Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTHENTICATE\_SMDS) | #R\_AUTHENTICATE\_SMDS  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTHENTICATE\_SMDS. • Verify that the <S\_SMDS\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTHENTICATE\_SMDS | RQ24\_008RQ26\_005 RQ26\_006 RQ26\_008 RQ26\_012 RQ26\_013 RQ26\_029 RQ26\_034 RQ31\_025 RQ31\_078 RQ43\_002 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ57\_094RQ57\_095RQ57\_096RQ57\_097RQ57\_098RQ57\_099 RQ57\_101RQ57\_102RQ57\_103RQ57\_104RQ57\_105RQ57\_106RQ57\_107RQ57\_108 RQ31\_046RQ31\_047RQ31\_048RQ31\_049RQ31\_050RQ31\_051RQ31\_053RQ31\_054RQ31\_055RQ26\_029 RQ36\_017 | |

Test Sequence #02 Nominal: With IMEI and EventID in Device Capabilities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID based on NIST P-256 curve | RQ36\_017 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> | RQ36\_017 |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID based on NIST P-256 curve   Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTH\_SMDS\_IMEI) | #R\_AUTH\_SMDS\_IMEI  SW = 0x9000  • Verify the <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned1 is the same as in #AUTH\_SMDS\_IMEI • Verify that the <S\_SMDS\_CHALLENGE> present in the euiccSigned1 is the same as in #AUTH\_SMDS\_IMEI | RQ24\_008RQ26\_005 RQ26\_006 RQ26\_008 RQ26\_012 RQ26\_013 RQ26\_029 RQ26\_034 RQ31\_025 RQ31\_078 RQ43\_002 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ57\_094RQ57\_095RQ57\_096RQ57\_097RQ57\_098RQ57\_099 RQ57\_101RQ57\_102RQ57\_103RQ57\_104RQ57\_105RQ57\_106RQ57\_107RQ57\_108 RQ31\_046RQ31\_047RQ31\_048RQ31\_049RQ31\_050RQ31\_051RQ31\_053RQ31\_054RQ31\_055RQ26\_029 RQ36\_017 |

##### 4.2.18.2.7 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_FRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.2.18.2.8 TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_ErrorCases

Test Sequence #01 Error: With Incorrect SM-DSauth certificate (i.e. invalid signature)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 | |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> | |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DSauth\_INV\_SIGN leading to the same Root CI certificate | | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTH\_SMDS\_INV\_CERT) | | #R\_AUTH\_SERVER\_INV\_CERT  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTH\_SMDS\_INV\_CERT. | RQ45\_028 RQ57\_100 RQ31\_052 RQ57\_095 RQ57\_105 RQ57\_107 RQ26\_010 |

Test Sequence #02 Error: With Invalid SM-DS Signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1> NOT computed with the #SK\_S\_SM\_DSauth\_ECDSA but SHALL have the same length as for a valid signature   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTHENTICATE\_SMDS) | #R\_AUTH\_SERVER\_INV\_SIGN  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTHENTICATE\_SMDS | RQ57\_100 RQ31\_052 RQ57\_097 RQ57\_105 RQ57\_107 RQ31\_049 RQ26\_010 |

Test Sequence #03 Error: Unsupported Curve

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <RANDOM\_SM\_DS\_SIGN>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   #CERT\_S\_SM\_DSauth\_INV\_CURVE | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTH\_SMDS\_INV\_CURV) | #R\_AUTH\_SERVER\_INV\_CURV  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTH\_SMDS\_INV\_CURV. | RQ57\_100 RQ31\_052 RQ57\_105 RQ57\_107 RQ26\_010 |

Test Sequence #04 Error: eUICC Challenge Mismatch

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000 |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   * <S\_TRANSACTION\_ID> * #S\_EUICC\_CHALLENGE considered as different from <EUICC\_CHALLENGE> * <S\_SMDS\_CHALLENGE> * <S\_SMDS\_SIGNATURE1> * Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> * Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (  #AUTH\_SMDS\_INV\_CHALLENGE) | #R\_AUTH\_SERVER\_INV\_CHALLENGE  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTH\_SMDS\_INV\_CHALLENGE. | RQ57\_100 RQ31\_052 RQ57\_098 RQ57\_105 RQ57\_107 RQ31\_050 RQ26\_010 |

Test Sequence #05 Error: Unknown CI PK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| 3 | The following inputs are required for Step 4 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   <EUICC\_CHALLENGE>   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to a CI Key ID not present in the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> (a random SubjectKeyIdentifier can be used)   Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT (#AUTHENTICATE\_SMDS) | #R\_AUTH\_SERVER\_INV\_CI  SW = 0x9000  • Verify that the <S\_TRANSACTION\_ID> present in the AuthenticateResponseError is the same as in #AUTHENTICATE\_SMDS. | RQ26\_029 RQ45\_028 RQ57\_100 RQ31\_052 RQ57\_099 RQ57\_105 RQ57\_107 RQ31\_051 RQ31\_048 RQ26\_010 |

Test Sequence #06 Error: No RSP session on-going

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial state |
| eUICC | No RSP session is on-going (i.e. no ES10b.getEUICCChallenge has been sent to the eUICC). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000 |  |
| 2 | The following inputs are required for Step 3 as described in the InitiateAuthentication function:   <S\_TRANSACTION\_ID>   #S\_EUICC\_CHALLENGE   <S\_SMDS\_CHALLENGE>   <S\_SMDS\_SIGNATURE1>   Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>   Choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | | | |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTH\_SMDS\_INV\_CHALLENGE) | #R\_AUTH\_SERVER\_NO\_SESSION  SW = 0x9000  The transactionId returned in the response SHALL not be checked (any value SHALL be accepted) | RQ57\_100 RQ31\_052 RQ57\_094RQ57\_105 RQ57\_107 |

### 4.2.19 ES10b (LPA -- eUICC): CancelSession

#### 4.2.19.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_034, RQ26\_035

 RQ31\_099, RQ31\_101, RQ31\_114, RQ31\_115, RQ31\_116, RQ31\_117, RQ31\_160, RQ31\_162\_1, RQ31\_188\_1

 RQ57\_041\_1, RQ57\_109, RQ57\_110, RQ57\_111, RQ57\_113, RQ57\_114, RQ57\_115, RQ57\_116

#### 4.2.19.2 Test Cases

##### 4.2.19.2.1 TC\_eUICC\_ES10b.CancelSessionNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+ (i.e. the response has been sent by the eUICC for ES10b.AuthenticateServer)   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI based on NIST P-256 curve has been chosen for signing and for verification |

Test Sequence #01 Nominal: End User Rejection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_REJECT) | #R\_CANCEL\_SESSION\_REJ  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_REJECT | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ26\_034 RQ26\_035 RQ31\_160 |
| 2 | | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

Test Sequence #02 Nominal: End User Postponed

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #CANCEL\_SESSION\_POSTPONED) | #R\_CANCEL\_SESSION\_POSTPONED  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_POSTPONED | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ26\_034 RQ26\_035 RQ31\_160 |
| 2 | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

Test Sequence #03 Nominal: Timeout

The RSP session is delayed because the End User does not confirm the download of the Profile within the timeout interval defined by the LPAd.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_TIMEOUT) | #R\_CANCEL\_SESSION\_TIMEOUT  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_TIMEOUT | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ26\_034 RQ26\_035 |
| 2 | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

Test Sequence #04 Nominal: PPR not allowed

The RSP session is terminated because the LPAd detected that PPR(s) set in the Profile Metadata is/are not allowed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_PPR) | #R\_CANCEL\_SESSION\_PPR  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_PPR | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ26\_034 RQ26\_035 RQ31\_099 RQ31\_101 |
| 2 | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

Test Sequence #05 Nominal: Metadata Mismatch

The RSP session is terminated because the LPAd detected that the Profile Metadata in the response to "ES9+.AuthenticateClient" does not match the Profile Metadata in the Bound Profile Package.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+ (i.e. the response has been sent by the eUICC for ES10b.PrepareDownload)   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_METADATA) | #R\_CANCEL\_SESSION\_METADATA  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_METADATA | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ26\_034 RQ26\_035 |
| 2 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| 3 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| 4 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x6985 or 0x6A88 | RQ57\_113 RQ57\_041\_1 |
| 6 | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

Test Sequence #06 Nominal: BPP Parsing Error

The RSP session is terminated because the LPAd has encountered an error while parsing the Bound Profile Package received from the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+ (i.e. the response has been sent by the eUICC for ES10b.PrepareDownload)   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_LOAD\_BPP) | #R\_CANCEL\_SESSION\_LOAD\_BPP  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_LOAD\_BPP | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ26\_034 RQ26\_035 RQ31\_162\_1 |
| 2 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| 3 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| 4 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x6985 or 0x6A88 | RQ57\_113 |
| 6 | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

Test Sequence #07 Nominal: Load BPP Execution Error

The RSP session is terminated because the LPAd has encountered an error while installing the Bound Profile Package received from the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+ (i.e. the response has been sent by the eUICC for ES10b.PrepareDownload)   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| IC2 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_PROF1,  #METADATA\_OP\_PROF1,  NO\_PARAM,  #UPP\_OP\_PROF1) | | | |
| IC3 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_LOAD\_BPP) | #R\_CANCEL\_SESSION\_LOAD\_BPP  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_LOAD\_BPP | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ31\_188\_1 RQ26\_034 RQ26\_035 RQ31\_162\_1 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0) | SW=0x6985 or 0x6A88 | RQ57\_113 |
| 3 | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

Test Sequence #08 Nominal: Undefined Reason

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_UNDEF) | #R\_CANCEL\_SESSION\_UNDEF  SW = 0x9000  The <EUICC\_CS\_SIGNATURE> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the response is the same as in #CANCEL\_SESSION\_UNDEF | RQ31\_114 RQ31\_115 RQ31\_116 RQ31\_117 RQ57\_110 RQ57\_111 RQ57\_114 RQ57\_115 RQ57\_116 RQ26\_034 RQ26\_035 |
| 2 | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | | RQ57\_113 |

##### 4.2.19.2.2 TC\_eUICC\_ES10b.CancelSessionBRP

In these test sequences, once the RSP session has been cancelled, verifications are performed in order to check that it is neither possible to execute the Download Confirmation procedure nor to execute the Common Mutual Authentication procedure by referring to the cancelled TransactionID.

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI based on BrainpoolP256r1 curve has been chosen for signing and for verification |

Test Sequence #01 Nominal: End User Rejection

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #02 Nominal: End User Postponed

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #03 Nominal: Timeout

This test sequence SHALL be the same as the Test Sequence #03 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #04 Nominal: PPR not allowed

This test sequence SHALL be the same as the Test Sequence #04 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #05 Nominal: Metadata Mismatch

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial state** |
| eUICC | Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

This test sequence SHALL be the same as the Test Sequence #05 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #06 Nominal: BPP Parsing Error

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial state** |
| eUICC | Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

This test sequence SHALL be the same as the Test Sequence #06 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #07 Nominal: Load BPP Execution Error

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial state** |
| eUICC | Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S\_SM-DP+   * #PREP\_DOWNLOAD\_NO\_CC has been sent to the eUICC |

This test sequence SHALL be the same as the Test Sequence #07 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #08 Nominal: Undefined Reason

This test sequence SHALL be the same as the Test Sequence #08 defined in section 4.2.19.2.1 – TC\_eUICC\_ES10b.CancelSessionNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

##### 4.2.19.2.3 TC\_eUICC\_ES10b.CancelSessionFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.2.19.2.4 TC\_eUICC\_ES10b.CancelSession\_ErrorCase

Test Sequence #01 Error: No on-going RSP session

On receiving a CancelSession request whereas there is no on-going RSP session, the eUICC SHALL return an error code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No RSP session is on-going (i.e. no Common Mutual Authentication procedure has been executed). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_INV\_TRANS\_ID) | #R\_CANCEL\_SESSION\_INV\_TRANS\_ID  SW = 0x9000 | RQ57\_109 RQ57\_114 RQ57\_115 |

Test Sequence #02 Error: Invalid Transaction ID

On receiving a CancelSession request with a TransactionID different from the on-going one, the eUICC SHALL not discard the current RSP session and return an error code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_INV\_TRANS\_ID) | #R\_CANCEL\_SESSION\_INV\_TRANS\_ID  SW = 0x9000 | RQ57\_109  RQ57\_114  RQ57\_115 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_NO\_CC) | #R\_PREP\_DOWNLOAD\_NO\_CC  SW=0x9000 |  |

### 4.2.20 ES10c (LPA -- eUICC): GetProfilesInfo

#### 4.2.20.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_029

 RQ32\_057

 RQ31\_183

 RQ57\_117, RQ57\_118, RQ57\_119, RQ57\_120, RQ57\_121, RQ57\_122, RQ57\_123, RQ57\_124, RQ57\_125, RQ57\_126

#### 4.2.20.2 Test Cases

##### 4.2.20.2.1 TC\_eUICC\_ES10c.GetProfilesInfo

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The Nickname of PROFILE\_OPERATIONAL1 and PROFILE\_OPERATIONAL2 is empty. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL3 is equal to #NICKNAME3. |

Test Sequence #01 Nominal: Get All Profiles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1,  #PROFILE\_INFO2,  #PROFILE\_INFO3  }  SW = 0x9000 | RQ32\_057 RQ57\_117 RQ57\_118 RQ57\_119 RQ57\_123 RQ24\_029 RQ31\_183 |

Test Sequence #02 Nominal: Get Profile by ICCID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW = 0x9000 | RQ57\_117 RQ57\_119 RQ57\_121 RQ57\_123 |

Test Sequence #03 Nominal: Get Profile by AID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW = 0x9000 | | RQ57\_117 RQ57\_119 RQ57\_121 RQ57\_123 |

Test Sequence #04 Nominal: Get All Operational Profiles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_PROFCLASS) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1,  #PROFILE\_INFO2,  #PROFILE\_INFO3  }  SW = 0x9000 | RQ57\_119 RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #05 Nominal: Get Profile ICCID list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_ICCID) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST\_ICCID  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #06 Nominal: Get Profile AID list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_ISDPAID) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST\_ISDPAID  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #07 Nominal: Get Profile Nickname list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_PROFILE\_NICKNAME) | response ProfileInfoListResponse::= profileInfoListOk : {  …  #PROFILES\_INFO\_TAGLIST\_PROFILE\_NICKNAME  ...  }  SW = 0x9000 | RQ57\_119 RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #08 Nominal: Get Profile SP Name list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_SP\_NAME) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST\_SP\_NAME  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #09 Nominal: Get Profile Name list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_PROFILE\_NAME) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST\_PROFILE\_NAME  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #10 Nominal: Get Profile Icon list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_ICON) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST\_ICON  }  SW = 0x9000 | RQ57\_120, RQ57\_122, RQ57\_123 |

Test Sequence #11 Nominal: Get Profile Owner list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_PROFILE\_OWNER) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST\_PROFILE\_OWNER  }  SW = 0x9000 | RQ57\_120, RQ57\_122, RQ57\_123, RQ57\_125 |

Test Sequence #12 Nominal: Get Profile SM-DP+ proprietary data list

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC with dpProprietaryData #SMDP\_PROP\_DATA1 (i.e. #CONF\_ISDP\_PROF1 is used during the Profile downloading). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST\_SMDP\_PROP\_DATA) | response ProfileInfoListResponse::=  profileInfoListOk : {  …  #PROFILES\_INFO\_TAGLIST\_SMDP\_PROP\_DATA  …  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #13 Nominal: Get Profile ICCID and State list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST1) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST1  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #14 Nominal: Get Profile Nickname and State list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST2) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST2  }  SW = 0x9000 | RQ57\_119 RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #15 Nominal: Get Profile Icon and Icon Type list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST3) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST3  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #16 Nominal: Get Profile Icon and State list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_TAGLIST4) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST4  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #17 Nominal: Get Operational Profile ICCID and State list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_OPTAGLIST1) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST1  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #18 Nominal: Get Operational Profile Nickname and State list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_OPTAGLIST2) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST2  }  SW = 0x9000 | RQ57\_119 RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #19 Nominal: Get Operational Profile Icon and Icon type list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_OPTAGLIST3) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST3  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #20 Nominal: Get Operational Profile Icon and State list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_OPTAGLIST4) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_TAGLIST4  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #21 Nominal: Get Profile State of the defined Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_ICCID\_TAGLIST1) | response ProfileInfoListResponse::=  profileInfoListOk : { #PROFILES\_INFO\_ICCID\_TAGLIST1  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #22 Nominal: Get Profile Icon Type of the defined Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_ICCID\_TAGLIST2) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_ICCID\_TAGLIST2  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #23 Nominal: Get Profile Class of the defined Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_ICCID\_TAGLIST3) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_ICCID\_TAGLIST3  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #24 Nominal: Get Notification Configuration of the defined Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_ICCID\_TAGLIST4) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_ICCID\_TAGLIST4  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 |

Test Sequence #25 Nominal: Get Profile Policy Rules of the defined Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_ICCID\_TAGLIST5) | response ProfileInfoListResponse::=  profileInfoListOk : {  #PROFILES\_INFO\_ICCID\_TAGLIST5  }  SW = 0x9000 | RQ57\_120 RQ57\_122 RQ57\_123 RQ57\_126 |

Test Sequence #26 Nominal: Get empty Profile list. No Profile installed

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Profile is loaded on the eUICC (this condition overrides the general initial condition defined in this test case). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (#GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW = 0x9000 | RQ57\_124 |

### 4.2.21 ES10c (LPA -- eUICC): EnableProfile

#### 4.2.21.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_010

 RQ29\_002, RQ29\_022

 RQ32\_011, RQ32\_012, RQ32\_016\_1, RQ32\_016\_2, RQ32\_016\_3, RQ32\_017, RQ32\_017\_1, RQ32\_017\_2, RQ32\_018\_1

 RQ34\_015

 RQ57\_127, RQ57\_127\_1, RQ57\_127\_2, RQ57\_128, RQ57\_129, RQ57\_130, RQ57\_132, RQ57\_132\_1, RQ57\_133\_1, RQ57\_133\_3, RQ57\_134, RQ57\_135\_1, RQ57\_135\_2,RQ57\_135\_4, RQ57\_136, RQ57\_137, RQ57\_138, RQ57\_139, RQ57\_140, RQ57\_140\_1

#### 4.2.21.2 Test Cases

##### 4.2.21.2.1 TC\_eUICC\_ES10c.EnableProfile\_Case3

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Nominal: Enable Profile by ISD-P AID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | No response data is returned  SW=0x91XX | RQ24\_010 RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_133\_3 RQ57\_138 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 RQ57\_135\_1 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ24\_010 RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_138 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #02 Nominal: Enable Profile by ICCID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | No response data is returned  SW=0x91XX | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_138 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 RQ57\_135\_1 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_138 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #03 Nominal: Enable Profile by ISD-P AID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | No response data is returned  SW=0x91XX | RQ24\_010 RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_138 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State change”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 RQ57\_135\_1 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ24\_010 RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_138 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 8 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #04 Nominal: Enable Profile by ICCID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | No response data is returned  SW=0x91XX | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_138 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State change”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 RQ57\_135\_1 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_138 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 8 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #05 Nominal: Enable Profile by ISD-P AID and “refreshFlag” not set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | | Direction | Sequence / Description | | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | | | |
| 1 | S\_LPAd → eUICC | | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | No response data is returned  SW=0x9000 | | RQ24\_010 RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_132\_1 RQ57\_138 RQ57\_132\_1 | |
| 2 | S\_Device 🡪 eUICC | | [TERMINAL\_PROFILE] | Toolkit initialization THEN SW=0x9000 | | RQ32\_018\_1 RQ57\_135\_4 | |
| 3 | S\_LPAd 🡪eUICC | | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | | RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_132\_1 RQ57\_138 RQ57\_132\_1 RQ24\_010 | |
| 4 | S\_Device 🡪 eUICC | | [SELECT\_ICCID] | SW=0x9000 | |  | |
| 5 | S\_Device 🡪 eUICC | | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | | RQ34\_015 | |

Test Sequence #06 Nominal: Enable Profile by ICCID and “refreshFlag” not set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  FALSE)) | No response data is returned  SW=0x9000 | RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_132\_1 RQ57\_138 RQ57\_132\_1 |
| 2 | S\_Device 🡪 eUICC | [TERMINAL\_PROFILE] | Toolkit initialization THEN SW=0x9000 | RQ32\_018\_1 RQ57\_135\_4 |
| 3 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_132\_1 RQ57\_138 RQ57\_132\_1 |
| 4 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 5 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #07 Nominal: Enable Profile by ICCID with refreshFLag set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM,  TRUE)) | No response data is returned  SW=0x91YY | RQ32\_011 |
| 2 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 |

Test Sequence #08 Nominal: Enable Profile by ICCID with refreshFLag not set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM,  FALSE)) | No response data is returned  SW=0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 4 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 |

Test Sequence #09 Nominal: Enable Profile by ICCID with refreshFLag set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| 1 | S\_Device → eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM,  TRUE)) | No response data returned SW=0x9000 | RQ32\_011 |
| 3 | S\_Device → eUICC | TERMINAL RESPONSE | SW=0x91YY |  |
| 4 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 5 | Repeat IC1 and IC2 | | | |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 8 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 |

Test Sequence #10 Nominal: Enable Profile by ICCID with refreshFLag not set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| 1 | S\_Device → eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM,  FALSE)) | No response data is returned  SW=0x9000 |  |
| 3 | S\_Device → eUICC | TERMINAL RESPONSE | SW= any value except 91XX |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 |

##### 4.2.21.2.2 TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case3

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Error: Enable Profile by an unknown ISD-P AID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The Operational Profile identified by the ISD-P AID <ISD\_P\_AIDX> is not loaded. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AIDX>,  TRUE)) | SW=0x6A82 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_139 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_139 |

Test Sequence #02 Error: Enable Profile by an unknown ICCID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The Operational Profile identified by the ICCID #ICCID\_OP\_PROFX is not loaded. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROFX,  NO\_PARAM,  TRUE)) | SW=0x6A82 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_139 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_139 |

Test Sequence #03 Error: Enable Profile (by ISD-P AID) is not possible when this Operational Profile is in Enabled state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | SW=0x6985 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |

Test Sequence #04 Error: Enable Profile (by ICCID) is not possible when this Operational Profile is in Enabled state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | SW=0x6985 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |

Test Sequence #05 Error: Enable Profile (by ISD-P AID) not possible when an Operational Profile with a PPR1 is loaded

The purpose of this test is to ensure that it is NOT possible to enable an Operational Profile when there is another Operational Profile Enabled with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Profile is installed on the eUICC  (this condition overrides the general initial condition defined in this test case). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL4  NOTE: The PROFILE\_OPERATIONAL4 corresponds to <ISD\_P\_AID4> | | | |
| IC4 | Install PROFILE\_OPERATIONAL1  NOTE: The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1> | | | |
| IC5 | Enable PROFILE\_OPERATIONAL4 | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | SW=0x6985 | RQ29\_002 RQ29\_022 RQ32\_011 RQ32\_012 RQ32\_014 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ29\_002 RQ32\_011 RQ32\_012 RQ32\_014 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |

Test Sequence #06 Error: Enable Profile (by ICCID) not possible with an Operational Profile with PPR1 is loaded

The purpose of this test is to ensure that it is NOT possible to enable an Operational Profile when there is another Operational Profile Enabled with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Profile is installed on the eUICC  (this condition overrides the general initial condition defined in this test case). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | Install PROFILE\_OPERATIONAL4 | | | | |
| IC4 | Install PROFILE\_OPERATIONAL1 | | | | |
| IC5 | Enable PROFILE\_OPERATIONAL4 | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | | SW=0x6985 | RQ29\_002 RQ29\_022 RQ32\_011 RQ32\_012 RQ32\_014 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA ( #GET\_PROFILES\_INFO\_ALL) | | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ29\_002 RQ32\_011 RQ32\_012 RQ32\_014 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_135\_2 RQ57\_138 RQ57\_140 |

Test Sequence #07 Error: Enable Profile by ISD-P AID without refreshFlag while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>,  FALSE)) | SW=0x9300 | RQ57\_127\_1RQ57\_140\_1 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1,  #PROFILE\_INFO2  }  SW=0x9000 |  |

Test Sequence #08 Error: Enable Profile by ICCID with refreshFLag set while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM,  TRUE)) | SW=0x9300 | RQ57\_133\_1RQ57\_140\_1 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1,  #PROFILE\_INFO2  }  SW=0x9000 |  |

##### 4.2.21.2.3 TC\_eUICC\_ES10c.EnableProfile\_Case4

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Nominal: Enable Profile by ISD-P AID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | #R\_ENABLE\_PROFILE\_OK  SW=0x91XX | RQ24\_010 RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ24\_010 RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #02 Nominal: Enable Profile by ICCID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | #R\_ENABLE\_PROFILE\_OK  SW=0x91XX | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #03 Nominal: Enable Profile by ISD-P AID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | #R\_ENABLE\_PROFILE\_OK  SW=0x91XX | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 RQ24\_010 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State change”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 RQ24\_010 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 8 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #04 Nominal: Enable Profile by ICCID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | #R\_ENABLE\_PROFILE\_OK  SW=0x91XX | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State change”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ32\_016\_2 RQ57\_128 RQ57\_129 RQ57\_133\_3 RQ57\_136 RQ57\_137 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 8 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #05 Nominal: Enable Profile by ISD-P AID and “refreshFlag” not set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | #R\_ENABLE\_PROFILE\_OK  SW=0x9000 | RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_136 RQ57\_137 RQ57\_132\_1 RQ24\_010 |
| 2 | S\_Device 🡪 eUICC | [TERMINAL\_PROFILE] | Toolkit initialization THEN SW=0x9000 | RQ32\_018\_1RQ57\_135\_4 |
| 3 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_136 RQ57\_137 RQ57\_132\_1 RQ24\_010 |
| 4 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 5 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #06 Nominal: Enable Profile by ICCID and “refreshFlag” not set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  FALSE)) | #R\_ENABLE\_PROFILE\_OK  SW=0x9000 | RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_136 RQ57\_137 RQ57\_132\_1 |
| 2 | S\_Device 🡪 eUICC | [TERMINAL\_PROFILE] | Toolkit initialization THEN SW=0x9000 | RQ32\_018\_1 RQ57\_135\_4 |
| 3 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_017\_1 RQ32\_017\_2 RQ57\_128 RQ57\_129 RQ57\_132 RQ57\_136 RQ57\_137 RQ57\_132\_1 |
| 4 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  |
| 5 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF1  SW=0x9000 | RQ34\_015 |

Test Sequence #07 Nominal: Enable Profile by ISD-P AID and “refreshFlag” set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| IC4 | Do not send FETCH command | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>,  TRUE)) | resp EnableProfileResponse ::= {  enableResult ok  }  SW=0x91YY | RQ32\_011 | |
| 2 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 | |
| 3 | Repeat IC1 and IC2 | | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 | |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  | |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 | |

Test Sequence #08 Nominal: Enable Profile by ISD-P AID and “refreshFlag” not set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| IC4 | Do not send FETCH command | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>,  FALSE)) | resp EnableProfileResponse ::= {  enableResult ok  }  SW=0x9000 |  | |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 | |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  | |
| 4 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 | |

Test Sequence #09 Nominal: Enable Profile by ISD-P AID and “refreshFlag” set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| 1 | S\_Device → eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  | |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>,  TRUE)) | resp EnableProfileResponse ::= {  enableResult ok  }  SW=0x9000 | RQ32\_011 | |
| 3 | S\_Device → eUICC | TERMINAL RESPONSE | SW=0x91YY |  | |
| 4 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 | |
| 5 | Repeat IC1 and IC2 | | | | |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 | |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  | |
| 8 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 | |

Test Sequence #10 Nominal: Enable Profile by ISD-P AID and “refreshFlag” not set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| 1 | S\_Device → eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  | |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>,  FALSE)) | resp EnableProfileResponse ::= {  enableResult ok  }  SW=0x9000 |  | |
| 3 | S\_Device → eUICC | TERMINAL RESPONSE | SW= any value except91XX |  | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ32\_011 | |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x9000 |  | |
| 6 | S\_Device 🡪 eUICC | [READ\_BINARY] with <L>=0x0A | #ICCID\_OP\_PROF2  SW=0x9000 | RQ34\_015 | |

##### 4.2.21.2.4 TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case4

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Error: Enable Profile by an unknown ISD-P AID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The Operational Profile identified by the ISD-P AID <ISD\_P\_AIDX> is not loaded. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AIDX>,  TRUE)) | #R\_ENABLE\_PROFILE\_ICCID\_ISDP\_NOTFOUND  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_136 RQ57\_137 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_136 RQ57\_137 |

Test Sequence #02 Error: Enable Profile by an unknown ICCID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The Operational Profile identified by the ICCID #ICCID\_OP\_PROFX is not loaded. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROFX,  NO\_PARAM,  TRUE)) | #R\_ENABLE\_PROFILE\_ICCID\_ISDP\_NOTFOUND  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_136 RQ57\_137 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_011 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_130 RQ57\_136 RQ57\_137 |

Test Sequence #03 Error: Enable Profile (by ISD-P AID) is not possible when this Operational Profile is in Enable state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | #R\_ENABLE\_PROFILE\_NOT\_DISABLE\_STATE  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_136 RQ57\_137 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_136 RQ57\_137 |

Test Sequence #04 Error: Enable Profile (by ICCID) is not possible when this Operational Profile is in Enabled state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | #R\_ENABLE\_PROFILE\_NOT\_DISABLE\_STATE  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_136 RQ57\_137 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_011 RQ32\_012 RQ32\_016\_1 RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_136 RQ57\_137 |

Test Sequence #05 Error: Enable Profile (by ISD-P AID) not possible when an Operational Profile with PPR1 is loaded

The purpose of this test is to ensure that it is NOT possible to enable an Operational Profile when there is another Operational Profile Enabled with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Profile is installed on the eUICC  (this condition overrides the general initial condition defined in this test case). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL4  NOTE: The PROFILE\_OPERATIONAL4 corresponds to <ISD\_P\_AID4> | | | |
| IC4 | Install PROFILE\_OPERATIONAL1  NOTE: The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1> | | | |
| IC5 | Enable PROFILE\_OPERATIONAL4 | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA (  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | #R\_ENABLE\_PROFILE\_DISALLOWEDbyPOLICY  SW=0x9000 | RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_136 RQ57\_147 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ57\_127  RQ57\_128  RQ57\_129 |

Test Sequence #06 Error: Enable Profile (by ICCID) not possible when an Operational Profile with PPR1 is loaded

The purpose of this test is to ensure that it is NOT possible to enable an Operational Profile when there is another Operational Profile Enabled with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No Profile is installed on the eUICC  (this condition overrides the general initial condition defined in this test case). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | Install PROFILE\_OPERATIONAL4 | | | |
| IC4 | Install PROFILE\_OPERATIONAL1 | | | |
| IC5 | Enable PROFILE\_OPERATIONAL4 | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | #R\_ENABLE\_PROFILE\_DISALLOWEDbyPOLICY  SW=0x9000 | RQ57\_127 RQ57\_128 RQ57\_129 RQ57\_136 RQ57\_147 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ57\_127 RQ57\_128 RQ57\_129 |

Test Sequence #07 Error: Enable Profile by ISD-P AID without refreshFlag while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| IC4 | Do not send FETCH command | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>,  FALSE)) | resp EnableProfileResponse ::= {  enableResult catBusy  }  SW=0x9000 or 0x91XX | RQ57\_127\_1 | |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  | |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1,  #PROFILE\_INFO2  }  SW=0x9000 |  | |

Test Sequence #08 Error: Enable Profile by ICCID with refreshFlag set while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_ENABLE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM,  TRUE)) | resp EnableProfileResponse ::= {  enableResult catBusy  }  SW=0x9000 or 0x91XX | RQ57\_133\_1 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1,  #PROFILE\_INFO2  }  SW=0x9000 |  |

### 4.2.22 ES10c (LPA -- eUICC): DisableProfile

#### 4.2.22.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_025

 RQ29\_002, RQ29\_022

 RQ32\_031, RQ32\_032, RQ32\_033, RQ32\_034, RQ32\_038, RQ32\_037\_1, RQ32\_039, RQ32\_039\_1, RQ32\_041\_1, RQ32\_041\_2

 RQ57\_141, RQ57\_142, RQ57\_142\_1, RQ57\_142\_2, RQ57\_142\_3, RQ57\_142\_4, RQ57\_142\_6, RQ57\_142\_9, RQ57\_142\_10, RQ57\_142\_12, RQ57\_142\_13, RQ57\_142\_14, RQ57\_149, RQ57\_150, RQ57\_151, RQ57\_152, RQ57\_153, RQ57\_153\_1

#### 4.2.22.2 Test Cases

##### 4.2.22.2.1 TC\_eUICC\_ES10c.DisableProfile\_Case3

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Nominal: Disable Profile by ISD-P AID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | No response data is returned  SW=0x91XX | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_151 RQ24\_010 | |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_038 RQ32\_039 RQ32\_039\_1 RQ32\_041\_2 RQ57\_142\_13 RQ57\_142\_14 | |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_151 RQ24\_010 | |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 | |

Test Sequence #02 Nominal: Disable Profile by ICCID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | No response data is returned  SW=0x91XX | RQ32\_031  RQ32\_033  RQ32\_037\_1  RQ57\_142\_2  RQ57\_142\_3  RQ57\_142\_12  RQ57\_151 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_038  RQ32\_039  RQ32\_039\_1  RQ32\_041\_2  RQ57\_142\_13  RQ57\_142\_14 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031  RQ32\_033  RQ32\_037\_1  RQ57\_142\_2  RQ57\_142\_3  RQ57\_142\_12  RQ57\_151 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 |

Test Sequence #03 Nominal: Disable Profile by ISD-P AID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | No response data is returned  SW=0x91XX | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_151 RQ24\_010 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State changed”) | RQ32\_038 RQ32\_039 RQ32\_039\_1 RQ32\_041\_2 RQ57\_142\_13 RQ57\_142\_14 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_151 RQ24\_010 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 |

Test Sequence #04 Nominal: Disable Profile by ICCID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | No response data is returned  SW=0x91XX | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_151 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State changed”) | RQ32\_038 RQ32\_039 RQ32\_039\_1 RQ32\_041\_2 RQ57\_142\_13 RQ57\_142\_14 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_151 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 |

Test Sequence #05 Nominal: Disable Profile by ISD-P AID and “refreshFlag” no set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | No response data is returned  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_151 RQ29\_002 RQ29\_022 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : { #PROFILE\_INFO1\_DISABLED    }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_151 |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

Test Sequence #06 Nominal: Disable Profile by ICCID and “refreshFlag” no set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  FALSE)) | No response data is returned  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_151 RQ29\_002 RQ29\_022 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_151 |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

Test Sequence #07 Nominal: Disable Profile by ICCID with refreshFLag set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | No response data is returned  SW=0x91YY |  |
| 2 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

Test Sequence #08 Nominal: Disable Profile by ICCID with refreshFLag not set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  FALSE)) | No response data is returned  SW=0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

Test Sequence #09 Nominal: Disable Profile by ICCID with refreshFLag set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| 1 | S\_Device 🡪eUICC | FETCH ‘XX’ | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | No response data is returned  SW=0x9000 |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x91YY |  |
| 4 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 |
| 5 | Repeat IC1 and IC2 | | | |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

Test Sequence #10 Nominal: Disable Profile by ICCID with refreshFLag not set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| 1 | S\_Device 🡪eUICC | FETCH ‘XX’ | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  FALSE)) | No response data is returned  SW=0x9000 |  |
| 3 | S\_Device 🡪eUICC | TERMINAL RESPONSE | SW= any value except91XX |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

##### 4.2.22.2.2 TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case3

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Error: Disable Profile by an unknown ISD-P AID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The Operational Profile identified by the ISD-P AID <ISD\_P\_AIDX> is not loaded. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AIDX>,  TRUE)) | SW=0x6A82 | RQ32\_031 RQ32\_033 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_151 RQ57\_152 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_151 RQ57\_152 |

Test Sequence #02 Error: Disable Profile by an unknown ICCID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The Operational Profile identified by the ICCID #ICCID\_OP\_PROFX is not loaded. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROFX,  NO\_PARAM,  TRUE)) | SW=0x6A82 | RQ32\_031 RQ32\_033 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_151 RQ57\_152 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_151 RQ57\_152 |

Test Sequence #03 Error: Disable Profile (by ISD-P AID) is not possible when this Operational Profile is in Disabled state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | SW=0x6985 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 |

Test Sequence #04 Error: Disable Profile (by ICCID) is not possible when this Operational Profile is in Disabled state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | SW=0x6985 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 | |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 | |

Test Sequence #05 Error: Disable Profile (by ISD-P AID) not possible when PPR1 is set

The purpose of this test is to ensure that it is NOT possible to disable an Operational Profile4 with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded  (this condition overrides the general initial condition defined in this test case). |
| eUICC | The PROFILE\_OPERATIONAL4 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 corresponds to <ISD\_P\_AID4>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID4>,  TRUE)) | SW=0x6985 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID4>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 |

Test Sequence #06 Error: Disable Profile (by ICCID) not possible when PPR1 is set

The purpose of this test is to ensure that it is NOT possible to disable an Operational Profile4 with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded  (this condition overrides the general initial condition defined in this test case). |
| eUICC | The PROFILE\_OPERATIONAL4 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF4,  NO\_PARAM,  TRUE)) | SW=0x6985 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF4,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_151 RQ57\_153 |

Test Sequence #07 Error: Disable Profile by ISDP-AID without refreshFlag while proactive session is ongoing –catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | SW=0x9300 | RQ57\_142 RQ57\_153\_1 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 |  |

Test Sequence #08 Error: Disable Profile by ICCID with refreshFlag set while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | SW=0x9300 | RQ57\_142\_10 RQ57\_153\_1 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 |  |

##### 4.2.22.2.3 TC\_eUICC\_ES10c.DisableProfile\_Case4

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Nominal: Disable Profile by ISD-P AID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | #R\_DISABLE\_PROFILE\_OK  SW=0x91XX | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 RQ24\_010 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_038 RQ32\_039 RQ32\_039\_1 RQ32\_041\_2 RQ57\_142\_13 RQ57\_142\_14 RQ57\_147 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 RQ24\_010 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 |

Test Sequence #02 Nominal: Disable Profile by ICCID and “refreshFlag” set when Device supports “UICC Reset”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | #R\_DISABLE\_PROFILE\_OK  SW=0x91XX | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ32\_038 RQ32\_039 RQ32\_039\_1 RQ32\_041\_2 RQ57\_142\_13 RQ57\_142\_14 RQ57\_147 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 |

Test Sequence #03 Nominal: Disable Profile by ISD-P AID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | #R\_DISABLE\_PROFILE\_OK  SW=0x91XX | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 RQ24\_010 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State changed”) | RQ32\_038 RQ32\_039 RQ32\_039\_1 RQ32\_041\_2 RQ57\_142\_13 RQ57\_142\_14 RQ57\_147 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 RQ24\_010 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 |

Test Sequence #04 Nominal: Disable Profile by ICCID and “refreshFlag” set when Device supports “eUICC Profile State Change”

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | #R\_DISABLE\_PROFILE\_OK  SW=0x91XX | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“eUICC Profile State changed”) | RQ32\_038 RQ32\_039 RQ32\_039\_1 RQ32\_041\_2 RQ57\_142\_13 RQ57\_142\_14 RQ57\_147 |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | Execute IC1 from step 2 to step 4 | | | |
| 5 | Repeat IC2 | | | |
| 6 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_037\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_12 RQ57\_149 RQ57\_150 |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=6A82 | RQ24\_025 |

Test Sequence #05 Nominal: Disable Profile by ISD-P AID and “refreshFlag” no set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | #R\_DISABLE\_PROFILE\_OK  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_149 RQ57\_150 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED    }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_149 RQ57\_150 |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

Test Sequence #06 Nominal: Disable Profile by ICCID and “refreshFlag” no set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  FALSE)) | #R\_DISABLE\_PROFILE\_OK  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_149 RQ57\_150 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_033 RQ32\_038 RQ32\_041\_1 RQ57\_142\_1 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_6 RQ57\_142\_9 RQ57\_142\_14 RQ57\_149 RQ57\_150 |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 |

Test Sequence #07 Nominal: Disable Profile by ISD-P AID and “refreshFlag” set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| IC4 | Do not send FETCH command | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | resp DisableProfileResponse ::= {  DisableResult ok  }  SW=0x91YY |  | |
| 2 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 | |
| 3 | Repeat IC1 and IC2 | | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  | |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 | |

Test Sequence #08 Nominal: Disable Profile by ISD-P AID and “refreshFlag” not set while proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| IC4 | Do not send FETCH command | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | resp DisableProfileResponse ::= {  DisableResult ok  }  SW=0x9000 |  | |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  | |
| 3 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 | |

Test Sequence #09 Nominal: Disable Profile by ISD-P AID and “refreshFlag” set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| 1 | S\_Device 🡪eUICC | FETCH ‘XX’ | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  | |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | resp DisableProfileResponse ::= {  DisableResult ok  }  SW=0x9000 |  | |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x91YY |  | |
| 4 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | RQ32\_016\_3 RQ32\_017 RQ57\_134 | |
| 5 | Repeat IC1 and IC2 | | | | |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  | |
| 7 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 | |

Test Sequence #10 Nominal: Disable Profile by ISD-P AID and “refreshFlag” not set while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| 1 | S\_Device 🡪eUICC | FETCH ‘XX’ | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  | |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | resp DisableProfileResponse ::= {  DisableResult ok  }  SW=0x9000 |  | |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW= any value except 91XX |  | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  | |
| 5 | S\_Device 🡪 eUICC | [SELECT\_ICCID] | SW=0x6A82 | RQ24\_025 | |

##### 4.2.22.2.4 TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case4

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Error: Disable Profile by an unknown ISD-P AID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The Operational Profile identified by the ISD-P AID <ISD\_P\_AIDX> is not loaded. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AIDX>,  TRUE)) | #R\_DISABLE\_PROFILE\_ICCID\_ISDP\_NOTFOUND  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 | |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 | |

Test Sequence #02 Error: Disable Profile by an unknown ICCID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The Operational Profile identified by the ICCID #ICCID\_OP\_PROFX is not loaded. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROFX,  NO\_PARAM,  TRUE)) | #R\_DISABLE\_PROFILE\_ICCID\_ISDP\_NOTFOUND  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 |

Test Sequence #03 Error: Disable Profile (by ISD-P AID) is not possible when this Operational Profile is in Disabled state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  TRUE)) | #R\_DISABLE\_PROFILE\_NOT\_ENABLE\_STATE  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 |

Test Sequence #04 Error: Disable Profile (by ICCID) is not possible when this Operational Profile is in Disabled state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | #R\_DISABLE\_PROFILE\_NOT\_ENABLE\_STATE  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_2 RQ57\_142\_15 RQ57\_149 RQ57\_150 |

Test Sequence #05 Error: Disable Profile (by ISD-P AID) not possible when PPR1 is set

The purpose of this test is to ensure that it is NOT possible to disable an Operational Profile with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded  (this condition overrides the general initial condition defined in this test case). |
| eUICC | The PROFILE\_OPERATIONAL4 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 corresponds to <ISD\_P\_AID4>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID4>,  TRUE)) | #R\_DISABLE\_PROFILE\_DISALLOWEDbyPOLICY  SW=0x9000 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_149 RQ57\_150 RQ57\_141 RQ57\_142 RQ57\_149 RQ57\_150 RQ29\_002 RQ29\_022 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID4>)**)** | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_149 RQ57\_150 |

Test Sequence #06 Error: Disable Profile (by ICCID) not possible when PPR1 is set

The purpose of this test is to ensure that it is NOT possible to disable an Operational Profile4 with the Policy Rule “Disabling of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded  (this condition overrides the general initial condition defined in this test case). |
| eUICC | The PROFILE\_OPERATIONAL4 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF4,  NO\_PARAM,  TRUE)) | #R\_DISABLE\_PROFILE\_DISALLOWEDbyPOLICY  SW=0x9000 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_149 RQ57\_150 RQ29\_002 RQ29\_022 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF4,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO4\_ENABLED  }  SW=0x9000 | RQ32\_031 RQ32\_032 RQ32\_033 RQ32\_034 RQ57\_142\_2 RQ57\_142\_3 RQ57\_142\_4 RQ57\_142\_15 RQ57\_149 RQ57\_150 |

Test Sequence #07 Error: Disable Profile by ISD-P AID without refreshFlag while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>,  FALSE)) | resp DisableProfileResponse ::= {  disableResult catBusy  }  SW=0x9000 or 0x91XX | RQ57\_142 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 |  |

Test Sequence #08 Error: DisableProfile by ICCID with refreshFlag set while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  |
| IC4 | Do not send FETCH command | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DISABLE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM,  TRUE)) | resp DisableProfileResponse ::= {  disableResult catBusy  }  SW=0x9000 or 0x91XX | RQ57\_142\_10 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1  }  SW=0x9000 |  |

### 4.2.23 ES10c (LPA -- eUICC): DeleteProfile

#### 4.2.23.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_016, RQ24\_020

 RQ29\_002, RQ29\_022

 RQ32\_049, RQ32\_050, RQ32\_051, RQ32\_052

 RQ57\_119, RQ57\_154, RQ57\_155, RQ57\_156, RQ57\_157, RQ57\_158, RQ57\_159, RQ57\_160, RQ57\_161, RQ57\_162

#### 4.2.23.2 Test Cases

##### 4.2.23.2.1 TC\_eUICC\_ES10c.DeleteProfile\_Case3

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Nominal: Delete Profile by ISD-P AID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>) | No response data is returned  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_051 RQ32\_052 RQ57\_154 RQ57\_160 RQ24\_010 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW=0x9000 | RQ57\_119 RQ24\_010 |

Test Sequence #02 Nominal: Delete Profile by ICCID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM) | No response data is returned  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_051 RQ32\_052 RQ57\_154 RQ57\_158 RQ57\_160 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW=0x9000 | RQ24\_020 RQ57\_119 |

##### 4.2.23.2.2 TC\_eUICC\_ES10c.DeleteProfile\_ErrorCases\_Case3

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |

Test Sequence #01 Error: Delete Profile not possible with unknown ISD-P AID

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile with an unknown ISD-P AID.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Operational Profile identified by the ISD-P AID <ISD\_P\_AIDX> is not loaded. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AIDX>) | SW=0x6A82 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_160 RQ57\_161 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_160 RQ57\_161 |

Test Sequence #02 Error: Delete Profile not possible with unknown ICCID

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile with an unknown ICCID.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Operational Profile identified by the ICCID #ICCID\_OP\_PROFX is not loaded. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROFX,  NO\_PARAM**)** | SW=0x6A82 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_160 RQ57\_161 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_160 RQ57\_161 |

Test Sequence #03 Error: Delete Profile (by ISD-P AID) not possible when this Operational Profile is in Enabled state

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile in Enabled state.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>) | SW=0x6985 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_160 RQ57\_162 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA ( #GET\_PROFILES\_INFO\_ALL) | profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_160 RQ57\_162 |

Test Sequence #04 Error: Delete Profile (by ICCID) not possible when this Operational Profile is in Enabled state

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile in Enabled state.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM**)** | SW=0x6985 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_160 RQ57\_162 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_160 RQ57\_162 |

Test Sequence #05 Error: Delete Profile (by ISD-P AID) not possible when PPR2 is set

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile with the Policy Rule “Deletion of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 corresponds to <ISD\_P\_AID3>. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL3 is equal to #NICKNAME3. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID3>) | SW=0x6985 | RQ24\_016 RQ29\_002 RQ29\_022 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_156 RQ57\_160 RQ57\_162 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED,  #PROFILE\_INFO3  }  SW=0x9000 | RQ57\_154 RQ57\_156 RQ57\_160 RQ57\_162 |

Test Sequence #06 Error: Delete Profile (by ICCID) not possible when PPR2 is set

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile with the Policy Rule “Deletion of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL3 is equal to #NICKNAME3. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_Case3(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROF3,  NO\_PARAM) | SW=0x6985 | RQ24\_016 RQ29\_002 RQ29\_022 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_156 RQ57\_160 RQ57\_162 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED,  #PROFILE\_INFO3  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_156 RQ57\_160 RQ57\_162 |

##### 4.2.23.2.3 TC\_eUICC\_ES10c.DeleteProfile\_Case4

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |

Test Sequence #01 Nominal: Delete Profile by ISD-P AID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID1>) | #R\_DELETE\_PROFILE\_OK  SW=0x9000 | RQ24\_010 RQ24\_016 RQ24\_020 RQ32\_049 RQ32\_051 RQ32\_052 RQ57\_154 RQ57\_158 RQ57\_159 RQ57\_160 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  NO\_PARAM,  <ISD\_P\_AID1>)**)** | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW=0x9000 | RQ24\_010 RQ24\_020 RQ57\_119 |

Test Sequence #02 Nominal: Delete Profile by ICCID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROF1,  NO\_PARAM) | #R\_DELETE\_PROFILE\_OK  SW=0x9000 | RQ24\_016 RQ24\_020 RQ32\_049 RQ32\_051 RQ32\_052 RQ57\_154 RQ57\_158 RQ57\_159 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW=0x9000 | RQ24\_020 RQ57\_119 |

##### 4.2.23.2.4 TC\_eUICC\_ES10c.DeleteProfile\_ErrorCases\_Case4

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |

Test Sequence #01 Error: Delete Profile not possible with unknown ISD-P AID

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile with an unknown ISD-P AID.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | A Operational Profile identified by the ISD-P AID <ISD\_P\_AIDX> is not loaded. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AIDX>) | #R\_DELETE\_PROFILE\_ICCID\_ISDP\_NOTFOUND  SW=0x9000 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_158 RQ57\_159 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_158 RQ57\_159 |

Test Sequence #02 Error: Delete Profile not possible with unknown ICCID

The purpose of this test is to ensure that it is NOT possible to delete an Operational with an ICCID unknown.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Operational Profile identified by the ICCID #ICCID\_OP\_PROFX is not loaded. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROFX,  NO\_PARAM**)** | #R\_DELETE\_PROFILE\_ICCID\_ISDP\_NOTFOUND  SW=0x9000 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_158 RQ57\_159 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ57\_154 RQ57\_157 RQ57\_158 RQ57\_159 |

Test Sequence #03 Error: Delete Profile (by ISD-P AID) not possible when this Operational Profile is in Enabled state

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile in Enabled state.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 corresponds to <ISD\_P\_AID1>. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 corresponds to <ISD\_P\_AID2>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID2>) | #R\_DELETE\_PROFILE\_NOTDISABLESTATE  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_158 RQ57\_159 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_158 RQ57\_159 |

Test Sequence #04 Error: Delete Profile (by ICCID) not possible when this Operational Profile is in Enabled state

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile in Enabled state.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROF2,  NO\_PARAM**)** | #R\_DELETE\_PROFILE\_NOTDISABLESTATE  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_158 RQ57\_159 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED  }  SW=0x9000 | RQ24\_016 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_158 RQ57\_159 |

Test Sequence #05 Error: Delete Profile (by ISD-P AID) not possible when PPR2 is set

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile with the Policy Rule “Deletion of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 corresponds to <ISD\_P\_AID3>. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL3 is equal to #NICKNAME3. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  NO\_PARAM,  <ISD\_P\_AID3>) | #R\_DELETE\_PROFILE\_DISALLOWEDBYPOLICY  SW=0x9000 | RQ24\_016 RQ29\_002 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_156 RQ57\_158 RQ57\_159 RQ29\_002 RQ29\_022 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED, #PROFILE\_INFO3  }  SW=0x9000 | RQ24\_016 RQ29\_002 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_158 RQ57\_159 |

Test Sequence #06 Error: Delete Profile (by ICCID) not possible when PPR2 is set

The purpose of this test is to ensure that it is NOT possible to delete an Operational Profile with the Policy Rule “Deletion of this Profile is not allowed”.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL3 is equal to #NICKNAME3. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_DELETE\_PROFILE(  #ICCID\_OP\_PROF3,  NO\_PARAM) | #R\_DELETE\_PROFILE\_DISALLOWEDBYPOLICY  SW=0x9000 | RQ24\_016 RQ29\_002 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_156 RQ57\_158 RQ57\_159 RQ29\_002 RQ29\_022 |
| 2 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO2\_ENABLED,  #PROFILE\_INFO3  }  SW=0x9000 | RQ24\_016 RQ29\_002 RQ32\_049 RQ32\_050 RQ57\_154 RQ57\_155 RQ57\_158 RQ57\_159 |

### 4.2.24 ES10c (LPA -- eUICC): eUICCMemoryReset

#### 4.2.24.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_020

 RQ29\_005

 RQ31\_027, RQ31\_028

 RQ33\_011, RQ33\_008, RQ33\_009, RQ33\_010, RQ33\_012

 RQ35\_006

 RQ57\_051, RQ57\_052, RQ57\_054, RQ57\_163, RQ57\_165, RQ57\_165\_1, RQ57\_166, RQ57\_167, RQ57\_167\_1, RQ57\_168, RQ57\_169, RQ57\_170

#### 4.2.24.2 Test Cases

##### 4.2.24.2.1 TC\_eUICC\_ES10c.eUICCMemoryReset

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |
| eUICC | The Default SM-DP+ Address #TEST\_DP\_ADDRESS1 has been set on the ISD-R. |

Test Sequence #01 Nominal: Reset All Operational Profiles (without Enabled Profile)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL3 is equal to #NICKNAME3. |
| eUICC | No Notification is stored in the eUICC's Pending Notifications List. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | Retrieve free non-volatile memory value (tag 0x82) from <EXT\_CARD\_RESOURCE> in EUICCInfo2 as <FREE\_MEM\_OP\_PROF\_INSTALLED> |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET\_OP\_PRO) | #R\_EUICC\_MEMORY\_RESET\_OK  SW=0x9000 | RQ57\_163 RQ57\_166 RQ57\_169 RQ57\_170 RQ33\_010 |
| 3 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW=0x9000 | RQ33\_011 RQ33\_008 RQ33\_012 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000 | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_167\_1 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | Retrieve free non-volatile memory value (tag 0x82) from <EXT\_CARD\_RESOURCE> in EUICCInfo2 as <FREE\_MEMORY\_NO\_PROFILE>  Verify that <FREE\_MEM\_OP\_PROF\_INSTALLED> is lower than <FREE\_MEMORY\_NO\_PROFILE> | RQ31\_027 RQ31\_028 RQ57\_051 RQ57\_052 RQ57\_054 RQ24\_020 |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS\_DP\_1  SW = 0x9000 | RQ33\_009 |

Test Sequence #02 Nominal: Reset All Operational Profiles (with Enabled Profile)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | No Notification is stored in the eUICC's Pending Notifications List. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET\_OP\_PRO) | #R\_EUICC\_MEMORY\_RESET\_OK  SW=0x91XX | RQ57\_163 RQ57\_166 RQ57\_169RQ57\_170 RQ33\_010 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | REFRESH Command (“UICC Reset”) | RQ57\_168 |
| 3 | Repeat IC1 and IC2 | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000  NOTE : A Disable Notification for PROFILE\_OPERATIONAL1 MAY be also present in the response. | RQ57\_071 RQ57\_071\_1 RQ57\_071\_2 RQ57\_072 RQ57\_072\_1 RQ57\_072\_2 RQ57\_074 RQ57\_167\_1RQ35\_006 |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_RAT) | #R\_DEFAULT\_RAT  SW = 0x9000 | RQ29\_005 RQ57\_179 RQ57\_180 RQ57\_181 RQ57\_182 RQ57\_184 |
| 6 | S\_LPAd 🡪 eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW=0x9000 | RQ33\_011 RQ33\_008 RQ33\_012 |

Test Sequence #03 Nominal: Reset the Default SM-DP+ Address only

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL3 is equal to #NICKNAME3. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET\_DEF\_SMDPADDRESS) | The response data is one of:  resp EuiccMemoryResetResponse::= {  resetResult nothingToDelete  }  or  resp EuiccMemoryResetResponse::= {  resetResult ok  }  SW=0x9000 | RQ57\_163 RQ57\_167 RQ57\_169RQ57\_170 RQ33\_010 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1\_DISABLED,  #PROFILE\_INFO3  }  SW=0x9000 |  |
| 3 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS  SW = 0x9000 | RQ33\_008 |

Test Sequence #04 Nominal: Reset All Operational Profiles and the Default SM-DP+ Address

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET) | #R\_EUICC\_MEMORY\_RESET\_OK  SW=0x9000 | RQ57\_163 RQ57\_166 RQ57\_167 RQ57\_169RQ57\_170 RQ33\_010 |
| 2 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  }  SW=0x9000 | RQ33\_011 RQ33\_008 RQ33\_012 |
| 3 | S\_LPAd 🡪eUICC | MTD\_STORE\_DATA( #GET\_EUICC\_CONFIGURED\_ADDRESSES) | #R\_ES10a\_GECA\_DS  SW = 0x9000 | RQ33\_008 |

Test Sequence #05 Nominal: eUICC Memory Reset, one Operational Profile Enabled, proactive session is ongoing – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | No Notification is stored in the eUICC's Pending Notifications List. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| IC4 | Do not send FETCH command | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET\_OP\_PRO) | #R\_EUICC\_MEMORY\_RESET\_OK  SW=0x91YY | | RQ33\_012 |
| 2 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | |  |
| 3 | Repeat IC1 and IC2 | | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000  NOTE : A Disable Notification for PROFILE\_OPERATIONAL1 MAY be also present in the response. | |  |
| 5 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_RAT) | #R\_DEFAULT\_RAT  SW = 0x9000 | |  |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  }  SW=0x9000 | | RQ33\_012 |

Test Sequence #06 Nominal: eUICC Memory Reset (with Enabled Profile) while proactive session is ongoing with Terminal Response outstanding – catBusy not supported

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL3 is Disabled on the eUICC. |
| eUICC | No Notification is stored in the eUICC's Pending Notifications List. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| 1 | S\_Device → eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK | |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET\_OP\_PRO) | #R\_EUICC\_MEMORY\_RESET\_OK  SW=0x9000 | | RQ33\_012 |
| 3 | S\_Device → eUICC | TERMINAL RESPONSE | SW=0x91YY | |  |
| 4 | S\_Device 🡪eUICC | FETCH 'YY' | REFRESH Command (“UICC Reset”) | |  |
| 5 | Repeat IC1 and IC2 | | | | |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_DE1 SW = 0x9000  NOTE : A Disable Notification for PROFILE\_OPERATIONAL1 MAY be also present in the response. | |  |
| **7** | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_RAT) | #R\_DEFAULT\_RAT  SW = 0x9000 | |  |
| 8 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  }  SW=0x9000 | | RQ33\_012 |

##### 4.2.24.2.2 TC\_eUICC\_ES10c.eUICCMemoryReset\_ErrorCases

Test Sequence #01 Error: eUICC Memory Reset while proactive session is ongoing – catBusy supported

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 has been installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled on the eUICC. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [GET\_MNO\_SD]) | SW=0x91XX |  | |
| IC4 | Do not send FETCH command | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET\_OP\_PRO) | resp EuiccMemoryResetResponse::= {  resetResult catBusy  }  SW=0x9000 or 0x91XX | | RQ57\_165\_1 |
| 2 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x00 – POR OK | |  |
| 3 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 | |  |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1,  #PROFILE\_INFO2  }  SW=0x9000 | | RQ57\_165 |

Test Sequence #02 Error: Nothing to delete

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | No Profile is loaded on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #EUICC\_MEMORY\_RESET\_OP\_PRO) | resp EuiccMemoryResetResponse::= {  resetResult nothingToDelete  }  SW=0x9000 | RQ57\_163 |

### 4.2.25 ES10c (LPA -- eUICC): GetEID

#### 4.2.25.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ33\_002

 RQ57\_171, RQ57\_172, RQ57\_172\_1

#### 4.2.25.2 Test Cases

##### 4.2.25.2.1 TC\_eUICC\_ES10c.GetEID

Test Sequence #01 Nominal

The purpose of this test is to ensure that it is possible to retrieve the EID.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EID) | resp GetEuiccDataResponse ::= {  eidValue #EID1  }  SW=0x9000 | RQ33\_002 RQ57\_171 RQ57\_172 |

Test Sequence #02 Error

The purpose of this test is to ensure that if the provided tagList is invalid or unsupported, the eUICC returns an error status word.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EID\_INVALID) | No response data return and SW different than 0x9000 | RQ33\_002 RQ57\_172\_1 |

### 4.2.26 ES10c (LPA -- eUICC): SetNickname

#### 4.2.26.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ57\_173, RQ57\_174, RQ57\_175, RQ57\_176, RQ57\_177, RQ57\_178

#### 4.2.26.2 Test Cases

##### 4.2.26.2.1 TC\_eUICC\_ES10c.SetNickname

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is loaded on the eUICC. |

Test Sequence #01 Nominal: Add a Nickname to a Disabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is empty. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #SET\_NICKNAME\_OP\_PROF1) | resp SetNicknameResponse ::= {  setNicknameResult ok  }  SW=0x9000 | RQ57\_177 RQ57\_178 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  profileNickname #NICKNAME2  …  }  }  SW=0x9000 | RQ57\_174 |

Test Sequence #02 Nominal: Update a Nickname of a Disabled Operational Profile

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is equal to #NICKNAME1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #SET\_NICKNAME\_OP\_PROF1) | resp SetNicknameResponse ::= {  setNicknameResult ok  }  SW=0x9000 | RQ57\_177 RQ57\_178 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  profileNickname #NICKNAME2  …  }  }  SW=0x9000 | RQ57\_174 |

Test Sequence #03 Nominal: Remove a Nickname from a Disabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is equal to #NICKNAME1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #SET\_NICKNAME\_EMPTY\_OP\_PROF1) | resp SetNicknameResponse ::= {  setNicknameResult ok  }  SW=0x9000 | RQ57\_177 RQ57\_178 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  *-- profileNickname SHALL not*  *-- be present*  …  }  }  SW=0x9000 | RQ57\_175 |

Test Sequence #04 Nominal: Remove a non-existing Nickname from a Disabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is empty. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #SET\_NICKNAME\_EMPTY\_OP\_PROF1) | resp SetNicknameResponse ::= {  setNicknameResult ok  }  SW=0x9000 | RQ57\_177 RQ57\_178 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  MTD\_GET\_PROFILE\_INFO(  #ICCID\_OP\_PROF1,  NO\_PARAM)) | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {   …  *-- profileNickname SHALL not*  *-- be present*  …  }  }  SW=0x9000 | RQ57\_176 |

Test Sequence #05 Nominal: Add a Nickname to an Enabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is empty. |

This test sequence SHALL be the same as the Test Sequence #01 defined in this section.

Test Sequence #06 Nominal: Update a Nickname of an Enabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is equal to #NICKNAME1. |

This test sequence SHALL be the same as the Test Sequence #02 defined in this section.

Test Sequence #07 Nominal: Remove a Nickname from an Enabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is equal to #NICKNAME1. |

This test sequence SHALL be the same as the Test Sequence #03 defined in this section.

Test Sequence #08 Nominal: Remove a non-existing Nickname from an Enabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is empty. |

This test sequence SHALL be the same as the Test Sequence #04 defined in this section.

Test Sequence #09 Error: ICCID not found

The purpose of this test is to ensure that the method ES10c.SetNickname returns an error in case the targeted Profile does not exist on the eUICC.

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | The Profile identified by the ICCID #ICCID\_UNKNOWN is not present on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA( #SET\_NICKNAME\_ICCID\_UNKNOWN) | resp SetNicknameResponse ::= {  setNicknameResult iccidNotFound  }  SW=0x9000 | RQ57\_173 RQ57\_178 |

### 4.2.27 ES10b (LPA -- eUICC): GetRAT

#### 4.2.27.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ28\_001

 RQ29\_006, RQ29\_007, RQ29\_007\_1, RQ29\_008, RQ29\_008\_1, RQ29\_009, RQ29\_010\_1, RQ29\_011, RQ29\_012, RQ29\_016, RQ29\_022

 RQ57\_179, RQ57\_180, RQ57\_181 , RQ57\_182, RQ57\_184, RQ57\_186

#### 4.2.27.2 Test Cases

##### 4.2.27.2.1 TC\_eUICC\_ES10b.GetRAT

Test Sequence #01 Nominal: Get Default RAT

The purpose of this test is to verify that the eUICC can be configured with a RAT as defined in SGP.21 [3] Annex H.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The EUM has configured the eUICC's RAT as defined in section G.2.4. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_RAT) | #R\_DEFAULT\_RAT  SW = 0x9000 | RQ28\_001 RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_011 RQ29\_016 RQ57\_179 RQ57\_180 RQ57\_181 RQ57\_182 RQ57\_184 RQ57\_186 RQ29\_007 |

Test Sequence #02 Nominal: With additional PPARs

The purpose of this test is to verify that the eUICC can be configured with a RAT that contains custom rules reflecting agreements between some Operators and OEMs. After having checked the content of the RAT, Profiles with PPR1 and PPR2 are installed in order to make sure that the eUICC accepts such PPRs.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The EUM has configured the eUICC's RAT as defined in section G.2.5. |
| eUICC | There is no Profile installed in the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_RAT) | #R\_RAT\_WITH\_OTHER\_RULES with exact same structure and order  SW = 0x9000 | RQ28\_001 RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_010\_1 RQ29\_011 RQ29\_016 RQ57\_179 RQ57\_180 RQ57\_181 RQ57\_182 RQ57\_184 RQ57\_186 RQ29\_007 RQ29\_008\_1 |
| 2 | S\_LPAd → eUICC | Install PROFILE\_OPERATIONAL4 | Profile successfully downloaded (i.e. ProfileInstallationResult contains a SuccessResult) | RQ29\_010\_1 RQ29\_022 RQ29\_008\_1 |
| 3 | S\_LPAd → eUICC | Delete PROFILE\_OPERATIONAL4 |  |  |
| 4 | S\_LPAd → eUICC | Install PROFILE\_OPERATIONAL3 | Profile successfully downloaded (i.e. ProfileInstallationResult contains a SuccessResult) | RQ29\_010\_1 RQ29\_022 RQ29\_008\_1 |

## 4.3 SM-DP+ Interfaces

### 4.3.1 ES2+ (Operator -- SM-DP+): DownloadOrder

#### 4.3.1.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

* Section 5.3.1

#### 4.3.1.2 Test Cases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Available’ state for S\_MNO |
| S\_MNO | For the TLS connection, CERT\_CLIENT\_TLS = #CERT\_S\_OPERATOR\_TLS |

##### 4.3.1.2.1 TC\_SM-DP+\_ES2+.DownloadOrder

Test Sequence #01 Nominal: EID and ICCID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_ICCID1) |  |
| 2 | S\_LPAd →SM-DP+ | PROC\_ES9+\_VERIFY\_PROFILE\_NOT\_RELEASED\_EMPTY\_MID | |  |

##### 4.3.1.2.2 TC\_SM-DP+\_ES2+.DownloadOrder\_RetryCases

Test Sequence #01 Nominal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_ICCID1) |  |
| 2 | S\_MNO → SM-DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 3 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 4 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_ICCID1) |  |

##### 4.3.1.2.3 TC\_SM-DP+\_ES2+.DownloadOrder\_ErrorCases

Test Sequence #01 Error: ICCID already in use (8.2.1/3.3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| IC2 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  NO\_PARAM,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_ICCID1) |  |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_3\_3) |  |

Test Sequence #02 Error: unknown profile (8.2.1/3.9)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | No Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in the SM-DP+ for S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_3\_9) |  |

### 4.3.2 ES2+ (Operator -- SM-DP+): ConfirmOrder

#### 4.3.2.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

* Section 5.3.2

#### 4.3.2.2 Test Cases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| S\_MNO | For the TLS connection, CERT\_CLIENT\_TLS #CERT\_S\_OPERATOR\_TLS |

##### 4.3.2.2.1 TC\_SM-DP+\_ES2+.ConfirmOrder

Test Sequence #01 Nominal: using ‘Allocated’ state / ICCID / matching ID / confirmation code / releaseFlag=”true”

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Allocated’ state by S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID) with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 2 | S\_LPAd →SM-DP+ | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_WITH\_MID\_WITH\_CC | |  |

Test Sequence #02 Nominal: using ‘Allocated’ state / ICCID / empty matching ID / EID / confirmation code / releaseFlag=”true”

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Allocated’ state by S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, #MATCHING\_ID\_EMPTY, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID with <MATCHING\_ID>= #MATCHING\_ID\_EMPTY) |  |
| 2 | S\_LPAd →SM-DP+ | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_EMPTY\_MID\_WITH\_CC | |  |

Test Sequence #03 Nominal: using ‘Allocated’ state / ICCID / matching ID / EID / smdsAddress / confirmation code / releaseFlag=”true”

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Allocated’ state by S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_MNO →SM-DP+ | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES2+ | |  |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, #TEST\_ROOT\_DS\_ADDRES, TRUE)) |  |  |
| 2 | SM-DP+ → S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | |  |
| 3 | SM-DP+ → S\_SM-DS | Call ES12.RegisterEvent | MTD\_HTTP\_REQ(    #TEST\_ROOT\_DS\_ADDRESS,    #PATH\_REGISTER\_EVENT,    MTD\_REGISTER\_EVENT(       <FUNCTION\_REQ\_ID>,       <FUNCTION\_CALL\_ID>,       #EID1,             #IUT\_SM\_DP\_ADDRESS,        #MATCHING\_ID\_1,       <FORWARDING\_INDICATOR\_ANY>)) |  |
| 4 | S\_SM-DS → SM-DP+ | MTD\_HTTP\_RESP(#R\_SUCCESS) on ES12 | No Error |  |
| 5 | SM-DP+ →S\_MNO | Return final result | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID) on ES2+  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 6 | S\_LPAd →SM-DP+ | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_EMPTY\_MID\_WITH\_CC | |  |

Test Sequence #04 Nominal: using ‘Linked’ state / ICCID / matching ID / confirmation code / releaseFlag=”true”

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state for S\_MNO, and bound to #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID)  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 2 | S\_LPAd →SM-DP+ | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_EMPTY\_MID\_WITH\_CC | |  |

Test Sequence #05 Nominal: using ‘Linked’ state / ICCID / empty matching ID / EID / confirmation code / releaseFlag=”true”

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state for S\_MNO, and bound to #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, #MATCHING\_ID\_EMPTY, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID with <MATCHING\_ID>= #MATCHING\_ID\_EMPTY) |  |
| 2 | S\_LPAd → SM-DP+ | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_EMPTY\_MID\_WITH\_CC | |  |

Test Sequence #06 Nominal: using ‘Linked’ state / ICCID / matching ID / EID / smdsAddress / confirmation code / releaseFlag=”true”

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state for S\_MNO, and bound to #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_MNO→ SM-DP+ | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES2+ | |  |
| 1 | S\_MNO→ SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, #TEST\_ROOT\_DS\_ADDRES, TRUE)) |  |  |
| 2 | SM-DP+→ S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | |  |
| 3 | SM-DP+→ S\_SM-DS | Call ES12.RegisterEvent | MTD\_HTTP\_REQ(    #TEST\_ROOT\_DS\_ADDRESS,    #PATH\_REGISTER\_EVENT,    MTD\_REGISTER\_EVENT(       <FUNCTION\_REQ\_ID>,       <FUNCTION\_CALL\_ID>,       #EID1,             #IUT\_SM\_DP\_ADDRESS,        #MATCHING\_ID\_1,       <FORWARDING\_INDICATOR\_ANY>)) |  |
| 4 | S\_SM-DS→ SM-DP+ | Return ES12.RegisterEvent result | MTD\_HTTP\_RESP(#R\_SUCCESS) on ES12 |  |
| 5 | SM-DP+→ S\_MNO | Return final result | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID) on ES2+  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 6 | S\_LPAd→ SM-DP+ | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_EMPTY\_MID\_WITH\_CC | |  |

##### 4.3.2.2.2 TC\_SM-DP+\_ES2+.ConfirmOrder\_RetryCases

Test Sequence #01 Nominal: using ‘Allocated’ state

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Allocated’ state by S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID)  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 2 | S\_MNO → SM-DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 3 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 4 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID)  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |

Test Sequence #02 Nominal: using ‘Linked’ state

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state by S\_MNO and bound to #EID1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID)  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 2 | S\_MNO → SM-DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 3 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 4 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID)  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |

Test Sequence #03 Error: different matchingID (unspecified Error Code)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state by S\_MNO and bound to #EID1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID)  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 2 | S\_MNO → SM-DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 3 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 4 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_2, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_ANY) |  |

Test Sequence #04 Error: different Confirmation Code (unspecified Error Code)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state by S\_MNO and bound to #EID1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_MATCHING\_ID\_EID)  with <MATCHING\_ID>= #MATCHING\_ID\_1 |  |
| 2 | S\_MNO → SM-DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 3 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 4 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE2, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_ANY) |  |



##### 4.3.2.2.3 TC\_SM-DP+\_ES2+.ConfirmOrder\_ErrorCases

Test Sequence #01 Error: unknown Profile (8.2.1/3.9)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | No Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in the server for S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_3\_9) |  |

Test Sequence #02 Error: Profile in ‘Available’ state (unspecified Error Code)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Available’ state for S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_ANY) |  |

Test Sequence #03 Error: conflicting matching ID (8.2.6/3.3)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Allocated’ state for S\_MNO.  The matchingID #MATCHING\_ID\_1 is already stored for S\_MNO and associated to another Profile identified by ICCID\_OP\_PROF2\_NON\_SWAP. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM, TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_3) |  |

Test Sequence #04 Error: incorrect smdsAddress (8.9/5.1)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state for S\_MNO. |
| S\_SM-DS | The S\_SM-DS is not reachable through #TEST\_DS\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, #TEST\_DS\_ADDRESS1, TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_1) |  |

Test Sequence #05 Error: missing EID (8.1.1/2.2)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Allocated’ state for S\_MNO. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_EMPTY , #CONFIRMATION\_CODE1, NO\_PARAM,  TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_1\_2\_2) |  |

Test Sequence #06 Error: different EID (8.1.1/3.10)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state for S\_MNO, associated to #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CONFIRM\_ORDER,  MTD\_CONFIRM\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID2, #MATCHING\_ID\_1, #CONFIRMATION\_CODE1, NO\_PARAM  TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_1\_3\_10) |  |

### 4.3.3 ES2+ (Operator -- SM-DP+): CancelOrder

#### 4.3.3.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

* Section 5.3.3

#### 4.3.3.2 Test Cases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| S\_MNO | For the TLS connection, CERT\_CLIENT\_TLS = #CERT\_S\_OPERATOR\_TLS |

##### 4.3.3.2.1 TC\_SM-DP+\_ES2+.CancelOrder

Test Sequence #01 Nominal: ‘Linked’ state, using EID, final status = Available

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Linked’ state for S\_MNO and bound to #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, NO\_PARAM, #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_SUCCESS) |  |
| 2 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, NO\_PARAM,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_ICCID1) |  |

Test Sequence #02 Nominal: ‘Confirmed’ state, using EID, final status = Available, SM-DS Use Case

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Confirmed’ state by S\_MNO and bound to #EID1. |
| SM-DP+ | The SM-DP+ has executed a confirm order procedure for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS used for Event Registration, and with MatchingID set to #MATCHING\_ID\_1 in input parameters. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, #MATCHING\_ID\_1, #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_SUCCESS) |  |
| 2 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, NO\_PARAM,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_ICCID1) |  |

Test Sequence #03 Nominal: ‘Error’ state, using MatchingID, final status = Available

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Error’ state for S\_MNO, associated with #MATCHING\_ID\_1 and not bound to any EID. The ‘Error’ state has been entered by ES9+.CancelSession sent with reason = endUserRejection after the ES9+.AuthenticateClient response of a previous Profile Download attempt. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1, #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_SUCCESS) |  |
| 2 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_DOWNLOAD\_ORDER,  MTD\_DOWNLOAD\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, NO\_PARAM,  #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM)) | MTD\_HTTP\_RESP( #R\_SUCCESS\_ICCID1) |  |

##### 4.3.3.2.2 TC\_SM-DP+\_ES2+.CancelOrder\_ErrorCases

Test Sequence #01 Error: unknown ICCID (8.2.1/3.9)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | No Profile identified by #ICCID\_OP\_PROF1\_NON\_SWAP is configured in the server for S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, NO\_PARAM, #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_3\_9) |  |

Test Sequence #02 Error: missing EID (unspecified Error Code)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by #ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Confirmed’ state by S\_MNO and bound to #EID1 |
| SM-DP+ | The SM-DP+ has previously executed a confirm order with MatchingID set to #MATCHING\_ID\_1 in input parameters |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, #MATCHING\_ID\_1 , #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_ERROR\_ANY) |  |

Test Sequence #03 Error: incorrect matchingID (8.2.6/3.10)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by #ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Released’ state by S\_MNO, is bound to #EID1 and is associated with matchingID #MATCHING\_ID\_1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, #MATCHING\_ID\_2, #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_10) |  |

Test Sequence #04 Error: profile in Available state (unspecified Error Code)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by #ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Available’ state for S\_MNO |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, NO\_PARAM, NO\_PARAM, #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_ERROR\_ANY) |  |

Test Sequence #05 Error: profile in Installed state (8.2.1/3.3)

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | The Profile identified by #ICCID\_OP\_PROF1\_NON\_SWAP is configured in ‘Installed’ state by S\_MNO and is bound to #EID1 |
| SM-DP+ | The SM-DP+ has previously executed a confirm order with MatchingID set to #MATCHING\_ID\_1 in input parameters |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | | |
| 1 | S\_MNO →SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS,  #PATH\_CANCEL\_ORDER,  MTD\_CANCEL\_ORDER(  #S\_MNO\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1, #ICCID\_OP\_PROF1\_NON\_SWAP, #EID1, #MATCHING\_ID\_1, #PROFILE\_STATUS\_AVAILABLE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_3\_3) |  |

### 4.3.4 ES2+ (Operator -- SM-DP+): ReleaseProfile

This test case is defined as FFS and not applicable for this version of test specification.

### 4.3.5 ES2+ (Operator -- SM-DP+): HandleDownloadProgressInfo

This test case is defined as FFS and not applicable for this version of test specification.

### 4.3.6 ES2+ (Operator -- SM-DP+): TLS, Mutual Authentication, Server,Session Establishment

This test case is defined as FFS and not applicable for this version of test specification.

### 4.3.7 ES8+ (SM-DP+ -- eUICC): InitialiseSecureChannel

#### 4.3.7.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

#### 4.3.7.2 Test Cases

All testing for ES8+ functions is performed in section 4.3.13 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage.

### 4.3.8 ES8+ (SM-DP+ -- eUICC): ConfigureISDP

#### 4.3.8.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

#### 4.3.8.2 Test Cases

All testing for ES8+ functions is performed in section 4.3.13 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage.

### 4.3.9 ES8+ (SM-DP+ -- eUICC): StoreMetadata

#### 4.3.9.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

#### 4.3.9.2 Test Cases

All testing for ES8+ functions is performed in section 4.3.13 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage.

### 4.3.10 ES8+ (SM-DP+ -- eUICC): ReplaceSessionKeys

#### 4.3.10.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

#### 4.3.10.2 Test Cases

All testing for ES8+ functions is performed in section 4.3.13 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage.

### 4.3.11 ES8+ (SM-DP+ -- eUICC): LoadProfileElements

#### 4.3.11.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

#### 4.3.11.2 Test Cases

All testing for ES8+ functions is performed in section 4.3.13 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage.

### 4.3.12 ES9+ (LPA -- SM-DP+): InitiateAuthentication

The test sequences defined in this section are intended for testing on both the SM‑DP+ and the SM‑DS.

#### 4.3.12.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_033

* RQ31\_030, RQ31\_033, RQ31\_034, RQ31\_035, RQ31\_036, RQ31\_037, RQ31\_038, RQ31\_039, RQ31\_041, RQ31\_042, RQ31\_043, RQ31\_073

 RQ45\_006, RQ45\_026, RQ45\_026\_1

 RQ56\_004, RQ56\_005, RQ56\_006, RQ56\_007, RQ56\_008, RQ56\_009, RQ56\_010, RQ56\_011, RQ56\_012, RQ56\_013, RQ56\_014

 RQ57\_106

 RQ62\_001, RQ62\_002, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_017, RQ65\_018

#### 4.3.12.2 Test Cases

|  |  |
| --- | --- |
| General Initial Conditions for SM-DP + testing | |
| Entity | Description of the general initial condition |
| SM-DP+ | SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST. |

##### 4.3.12.2.1 TC\_SM-DP+\_ES9+.InitiateAuthenticationNIST

Test Sequence #01 Nominal

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK)  • Verify that <TRANSACTION\_ID\_IA> matches <TRANSACTION\_ID\_SIGNED\_IA>  • Verify the validity of <SERVER\_SIGNATURE1> using the public key #PK\_SM\_XXauth\_ECDSA contained in #CERT\_SM\_XXauth\_ECDSA | Common: RQ31\_030 RQ31\_033 RQ31\_034 RQ31\_035 RQ31\_037 RQ31\_038 RQ31\_039 RQ31\_041 RQ31\_042 RQ31\_043 RQ45\_006 RQ45\_026 RQ45\_026 RQ57\_106 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_017 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_007 RQ56\_009 RQ56\_010 RQ56\_012 RQ56\_013 SM‑DS: RQ58\_003 RQ58\_004 RQ58\_005 RQ58\_006 RQ58\_008 RQ58\_010 RQ58\_012 RQ58\_013 RQ58\_014 RQ58\_015 RQ58\_016 RQ58\_017 RQ58\_018 RQ58\_019 |

Test Sequence #02 Nominal: Uniqueness of Transaction ID and Server Challenge

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) | Common: RQ31\_030 RQ31\_033 RQ31\_034 RQ31\_035 RQ31\_037 RQ31\_038 RQ31\_039 RQ31\_041 RQ31\_042 RQ31\_043 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ57\_106 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_017 RQ65\_01 SM‑DP+ RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_007 RQ56\_009 RQ56\_010 RQ56\_012 RQ56\_013 SM‑DS RQ58\_003 RQ58\_004 RQ58\_005 RQ58\_006 RQ58\_008 RQ58\_010 RQ58\_012 RQ58\_013 RQ58\_014 RQ58\_015 RQ58\_016 RQ58\_017 RQ58\_018 RQ58\_019 |
| 2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 3 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK\_2)  Verify that:   <TRANSACTION\_ID\_2> received in this step is different to the <TRANSACTION\_ID\_IA> in Step 1  <TRANSACTION\_ID\_SIGNED\_2> received in this step is different to the <TRANSACTION\_ID\_SIGNED\_IA> in Step 1   <SERVER\_CHALLENGE\_2> received in this step is different to the <SERVER\_CHALLENGE> in Step 1. | Common: RQ31\_030 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_017 RQ65\_018 SM‑DP+: RQ56\_009 SM‑DS: RQ56\_008 |

Test Sequence #03 Error: Failed due to Invalid Server Address (Subject Code 8.8.1 Reason Code 3.8)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ (  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #UNKNOWN\_SERVER\_ADDRESS)) | MTD\_HTTP\_RESP(  #R\_ERROR\_SMXX\_1\_3\_8) | Common RQ31\_033 RQ31\_034 RQ57\_106 RQ62\_001 RQ62\_002 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_005 RQ56\_008 RQ56\_011 RQ56\_014 SM‑DS: RQ58\_003 RQ58\_004 RQ58\_007 RQ58\_011 RQ58\_020 |

Test Sequence #04 Error: Failed due to Unsupported Public Key Identifiers (Subject Code 8.8.2 Reason Code 3.1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #EUICC\_INFO1\_8\_8\_2\_3\_1,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP(  #R\_ERROR\_SMXX\_2\_3\_1) | Common: RQ26\_033 RQ31\_033 RQ31\_034 RQ31\_035 RQ31\_036 RQ57\_106 RQ62\_001 RQ62\_002 RQ65\_018 SM-DP+: RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_008 RQ56\_011 RQ56\_014 SM-DS: RQ58\_003 RQ58\_004 RQ58\_005 RQ58\_007 RQ58\_011 RQ58\_020 |

Test Sequence #05 Error: Failed due to Unsupported Specification Version Number (Subject Code 8.8.3 Reason Code 3.1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #EUICC\_INFO1\_8\_8\_3\_3\_1\_LOWER,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP(  #R\_ERROR\_SMXX\_3\_3\_1) | Common: RQ31\_033 RQ31\_034 RQ57\_106 RQ62\_001 RQ62\_002 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_008 RQ56\_011 RQ56\_014 SM‑DS: RQ58\_003 RQ58\_007 RQ58\_011 RQ58\_020 |
| 2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 3 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #EUICC\_INFO1\_8\_8\_3\_3\_1\_HIGHER,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP(  #R\_ERROR\_SMXX\_3\_3\_1) | Common: RQ31\_033 RQ31\_034 RQ57\_106 RQ62\_001 RQ62\_002 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_008 RQ56\_011 RQ56\_014 SM‑DS: RQ58\_003 RQ58\_007 RQ58\_011 RQ58\_020 |

Test Sequence #06 Error: Failed due to Unavailable Server Auth Certificate (Subject Code 8.8.4 Reason Code 3.7)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #EUICC\_INFO1\_8\_8\_4\_3\_7,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP(  #R\_ERROR\_SMXX\_4\_3\_7) | Common: RQ26\_033 RQ31\_033 RQ31\_034 RQ31\_035 RQ31\_036 RQ57\_106 RQ62\_001 RQ62\_002 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_008 RQ56\_011 RQ56\_014 SM‑DS: RQ58\_003 RQ58\_004 RQ58\_005 RQ58\_006 RQ58\_007 RQ58\_011 RQ58\_020 |

Test Sequence #07 Nominal: eUICC v2.2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1\_V2\_2\_1,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK)  • Verify that <TRANSACTION\_ID\_IA> matches <TRANSACTION\_ID\_SIGNED\_IA>  • Verify the validity of <SERVER\_SIGNATURE1> using the public key #PK\_SM\_XXauth\_ECDSA contained in #CERT\_SM\_XXauth\_ECDSA | Common: RQ31\_030 RQ31\_033 RQ31\_034 RQ31\_035 RQ31\_037 RQ31\_038 RQ31\_039 RQ31\_041 RQ31\_042 RQ31\_043 RQ45\_006 RQ45\_026 RQ45\_026 RQ57\_106 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_017 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_007 RQ56\_009 RQ56\_010 RQ56\_012 RQ56\_013 SM‑DS: RQ58\_003 RQ58\_004 RQ58\_005 RQ58\_006 RQ58\_008 RQ58\_010 RQ58\_012 RQ58\_013 RQ58\_014 RQ58\_015 RQ58\_016 RQ58\_017 RQ58\_018 RQ58\_019 |

Test Sequence #08 Nominal: eUICC v2.2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1\_V2\_2\_2,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK)  • Verify that <TRANSACTION\_ID\_IA> matches <TRANSACTION\_ID\_SIGNED\_IA>  • Verify the validity of <SERVER\_SIGNATURE1> using the public key #PK\_SM\_XXauth\_ECDSA contained in #CERT\_SM\_XXauth\_ECDSA | Common: RQ31\_030 RQ31\_033 RQ31\_034 RQ31\_035 RQ31\_037 RQ31\_038 RQ31\_039 RQ31\_041 RQ31\_042 RQ31\_043 RQ45\_006 RQ45\_026 RQ45\_026 RQ57\_106 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_017 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_007 RQ56\_009 RQ56\_010 RQ56\_012 RQ56\_013 SM‑DS: RQ58\_003 RQ58\_004 RQ58\_005 RQ58\_006 RQ58\_008 RQ58\_010 RQ58\_012 RQ58\_013 RQ58\_014 RQ58\_015 RQ58\_016 RQ58\_017 RQ58\_018 RQ58\_019 |

Test Sequence #09 Nominal: eUICC v2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SERVER | MTD\_HTTP\_REQ(  #SERVER\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1\_V2\_3,  #SERVER\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK)  • Verify that <TRANSACTION\_ID\_IA> matches <TRANSACTION\_ID\_SIGNED\_IA>  • Verify the validity of <SERVER\_SIGNATURE1> using the public key #PK\_SM\_XXauth\_ECDSA contained in #CERT\_SM\_XXauth\_ECDSA | Common: RQ31\_030 RQ31\_033 RQ31\_034 RQ31\_035 RQ31\_037 RQ31\_038 RQ31\_039 RQ31\_041 RQ31\_042 RQ31\_043 RQ45\_006 RQ45\_026 RQ45\_026 RQ57\_106 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_017 RQ65\_018 SM‑DP+: RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_007 RQ56\_009 RQ56\_010 RQ56\_012 RQ56\_013 SM‑DS: RQ58\_003 RQ58\_004 RQ58\_005 RQ58\_006 RQ58\_008 RQ58\_010 RQ58\_012 RQ58\_013 RQ58\_014 RQ58\_015 RQ58\_016 RQ58\_017 RQ58\_018 RQ58\_019 |

##### 4.3.12.2.2 TC\_SM-DP+\_ES9+.InitiateAuthenticationFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.3.12.2.3 TC\_SM-DP+\_ES9+.InitiateAuthenticationBRP

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for BRP. |

Test Sequence #01 Nominal

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.3.12.2.1 TC\_SM-DP+\_ES9+.InitiateAuthenticationNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

### 4.3.13 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage

#### 4.3.13.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_028

 RQ25\_001, RQ25\_002, RQ25\_004, RQ25\_005, RQ25\_006, RQ25\_009, RQ25\_010, RQ25\_011, RQ25\_012, RQ25\_013, RQ25\_014, RQ25\_015

 RQ26\_018, RQ26\_019, RQ26\_020, RQ26\_021, RQ26\_022, RQ26\_029, RQ26\_031, RQ26\_034, RQ26\_035

 RQ31\_143, RQ31\_144, RQ31\_146, RQ31\_147, RQ31\_148, RQ31\_148\_2, RQ31\_148\_3, RQ31\_149, RQ31\_150, RQ31\_151, RQ31\_152, RQ31\_155, RQ31\_162, RQ31\_165, RQ31\_166, RQ31\_168, RQ31\_170

 RQ32\_069, RQ32\_070

 RQ44\_001

 RQ47\_001

 RQ55\_001, RQ55\_002, RQ55\_003, RQ55\_004, RQ55\_005, RQ55\_006, RQ55\_007, RQ55\_008, RQ55\_009, RQ55\_017, RQ55\_018, RQ55\_020, RQ55\_021, RQ55\_022, RQ55\_028, RQ55\_033, RQ55\_033\_1, RQ55\_037, RQ55\_040, RQ55\_041

 RQ56\_015, RQ56\_016, RQ56\_017, RQ56\_018, RQ56\_019, RQ56\_020, RQ56\_021, RQ56\_022, RQ56\_023, RQ56\_024, RQ56\_025, RQ56\_026, RQ56\_027, RQ56\_028

 RQ57\_028, RQ57\_039

 RQ62\_001, RQ62\_002, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_020, RQ65\_021

 RQG0\_001, RQG0\_002, RQG0\_003, RQG0\_004, RQG0\_005, RQG0\_006

#### 4.3.13.2 Test Cases

##### 4.3.13.2.1 TC\_SM-DP+\_ES9+.GetBoundProfilePackageNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * There have been no previous attempts to download the pending profile. |

Test Sequence #01 Nominal: Using S-ENC and S-MAC without Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is loaded as an Unprotected Profile Package. * Confirmation Code is not provided by the Operator to the SM-DP+.Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_SK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1) | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #02 Nominal: Using S-ENC and S-MAC with Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is loaded as an Unprotected Profile Package. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_SK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1) | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001, RQG0\_002, RQG0\_003, RQG0\_004, RQG0\_005, RQG0\_006 |

Test Sequence #03 Nominal: Using PPK-ENC and PPK-MAC without Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>,  #SMDP\_METADATA\_OP\_PROF1) | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #04 Nominal: Using PPK-ENC and PPK-MAC with Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM‑DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1) | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #05 Nominal: Using S-ENC and S-MAC with Metadata split over 2 segments without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package using the S-ENC and S-MAC with the metadata split over two sequenceOf88 segments without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1\_2\_SEG is loaded as an Unprotected Profile Package. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #01 defined in this section except that #SMDP\_METADATA\_OP\_PROF1\_2\_SEG replaces #SMDP\_METADATA\_OP\_PROF1.

NOTE: There is no testing required in addition to Test Sequence #01 as the R\_GET\_BPP\_RESP\_OP1\_SK constants allow for 1 or 2 segments and for the SM-DP+ to successfully pass this test sequence it SHALL use 2 segments to deliver the metadata.

Test Sequence #06 Nominal: Using PPK-ENC and PPK-MAC with Metadata split over 2 segments without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package using the PPK-ENC and PPK-MAC with the metadata split over two sequenceOf88 segments without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1\_2\_SEGis securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #03 defined in this section except that #SMDP\_METADATA\_OP\_PROF1\_2\_SEG replaces #SMDP\_METADATA\_OP\_PROF1.

NOTE: There is no testing required in addition to Test Sequence #03 as the R\_GET\_BPP\_RESP\_OP1\_PPK constants allow for 1 or 2 segments and for the SM-DP+ to successfully pass this test sequence it SHALL use 2 segments to deliver the metadata.

##### 4.3.13.2.2 TC\_SM-DP+\_ES9+.GetBoundProfilePackageFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.3.13.2.3 TC\_SM-DP+\_ES9+.GetBoundProfilePackageBRP

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for BRP. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * There have been no previous attempts to download the pending profile. |

Test Sequence #01 Nominal: Using S-ENC and S-MAC without Confirmation Code

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.3.13.2.1 TC\_SM-DP+\_ES9+.GetBoundProfilePackageNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #02 Nominal: Using PPK-ENC and PPK-MAC without Confirmation Code

This test sequence SHALL be the same as the Test Sequence #03 defined in section 4.3.13.2.1 TC\_SM-DP+\_ES9+.GetBoundProfilePackageNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

##### 4.3.13.2.4 TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_ReuseOTPK\_NIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

Test Sequence #01 Nominal: Retry with same otPK.eUICC.ECKA using S-ENC and S-MAC without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt for the same otPK.eUICC.ECKA using S-ENC and S-MAC for Profile protection without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is loaded as an Unprotected Profile Package. * Confirmation Code is not provided by the Operator to the SM-DP+. * There have been no previous attempts to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_SK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_SK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that  <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 matches the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_3 RQ31\_149 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_021 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #02 Nominal: Retry with same otPK.eUICC.ECKA using S-ENC and S-MAC with Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt for the same otPK.eUICC.ECKA using the S-ENC and S-MAC for Profile protection with a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is loaded as an Unprotected Profile Package. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. * There have been no previous attempts to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_SK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_SK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that  <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 matches the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_3 RQ31\_149 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_021 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #03 Nominal: Retry with same otPK.eUICC.ECKA using PPK-ENC and PPK-MAC without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt for the same otPK.eUICC.ECKA using the PPK-ENC and PPK-MAC for Profile protection without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. * There has been no previous attempts to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_PPK Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 matches the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_148\_3 RQ31\_149 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_021 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #04 Nominal: Retry with same otPK.eUICC.ECKA using PPK-ENC and PPK-MAC with Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package with a retry attempt for the same otPK.eUICC.ECKA using the PPK-ENC and PPK-MAC for Profile protection with a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. * There has been no previous attempts to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_PPK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 matches the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_3 RQ31\_149 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_021 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #05 Nominal: Retry with same otPK.EUICC.ECKA rejected by eUICC using S-ENC and S-MAC without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt with the same otPK.EUICC.ECKA rejected by the eUICC using the S-ENC and S-MAC without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is loaded as an Unprotected Profile Package. * Confirmation Code is not provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_SK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response inStep 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_3\_7)  OR  MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_SK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_148\_3 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_022 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #06 Nominal: Retry with same otPK.EUICC.ECKA rejected by eUICC using S-ENC and S-MAC with Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt with the same otPK.EUICC.ECKA rejected by the eUICC using the S-ENC and S-MAC with a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is loaded as an Unprotected Profile Package. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_SK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1in the GetBoundProfilePackage Response inStep 4 | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK\_CC)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_3\_7)  OR  MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_SK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_3 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_022 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #07 Nominal: Retry with same otPK.EUICC.ECKA rejected by eUICC using PPK-ENC and PPK-MAC without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt with the same otPK.EUICC.ECKA rejected by the eUICC using the PPK-ENC and PPK-MAC without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_PPK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_3\_7)  OR  MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_148\_3 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_022 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #08 Nominal: Retry with same otPK.EUICC.ECKA rejected by eUICC using PPK-ENC and PPK-MAC with Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt with the same otPK.EUICC.ECKA rejected by the eUICC using the PPK-ENC and PPK-MAC.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_PPK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC\_RETRY | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK\_CC)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_3\_7)  OR  MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_3 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_022 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ62\_001 RQ57\_039 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #09 Nominal: Confirmation Code retry

The purpose of this test is to test that the SM-DP+ accepts a subsequent correct Confirmation Code after the initial Confirmation Code supplied in the GetBoundProfilePackageRequest ASN.1 euiccSigned2 element is unknown.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and PPK\_MAC>. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. * The SM-DP+ is configured with two retries allowed for the receipt of a valid Confirmation Code. * There have been no previous attempts to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_INVALID\_CC | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1) | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148 RQ31\_148\_3 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_020 RQ56\_025 RQ56\_026 RQ56\_028 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

##### 4.3.13.2.5 VOID

##### 4.3.13.2.6 VOID

##### 4.3.13.2.7 TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_DifferentOTPK\_NIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

Test Sequence #01 Nominal: Retry without otPK.EUICC.ECKA using S-ENC and S-MAC without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt without otPK.EUICC.ECKA using the S-ENC and S-MAC without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is loaded as an Unprotected Profile Package. * Confirmation Code is not provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_SK  Extract <OTPK\_SM\_DP+\_ECKA> from the GetBoundProfilePackage Response inStep 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_SK)   • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #02 Nominal: Retry without otPK.EUICC.ECKA using S-ENC and S-MAC with Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt without otPK.EUICC.ECKA using the S-ENC and S-MAC with a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is loaded as an Unprotected Profile Package. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_SK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK\_CC)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_SK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_SK (#R\_GET\_BPP\_RESP\_OP1\_SK, <S\_MAC>, <S\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_004 RQ25\_006 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_022 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #03 Nominal: Retry without otPK.EUICC.ECKA using PPK-ENC and PPK-MAC without Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt without otPK.EUICC.ECKA using the PPK-ENC and PPK-MAC without a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_PPK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1 | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_148\_3 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_022 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

Test Sequence #04 Nominal: Retry without otPK.EUICC.ECKA using PPK-ENC and PPK-MAC with Confirmation Code

The purpose of this test is to test that the LPAd can request the delivery and the binding of a Profile Package for a retry attempt without otPK.EUICC.ECKA using the PPK-ENC and PPK-MAC with a Confirmation Code.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. * There have been no previous attempt to download the pending profile. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_PPK  Extract <OTPK\_SM\_DP+\_ECKA> from #INIT\_SC\_PROF1 in the GetBoundProfilePackage Response in Step 4. | | |  |
| IC2 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_NEW\_OTPK\_CC)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK)  • Verify that <TRANSACTION\_ID\_GBPP> matches <S\_TRANSACTION\_ID>  MTD\_TEST\_ES8+\_GET\_BPP\_PPK (#R\_GET\_BPP\_RESP\_OP1\_PPK, <S\_MAC>, <S\_ENC>, <PPK\_MAC>, <PPK\_ENC>, #SMDP\_METADATA\_OP\_PROF1)  • Verify that <OTPK\_SM\_DP+\_ECKA> in #INIT\_SC\_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1. | RQ25\_001 RQ25\_002 RQ25\_005 RQ25\_006 RQ25\_009 RQ25\_010 RQ25\_011 RQ25\_012 RQ25\_013 RQ25\_014 RQ25\_015 RQ26\_018 RQ26\_019 RQ26\_020 RQ26\_021 RQ26\_022 RQ26\_029 RQ26\_031 RQ26\_034 RQ26\_035 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_3 RQ31\_150 RQ31\_151 RQ31\_152 RQ31\_155 RQ31\_162 RQ31\_165 RQ31\_166 RQ31\_168 RQ31\_170 RQ32\_069 RQ32\_070 RQ44\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ47\_001 RQ55\_001 RQ55\_002 RQ55\_003 RQ55\_004 RQ55\_005 RQ55\_006 RQ55\_007 RQ55\_008 RQ55\_009 RQ55\_017 RQ55\_018 RQ55\_020 RQ55\_021 RQ55\_022 RQ55\_028 RQ55\_033 RQ55\_033\_1 RQ55\_037 RQ55\_040 RQ55\_041 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_022 RQ56\_023 RQ56\_024 RQ56\_026 RQ56\_027 RQ57\_039 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_020 RQ65\_021 RQG0\_001 RQG0\_002 RQG0\_003 RQG0\_004 RQG0\_005 RQG0\_006 |

##### 4.3.13.2.8 VOID

##### 4.3.13.2.9 VOID

##### 4.3.13.2.10 TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_ErrorCasesNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST. * PROFILE\_OPERATIONAL1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1.   There have been no previous attempts to download the pending profile. |

Test Sequence #01 Error: Invalid eUICC Signature (Subject Code 8.1 Reason Code 6.1)

The purpose of this test is to test that the SM-DP+ returns the correct error code for an invalid eUICC signature supplied in GetBoundProfilePackageRequest.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_8\_1\_6\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | RQ26\_029 RQ26\_031 RQ31\_143 RQ31\_148\_2 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_018 RQ56\_025 RQ56\_026 RQ56\_028 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 |

Test Sequence #02 Error: Unknown TransactionID in JSON transport layer (Subject Code 8.10.1 Reason Code 3.9)

The purpose of this test is to test that the SM-DP+ returns the correct error code when the TransactionID supplied in GetBoundProfilePackageRequest JSON transport layer is unknown.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <INVALID\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ26\_029 RQ26\_031 RQ31\_143 RQ31\_148\_2 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_018 RQ56\_025 RQ56\_026 RQ56\_028 RQ62\_001 RQ62\_002 |

Test Sequence #03 Error: Unknown TransactionID in ASN.1 euiccSigned2 element (Subject Code 8.10.1 Reason Code 3.9)

The purpose of this test is to test that the SM DP+ returns the correct error code when the TransactionID supplied in the GetBoundProfilePackageRequest ASN.1 euiccSigned2 element is unknown.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM‑DP+ | Confirmation Code is not provided by the Operator to the SM‑DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_8\_10\_1\_3\_9)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ26\_029 RQ26\_031 RQ31\_143 RQ31\_148\_2 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_018 RQ56\_025 RQ56\_026 RQ56\_028 RQ62\_001 RQ62\_002 |

Test Sequence #04 Error: Missing Confirmation Code (Subject Code 8.2.7 Reason Code 2.2)

The purpose of this test is to test that the SM-DP+ returns the correct error code when the Confirmation Code is missing in the PrepareDownloadResponse request ASN.1 euiccSigned2 element.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | Confirmation Code #CONFIRMATION\_CODE1 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_7\_2\_2) | RQ26\_029 RQ26\_031 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_2 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_018 RQ56\_025 RQ56\_026 RQ56\_028 RQ62\_001 RQ62\_002 |

Test Sequence #05 Error: Refused Confirmation Code (Subject Code 8.2.7 Reason Code 3.8)

The purpose of this test is to test that the SM-DP+ returns the correct error code when the Confirmation Code supplied in the GetBoundProfilePackageRequest ASN.1 euiccSigned2 element is unknown.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | Confirmation Code #CONFIRMATION\_CODE2 associated to PROFILE\_OPERATIONAL1 is provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| IC2 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | | |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_7\_3\_8) | RQ26\_029 RQ26\_031 RQ31\_143 RQ31\_144 RQ31\_146 RQ31\_147 RQ31\_148\_2 RQ56\_015 RQ56\_016 RQ56\_017 RQ56\_018 RQ56\_025 RQ56\_026 RQ56\_028 RQ62\_001 RQ62\_002 |

Test Sequence #06 VOID

##### 4.3.13.2.11 VOID

##### 4.3.13.2.12 VOID

### 4.3.14 ES9+ (LPA -- SM-DP+): AuthenticateClient

#### 4.3.14.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_033

* RQ31\_025, RQ31\_058RQ31\_058, RQ31\_059, RQ31\_060, RQ31\_061, RQ31\_067, RQ31\_080, RQ31\_081, RQ31\_082, RQ31\_083, RQ31\_085, RQ31\_086, RQ31\_089, RQ31\_090, RQ31\_091, RQ31\_092, RQ31\_093, RQ31\_094, RQ31\_095

 RQ41\_001, RQ41\_006, RQ41\_007, RQ41\_008

 RQ42\_001

 RQ45\_006, RQ45\_017, RQ45\_026, RQ45\_026\_1, RQ45\_027, RQ45\_028, RQ45\_029

 RQ47\_001

 RQ56\_029, RQ56\_030, RQ56\_031, RQ56\_032, RQ56\_033, RQ56\_034, RQ56\_035, RQ56\_036, RQ56\_036\_1, RQ56\_037, RQ56\_038, RQ56\_039, RQ56\_040, RQ56\_041, RQ56\_041\_1, RQ56\_041\_2

 RQ57\_037, RQ57\_057, RQ57\_057\_1, RQ57\_108

 RQ62\_001, RQ62\_002, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_007, RQ65\_008 RQ65\_009, RQ65\_022, RQ65\_023

#### 4.3.14.2 Test Cases

##### 4.3.14.2.1 TC\_SM-DP+\_ES9+.AuthenticateClientNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST and #CERT\_SM\_DPpb\_ECDSA for NIST. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. |

Test Sequence #01 Nominal for Default SM-DP+ Address Use Case without Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPpb\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #02 Nominal for Default SM-DP+ Address Use Case with Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code #CONFIRMATION\_CODE1 is provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE ,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_CC)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPpb\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ47\_001 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #03 Nominal for Default SM-DP+ Use Case Second Attempt without Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_AUTH\_CLIENT\_FAIL\_DEF\_DP\_USE\_CASE\_INVALID\_MATCHING\_ID | | | |
| IC2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC3 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPpb\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #04 VOID

Test Sequence #05 Nominal for SM-DS Use Case without Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 in the 'Released' state with a MatchingID equal to <MATCHING\_ID\_EVENT>. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_UC\_OK)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPpb\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ41\_007 RQ41\_008 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #06 Nominal for SM-DS Use Case with Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 in the 'Released' state with a MatchingID equal to <MATCHING\_ID\_EVENT>. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code #CONFIRMATION\_CODE1 is provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_CC)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPpb\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ41\_007 RQ41\_008 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ47\_001 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #07 VOID

Test Sequence #08 Nominal for Activation Code Use Case with Matching ID without Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPpb\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_001 RQ41\_006 RQ41\_007 RQ41\_008 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #09 Nominal for Activation Code Use Case with Matching ID with Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code #CONFIRMATION\_CODE1 is provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_CC)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPpb\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_001 RQ41\_006 RQ41\_007 RQ41\_008 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ47\_001 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #10 VOID

Test Sequence #11 Nominal for Activation Code Use Case with Matching ID without Confirmation Code not associated to EID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is not known to the SM-DP+ and is not associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #08 defined in this section.

Test Sequence #12 Nominal for Activation Code Use Case with Matching ID and Confirmation Code not associated to EID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is not known to the SM-DP+ and is not associated to PROFILE\_OPERATIONAL1. * Confirmation Code #CONFIRMATION\_CODE1 is provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #9 defined in this section.

Test Sequence #13 VOID

Test Sequence #14 Nominal for Default SM-DP+ Address Use Case with MatchingId omitted

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and is associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #01 defined in this section except that #AUTH\_SERVER\_RESP\_SMDP\_MATCHING\_ID\_OMITTED shall be used in MTD\_AUTHENTICATE\_CLIENT instead of #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK.

Test Sequence #15 Nominal for SM-DS Use Case with MatchingId omitted

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 in the 'Released' state with a MatchingID equal to <MATCHING\_ID\_EVENT>. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #05 defined in this section except that #AUTH\_SERVER\_RESP\_SMDP\_MATCHING\_ID\_OMITTED shall be used in MTD\_AUTHENTICATE\_CLIENT instead of #AUTH\_SERVER\_RESP\_SMDS\_UC\_OK.

Test Sequence #16 Nominal for SM-DS Use Case with empty MatchingId

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 in the 'Released' state with a MatchingID equal to <MATCHING\_ID\_EVENT>. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #05 defined in this section except that #AUTH\_SERVER\_RESP\_SMDP\_MATCHING\_ID\_EMPTY shall be used in MTD\_AUTHENTICATE\_CLIENT instead of #AUTH\_SERVER\_RESP\_SMDS\_UC\_OK.

Test Sequence #17 Nominal for Activation Code Use Case with MatchingId omitted

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #08 defined in this section except that #AUTH\_SERVER\_RESP\_SMDP\_MATCHING\_ID\_OMITTED shall be used in MTD\_AUTHENTICATE\_CLIENT instead of #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK.

Test Sequence #18 Nominal for Activation Code Use Case with empty MatchingId

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #08 defined in this section except that #AUTH\_SERVER\_RESP\_SMDP\_MATCHING\_ID\_EMPTY shall be used in MTD\_AUTHENTICATE\_CLIENT instead of #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK.

Test Sequence #19 Nominal with extended UICC Capability in eUICCInfo2

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_OK\_UICC\_EXT)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #20 Nominal with extended DeviceInfo

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_OK\_DEVICE\_EXT)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #21 Nominal with extended eUICCInfo2

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_OK\_eUICC\_EXT)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_025 RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

##### 4.3.14.2.2 TC\_SM-DP+\_ES9+.AuthenticateClientNIST\_ErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST and #CERT\_SM\_DPpb\_ECDSA for NIST. * Confirmation Code is not provided by the Operator to the SM-DP+ for the pending profile. |

Test Sequence #1 Error: Invalid EUM Certificate (Subject Code 8.1.2 Reason Code 6.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(   #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_SIG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(   #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 5 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_KU)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 6 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 7 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 8 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(   #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 9 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_CP)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ31\_061 RQ45\_028RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 10 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 11 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 12 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(   #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 13 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_BC\_cA)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 14 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 15 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 16 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(   #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 17 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_BC\_PLC)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #2 Error: Expired EUM Certificate (Subject Code 8.1.2 Reason Code 6.3)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial state |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the ‘Released’ state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_3)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_3) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #3 Error: Invalid eUICC Certificate (Subject Code 8.1.3 Reason Code 6.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_SIG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 5 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_EX\_KU)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 6 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 7 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 8 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 9 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_EX\_CP)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 10 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 11 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 12 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 13 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_SUB\_ORG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |
| 14 | S\_LPAd → SM‑DP+ | Close TLS session (unless SM-DP+ has already closed TLS session) |  |  |
| 15 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 16 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 17 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_SUB\_SN)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #4 Error: Expired eUICC Certificate (Subject Code 8.1.3 Reason Code 6.3)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_3)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_3) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #5 Error: Invalid eUICC Signature (Subject Code 8.1 Reason Code 6.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_6\_1\_SIG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | RQ31\_061 RQ45\_028 RQ56\_030 RQ56\_038 RQ56\_004 RQ62\_001 RQ62\_0021 |

Test Sequence #6 Error: Invalid Server Challenge (Subject Code 8.1 Reason Code 6.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_6\_1\_CHA)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | RQ31\_061 RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #7 Error: Unknown CI Public Key (Subject Code 8.11.1 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_11\_1\_3\_9)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_11\_1\_3\_9) | RQ26\_033 RQ31\_061 RQ45\_028RQ56\_030 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #8 Error: Profile not released (Subject Code 8.2 Reason Code 1.2)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is not in the ‘Released’ state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 iconfigured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_2) | RQ31\_061 RQ31\_083 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #9 Error: Unknown Transaction ID in JSON transport layer (Subject Code 8.10.1 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <INVALID\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ31\_061 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #10 Error: Unknown Transaction ID in ASN.1 euiccSigned1 payload (Subject Code 8.10.1 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_10\_1\_3\_9)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ31\_061 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #11 Error: Invalid Matching ID for Profile Download Default DP+ Address Use Case (Subject Code 8.2.6 Reason Code 3.8)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_8) | RQ31\_061 RQ41\_006 RQ41\_008 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #12 Error: Invalid Matching ID for Profile Download Activation Code Use Case (Subject Code 8.2.6 Reason Code 3.8)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_1. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,   #AUTH\_SERVER\_RESP\_ACT\_CODE\_2\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_8) | RQ31\_061 RQ41\_006 RQ41\_007 RQ41\_008 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #13 Error: Invalid Matching ID for Profile Download SM-DS Use Case (Subject Code 8.2.6 Reason Code 3.8)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 in the 'Released' state with a MatchingID equal to <MATCHING\_ID\_EVENT>. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_8) | RQ31\_061 RQ41\_006 RQ41\_007 RQ41\_008 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #14 Error: Un-matched EID for Default SM-DP+ Address Use Case (Subject Code 8.1.1 Reason Code 3.8)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * EID #EID2 is not known to the SM-DP+ and is not associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_1\_3\_8)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_1\_3\_8) | RQ31\_061 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #15 Error: No Eligible Profile (Subject Code 8.2.5 Reason Code 4.3)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL3 configured with #SMDP\_METADATA\_OP\_PROF3 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Pending Profile PROFILE\_OPERATIONAL3 is in the ‘Released’ state, with an empty MatchingID. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL3. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_2\_5\_4\_3)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_5\_4\_3) | RQ31\_061 RQ31\_086 RQ31\_090 RQ42\_001 RQ56\_033 RQ56\_038 RQ56\_041 RQ57\_057 RQ62\_001 RQ62\_002 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_089 |

Test Sequence #16 Error: Download Order Expired (Subject Code 8.8.5 Reason Code 4.10)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * The SM-DP+ has expired Profile download order.   NOTE: This is expected to be done through proprietary means. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_5\_4\_10) | RQ31\_061 RQ56\_031 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #17 Error: Maximum number of retries for Profile download order exceeded (Subject Code 8.8.5 Reason Code 6.4)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC TC\_SM-DP+\_ES9+.AuthenticateClientBRP. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * All previous attempts to download the pending Profile have been unsuccessful. * The SM-DP+'s maximum number of attempts as defined in #IUT\_SM-DP+\_MAX\_NUMBER\_DOWNLOAD\_ATTEMPTS for the Profile download order has been reached. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_5\_6\_4) | RQ31\_061 RQ31\_067 RQ31\_085 RQ56\_031\_1 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

Test Sequence #18 VOID

Test Sequence #19 Un-matched EID for SM-DS Use Case (Subject Code 8.1.1 Reason Code 3.8)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 in the 'Released' state with a MatchingID equal to <MATCHING\_ID\_EVENT>. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_UC\_OK\_EID2)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_1\_3\_8) | RQ31\_061 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 |

Test Sequence #20 Un-matched EID for Activation Code Use Case (Subject Code 8.1.1 Reason Code 3.8)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_1. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,   #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK\_EID2)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_1\_3\_8) | RQ31\_061 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 |

Test Sequence #21 Invalid MatchingId for Activation Code Use Case not associated to EID (Subject Code 8.2.6 Reason Code 3.8)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_1. * Pending Profile PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. * EID #EID1 is not known to the SM-DP+ and is not associated to PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 2 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,   #AUTH\_SERVER\_RESP\_ACT\_CODE\_2\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_8) | RQ31\_061 RQ41\_006 RQ41\_007 RQ41\_008 RQ56\_033 RQ56\_038 RQ56\_041 RQ62\_001 RQ62\_002 |

##### 4.3.14.2.3 TC\_SM-DP+\_ES9+.AuthenticateClientFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.3.14.2.4 VOID

##### 4.3.14.2.5 TC\_SM-DP+\_ES9+.AuthenticateClientBRP

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for BRP. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * There have been no previous attempts to download the pending profile. |

Test Sequence #01 Nominal for Default SM-DP+ Address Use Case without Confirmation Code

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.3.14.2.1 TC\_SM-DP+\_ES9+.AuthenticateClientNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

##### 4.3.14.2.6 TC\_SM-DP+\_ES9+.AuthenticateClient\_RetryCases\_Reuse\_OTPK

Test Sequence #01 Nominal Default SM-DP+ Use Case Retry Attempt without Confirmation Code for reuse of OTPK

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with #MATCHING\_ID\_EMPTY. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_PPK | | | |
| IC2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC3 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_RETRY\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPauth\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #02 Nominal SM-DS Use Case Retry Attempt without Confirmation Code for reuse of OTPK

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Pending Profile PROFILE\_OPERATIONAL1 in the 'Released' state with a MatchingID equal to <MATCHING\_ID\_EVENT>. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_SM\_DS\_USE\_CASE\_CANCEL\_SESSION | | | |
| IC2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC3 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_RETRY\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPauth\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_006 RQ41\_007 RQ41\_008 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #03 Nominal Activation Code Use Case with Matching ID Retry Attempt without Confirmation Code for reuse of OTPK

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROF\_DOWNLOAD\_ACT\_CODE\_USE\_CASE\_CANCEL\_SESSION | | | |
| IC2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC3 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_RETRY\_OK)  • Verify that <TRANSACTION\_ID\_AC> matches <S\_TRANSACTION\_ID>  • Verify the validity of the smdpSignature2 <SMDP\_SIGNATURE2> using the #PK\_SM\_DPauth\_ECDSA  • Verify that <TRANSACTION\_ID\_SIGNED\_AC> matches <S\_TRANSACTION\_ID> | RQ31\_058 RQ31\_059 RQ31\_060 RQ31\_080 RQ31\_081 RQ31\_082 RQ31\_091 RQ31\_092 RQ31\_093 RQ31\_094 RQ31\_095 RQ41\_001 RQ41\_006 RQ41\_007 RQ41\_008 RQ42\_001 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_029 RQ56\_029 RQ56\_032 RQ56\_034 RQ56\_035 RQ56\_036 RQ56\_036\_1 RQ56\_037 RQ56\_039 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 RQ57\_037 RQ57\_057\_1 RQ57\_108 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_023 |

Test Sequence #04 Nominal Activation Code Use Case with Matching ID for Retry Attempt without Confirmation Code not associated to EID for reuse of OTPK

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING\_ID\_1. * EID #EID1 is not known to the SM-DP+ and not associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

This test sequence SHALL be the same as the Test Sequence #03 defined in this section.

### 4.3.15 ES9+ (LPA -- SM-DP+): HandleNotification

#### 4.3.15.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ25\_016, RQ25\_018, RQ25\_023

 RQ25\_024, RQ25\_025, RQ25\_026

 RQ31\_171, RQ31\_176, RQ31\_177, RQ31\_177\_1, RQ31\_178, RQ31\_181

 RQ35\_017, RQ35\_019, RQ35\_022

 RQ45\_006, RQ45\_026, RQ45\_026\_1

 RQ55\_048\_1

 RQ56\_042, RQ56\_042\_1, RQ56\_042\_2

 RQ57\_075

 RQ62\_001, RQ62\_002, RQ62\_003, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007, RQ62\_009

 RQ63\_005

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_006, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_024

#### 4.3.15.2 Test Cases

##### 4.3.15.2.1 TC\_SM-DP+\_ES9+\_HandleNotificationNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| **Entity** | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * There have been no previous attempts to download pending Profile PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

Test Sequence #01 Nominal: All Notifications

The purpose of this test is to verify that the SM-DP+ acknowledges the incoming ProfileInstallationResult and OtherSignedNotification for all types of Profile notifications.

|  |  |
| --- | --- |
| **Initial Conditions** | |
| **Entity** | **Description of the initial condition** |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_ALL\_NOTIF is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC using #R\_AUTH\_CLIENT\_OK\_ALL\_NOTIF instead of #R\_AUTH\_CLIENT\_OK | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_OK1)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ31\_181 RQ35\_017 RQ35\_019 RQ35\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048\_1RQ56\_042 RQ56\_042\_1RQ56\_042\_2 RQ57\_075 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_005 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_024 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PENDING\_NOTIF\_OTHER\_INST1)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ31\_181 RQ35\_017 RQ35\_019 RQ35\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048\_1RQ56\_042 RQ56\_042\_1RQ56\_042\_2RQ57\_075 RQ62\_001  RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_005 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_024 |
| 5 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 6 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PENDING\_NOTIF\_EN1)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ31\_181 RQ35\_017 RQ35\_019 RQ35\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048\_1 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ57\_075 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_005 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_024 |
| 7 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 8 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PENDING\_NOTIF\_DIS1)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ31\_181 RQ35\_017 RQ35\_019 RQ35\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048\_1 RQ56\_042 RQ56\_042\_1RQ56\_042\_2 RQ57\_075 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_005 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_024 |
| 9 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 10 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PENDING\_NOTIF\_DE1)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ31\_181 RQ35\_017 RQ35\_019 RQ35\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048\_1 RQ56\_042 RQ56\_042\_1RQ56\_042\_2 RQ57\_075 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_005 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_024 |

Test Sequence #02 Nominal: Successful PIR, no install OtherSignedNotification and then Enable OtherSignedNotification Notifications

The purpose of this test is to verify that the SM-DP+ acknowledges the incoming ProfileInstallationResult and OtherSignedNotification for Profile enable.

|  |  |
| --- | --- |
| **Initial Conditions** | |
| **Entity** | **Description of the initial condition** |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1\_EN is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC\_EN | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_OK1)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ31\_181 RQ35\_017 RQ35\_019 RQ35\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048\_1 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ57\_075 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_005 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_024 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PENDING\_NOTIF\_EN1)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ31\_181 RQ35\_017 RQ35\_019 RQ35\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048\_1RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ57\_075 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_005 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_024 |

Test Sequence #03 Error: Invalid Transaction ID

|  |  |
| --- | --- |
| **Initial Conditions** | |
| **Entity** | **Description of the initial condition** |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_INVALID\_TRANS\_ID)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ31\_178 |

Test Sequence #04 Error: PIR Error Reason - incorrect Input Values

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_INCORRECT\_INPUT\_VALUES)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #05 Error: PIR Error Reason – invalid signature

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_INVALID\_SIGN)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #06 Error: PIR Error Reason – unsupported Crt Values

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>.. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_UNSUPPORTED\_CRT)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #07 Error: PIR Error Reason – unsupported Remote Operation Type

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_UNSUP\_REMOTE\_OP\_TYPE)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #08 Error: PIR Error Reason – unsupported Profile Class

|  |  |
| --- | --- |
| Initial Conditions | |
| **Entity** | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_UNSUP\_PROFILE\_CLASS)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #09 Error: PIR Error Reason – SCP03t Structure Error

|  |  |
| --- | --- |
| Initial Conditions | |
| **Entity** | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_SCP03T\_STRUCTURE\_ERROR)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #10 Error: PIR Error Reason – SCP03t Security Error

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_SCP03T\_SECURITY\_ERROR)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #11 Error: PIR Error Reason – install Failed Due To Iccid Already Exists On eUICC

|  |  |
| --- | --- |
| Initial Conditions | |
| **Entity** | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_ICCID\_ALREADY\_EXISTS)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #12 Error: PIR Error Reason – install Failed Due To Insufficient Memory For Profile

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_INSUFFICIENT\_MEMORY)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #13 Error: PIR Error Reason – install Failed Due To Interruption

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_INSTALL\_INTERRUPTION)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #14 Error: PIR Error Reason – install Failed Due To PE Processing Error

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_PE\_PROCESSING\_ERROR)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #15 Error: PIR Error Reason – install Failed Due To Data Mismatch

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_DATA\_MISMATCH)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #16 Error: PIR Error Reason – test Profile Install Failed Due To Invalid Naa Key

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF( #S\_PN\_PIR\_TEST\_PROFILE\_INVALID\_NAA\_KEY)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #17 Error: PIR Error Reason – PPR Not Allowed

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(  #S\_PN\_PIR\_PPR\_NOT\_ALLOWED)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

Test Sequence #18 Error: PIR Error Reason – install Failed Due To Unknown Error

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |
| 1 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_HANDLE\_NOTIF,   MTD\_HANDLE\_NOTIF(   #S\_PN\_PIR\_UNKNOWN\_ERROR)) | #R\_HTTP\_204\_OK | RQ25\_016 RQ25\_018 RQ25\_023 RQ25\_024 RQ25\_025 RQ25\_026 RQ31\_171 RQ31\_176 RQ31\_177 RQ31\_177\_1 RQ31\_178 RQ35\_017 RQ35\_019 RQ35\_022 RQ56\_042 RQ56\_042\_1 RQ56\_042\_2 RQ62\_001 RQ62\_002 RQ62\_009 RQ63\_005 RQ65\_006 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ31\_178 |

##### 4.3.15.2.2 TC\_SM-DP+\_ES9+\_HandleNotificationFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.3.15.2.3 TC\_SM-DP+\_ES9+\_HandleNotificationBRP

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for BRP. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * There have been no previous attempts to download pending Profile PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+. |

Test Sequence #01 Nominal: All Notifications

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.3.15.2.1 TC\_SM-DP+\_ES9+\_HandleNotificationNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

### 4.3.16 ES9+ (LPA -- SM-DP+): CancelSession

#### 4.3.16.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_118, RQ31\_119, RQ31\_120, RQ31\_121, RQ31\_122, RQ31\_123, RQ31\_123\_1, RQ31\_124, RQ31\_125, RQ31\_126, RQ31\_129, RQ31\_160

 RQ45\_006, RQ45\_026, RQ45\_026\_1

 RQ55\_048

 RQ56\_043, RQ56\_044, RQ56\_045, RQ56\_046, RQ56\_047, RQ56\_048, RQ56\_049

 RQ57\_114\_1, RQ57\_116

 RQ62\_001, RQ62\_002, RQ62\_003, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007, RQ62\_009

 RQ63\_004

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_025

#### 4.3.16.2 Test Cases

##### 4.3.16.2.1 TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientNIST

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * There have been no previous attempts to download the pending profile. |

Test Sequence #01 Nominal: End User Rejection after Authenticate Client

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'End User Rejection' reason after Authenticate Client, and that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_EU\_REJ)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ57\_114\_1 |

Test Sequence #02 Nominal: End User Postponed after Authenticate Client

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'End User postponed' reason after Authenticate Client, and the SM-DP+ keeps the RSP session’s corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_124 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #03 Nominal: Timeout after Authenticate Client

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'Timeout' reason after Authenticate Client, and the SM-DP+ keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |  | |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_TIMEOUT)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_124 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 | | |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | | RQ57\_114\_1 | |

Test Sequence #04 Nominal: PPR Not Allowed after Authenticate Client

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'PPR Not Allowed' reason after Authenticate Client, and that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is configured with #SMDP\_METADATA\_OP\_PROF1\_PPR2. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1\_PPR2 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC using #R\_AUTH\_CLIENT\_OK\_PPR2 instead of #R\_AUTH\_CLIENT\_OK | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_PPR\_NOT\_ALLOWED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ57\_114\_1 |

Test Sequence #05 Nominal: Undefined Reason after Authenticate Client

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'Undefined Reason' reason after Authenticate Client, and that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_UNDEFINED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 | |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | | RQ57\_114\_1 |

Test Sequence #06 Error: Unknown Transaction ID in JSON transport layer (Subject Code 8.10.1, Reason Code 3.9) after Authenticate Client

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using an Invalid Transaction ID after Authenticate Client, that the SM-DP+ returns a function execution status 'Failed' Subject Code 8.10.1, Reason Code 3.9, and keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <INVALID\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ65\_009 |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ63\_004 RQ65\_009 |
| 3 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #07 Error: Unknown Transaction ID in ASN.1 CancelSessionResponse Element (Subject Code 8.10.1, Reason Code 3.9) after Authenticate Client

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using an Invalid Transaction ID in the ASN.1 CancelSessionResponse element after Authenticate Client, that the SM-DP+ returns a function execution status 'Failed' with Subject Code 8.10.1, Reason Code 3.9, and keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>, #CS\_RESP\_ERROR\_8\_10\_1\_3\_9)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ65\_009 |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ63\_004 RQ65\_009 |
| 3 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #08 Error: Invalid eUICC Signature (Subject Code 8.1 Reason Code 6.1) after Authenticate Client

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using an Invalid Signature after Authenticate Client that the SM-DP+ returns a function execution status 'Failed' with Subject Code 8.1 Reason Code 6.1 and that the RSP session is stopped by the SM-DP+ and keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>,   #CS\_RESP\_ERROR\_8\_1\_6\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_123 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ65\_009 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #09 Error: Invalid OID (Subject Code 8.8 Reason Code 3.10) after Authenticate Client

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using an Invalid OID after Authenticate Client that the SM-DP+ returns a function execution status 'Failed' with Subject Code 8.8 Reason Code 3.10 and that the RSP session is stopped by the SM-DP+ and keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_ERROR\_8\_8\_3\_10)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_3\_10) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_123\_1 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ65\_009 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

##### 4.3.16.2.2 TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * There have been no previous attempts to download the pending profile. |

Test Sequence #01 Nominal: End User Rejection after GetBoundProfilePackage

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'End User Rejection' reason after GetBoundProfilePackage, and that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_EU\_REJ)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ31\_160 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ57\_114\_1 |

Test Sequence #02 Nominal: End User Postponed after GetBoundProfilePackage

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'End User postponed' reason after GetBoundProfilePackage, and the SM-DP+ keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_124 RQ31\_160 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #03 Nominal: Timeout after GetBoundProfilePackage

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'Timeout' reason after GetBoundProfilePackage , and the SM-DP+ keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_TIMEOUT)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_124 RQ31\_160 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #04 Nominal: PPR Not Allowed after GetBoundProfilePackage

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'PPR Not Allowed' reason after GetBoundProfilePackage, and that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 is configured with #SMDP\_METADATA\_OP\_PROF1\_PPR2. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1\_PPR2 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC using #R\_AUTH\_CLIENT\_OK\_PPR2 instead of #R\_AUTH\_CLIENT\_OK | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_PPR\_NOT\_ALLOWED)) | MTD\_HTTP\_RESP(  #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ31\_160 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ57\_114\_1 |

Test Sequence #05 Nominal: Metadata Mismatch after GetBoundProfilePackage

The purpose of this test is to verify that the LPAd can request the cancellation of an on-going RSP session using the 'Metadata Mismatch' reason after GetBoundProfilePackage, and that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_M\_DATA\_MISMATCH)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ31\_160 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ57\_114\_1 |

Test Sequence #06 Nominal: Load BPP Execution Error after GetBoundProfilePackage

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using that the 'loadBppExecutionError' reason after GetBoundProfilePackage, that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_L\_BPP\_EXE\_ERROR)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ31\_160 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ57\_114\_1 |

Test Sequence #07 Nominal: Undefined Reason after GetBoundProfilePackage

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using the 'Undefined Reason' reason after GetBoundProfilePackage, and that the RSP session is terminated by the SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_UNDEFINED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_122 RQ31\_125 RQ31\_126 RQ31\_129 RQ31\_160 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ55\_048 RQ56\_043 RQ56\_045 RQ56\_046 RQ56\_047 RQ56\_048 RQ57\_116 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_009 RQ63\_004 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_025 |
| 2 | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | | RQ57\_114\_1 |

Test Sequence #08 Error: Unknown Transaction ID in JSON transport layer (Subject Code 8.10.1, Reason Code 3.9) after GetBoundProfilePackage

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using an Invalid Transaction ID after GetBoundProfilePackage that the SM-DP+ returns a function execution status 'Failed' Subject Code 8.10.1, Reason Code 3.9 and keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <INVALID\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ65\_009 |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ31\_160 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ63\_004 RQ65\_009 |
| 3 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #09 Error: Unknown Transaction ID in ASN.1 CancelSessionResponse Element (Subject Code 8.10.1, Reason Code 3.9) after GetBoundProfilePackage

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using an Invalid Transaction ID in the ASN.1 CancelSessionResponse element after GetBoundProfilePackage that the SM-DP+ returns a function execution status 'Failed' with Subject Code 8.10.1, Reason Code 3.9 and keeps the RSP session's corresponding Profile download order in the 'Released' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>, #CS\_RESP\_ERROR\_8\_10\_1\_3\_9)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ65\_009 |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>, #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ31\_118 RQ31\_119 RQ31\_120 RQ31\_121 RQ31\_160 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ63\_004 RQ65\_009 |
| 3 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #10 Error: Invalid eUICC Signature (Subject Code 8.1 Reason Code 6.1) after GetBoundProfilePackage

The purpose of this test is to verify that if the LPAd can request the cancellation of an on-going RSP session using an Invalid Signature after GetBoundProfilePackage using S-ENC and S-MAC. But the SM-DP+ returns a function execution status 'Failed' with Subject Code 8.1 Reason Code 6.1 and that the RSP session is stopped by the SM-DP+ and keeps the RSP session's corresponding Profile download order in the 'Downloaded' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(   <S\_TRANSACTION\_ID>,   #CS\_RESP\_ERROR\_8\_1\_6\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_123 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ63\_004 RQ65\_009 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

Test Sequence #11 Error: Invalid OID (Subject Code 8.8 Reason Code 3.10) after GetBoundProfilePackage

The purpose of this test is to verify that if the LPAd requests the cancellation of an on-going RSP session using an Invalid OID after GetBoundProfilePackage that the SM-DP+ returns a function execution status 'Failed' with Subject Code 8.8 Reason Code 3.10 and that the RSP session is stopped by the SM-DP+ and keeps the RSP session's corresponding Profile download order in the 'Downloaded' state available for a further retry.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1 is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Confirmation Code is not provided by the Operator to the SM-DP+. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_RESP\_ERROR\_8\_8\_3\_10)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_3\_10) | RQ31\_118 RQ31\_119 RQ31\_121 RQ31\_123\_1 RQ56\_043 RQ56\_044 RQ56\_047 RQ56\_048 RQ56\_049 RQ62\_001 RQ62\_002 RQ63\_004 RQ65\_009 |
| 2 | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | | RQ57\_114\_1 |

##### 4.3.16.2.3 TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.3.16.2.4 TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageFRP

This test case is defined as FFS and not applicable for this version of test specification.

##### 4.3.16.2.5 TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientBRP

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for BRP. * PROFILE\_OPERATIONAL1 configured with #SMDP\_METADATA\_OP\_PROF1. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * The EID is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * There have been no previous attempts to download the pending profile. |

Test Sequence #01 Nominal: End User Rejection after Authenticate Client

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.3.16.2.1 TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #02 Nominal: End User Postponed after Authenticate Client

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.3.16.2.1 TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

##### 4.3.16.2.6 TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageBRP

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for BRP. |

Test Sequence #01 Nominal: End User Rejection after GetBoundProfilePackage

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.3.16.2.2 TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #02 Nominal: End User Postponed after GetBoundProfilePackage

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.3.16.2.2 TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageNIST except that all auth/pb keys and certificates SHALL be based on BrainpoolP256r1.

### 4.3.17 ES9+ (LPA -- SM-DP+): TLS, Server Authentication, Session Establishment

#### 4.3.17.1 TC\_SM-DP+\_ES9+\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST

Perform all test sequences defined in section 4.6.3.2.1 with the following variables set as follows:

 SERVER = SM-DP+ under test

o CERT\_SERVER\_TLS = #CERT\_SM\_DP\_TLS

#### 4.3.17.2 TC\_SM-DP+\_ES9+\_Server\_Authentication\_for\_HTTPS\_EstablishmentBRP

Perform all test sequences defined in section 4.6.3.2.2 with the following variables set as follows:

 SERVER = SM-DP+ under test

o CERT\_SERVER\_TLS = #CERT\_SM\_DP\_TLS

### 4.3.18 ES12 (SM-DP+ -- SM-DS): RegisterEvent

This test case is defined as FFS and not applicable for this version of test specification.

### 4.3.19 ES12 (SM-DP+ -- SM-DS): DeleteEvent

This test case is defined as FFS and not applicable for this version of test specification.

### 4.3.20 ES12 (SM-DP+ -- SM-DS): TLS, Mutual Authentication, Client, Session Establishment

#### 4.3.20.1 TC\_SM-DP+\_ES12\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST

Perform all test sequences defined in section 4.6.1.2.1 with the following variables set as follows:

 CLIENT = SM-DP+ under test

o CERT\_CLIENT\_TLS = #CERT\_SM\_DP\_TLS for NIST

 SERVER = S\_SM-DS

o CERT\_S\_SERVER\_TLS = #CERT\_S\_SM\_DS\_TLS for NIST

#### 4.3.20.2 TC\_SM-DP+\_ES12\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP

Perform all test sequences defined in section 4.6.1.2.2 with the following variables set as follows:

 CLIENT = SM-DP+ under test

o CERT\_CLIENT\_TLS = #CERT\_SM\_DP\_TLS for BRP

 SERVER = S\_SM-DS

o CERT\_S\_SERVER\_TLS = #CERT\_S\_SM\_DS\_TLS for BRP

## 4.4 LPAd Interfaces

### 4.4.1 ES10a (LPA -- eUICC): GetEuiccConfiguredAddresses

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.2 ES10a (LPA -- eUICC): SetDefaultDPAddress

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.3 ES10b (LPA -- eUICC): PrepareDownload

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.4 ES10b (LPA -- eUICC): LoadBoundProfilePackage

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.5 ES10b (LPA -- eUICC): GetEUICCChallenge

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.6 ES10b (LPA -- eUICC): GetEUICCInfo

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.7 ES10b (LPA -- eUICC): ListNotification

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.8 ES10b (LPA -- eUICC): RetrieveNotificationsList

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.9 ES10b (LPA -- eUICC): RemoveNotificationFromList

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.10 ES10b (LPA -- eUICC): LoadCRL

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.11 ES10b (LPA -- eUICC): AuthenticateServer

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.12 ES10b (LPA -- eUICC): CancelSession

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.13 ES10c (LPA -- eUICC): GetProfilesInfo

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.14 ES10c (LPA -- eUICC): EnableProfile

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.15 ES10c (LPA -- eUICC): DisableProfile

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.16 ES10c (LPA -- eUICC): DeleteProfile

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.17 ES10c (LPA -- eUICC): eUICCMemoryReset

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.18 ES10c (LPA -- eUICC): GetEID

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.19 ES10c (LPA -- eUICC): SetNickname

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.20 ES10b (LPA -- eUICC): GetRAT

This test case is defined as FFS and not applicable for this version of test specification.

### 4.4.21 ES9+ (LPA -- SM-DP+): InitiateAuthentication

#### 4.4.21.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ21\_001

 RQ31\_028, RQ31\_033, RQ31\_034, RQ31\_035, RQ31\_036, RQ31\_043, RQ31\_045, RQ31\_052, RQ31\_075

 RQ56\_004, RQ56\_005, RQ56\_006, RQ56\_007, RQ56\_008, RQ56\_011, RQ56\_012, RQ56\_009, RQ56\_010

 RQ62\_001, RQ62\_002, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007, RQ62\_008

 RQ63\_001\_1, RQ63\_004, RQ63\_006

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_017

#### 4.4.21.2 Test Cases

##### 4.4.21.2.1 TC\_LPAd\_InitiateAuthentication\_Nominal

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |

Test Sequence #01 Nominal: Initiate Authentication

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  <EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1))  • Extract <EUICC\_CHALLENGE> | RQ31\_028 RQ31\_033 RQ56\_004 RQ56\_005 RQ56\_006 RQ56\_007 RQ56\_012 |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_OK) | No error: Next step of common mutual authentication procedure is performed. | RQ31\_043 RQ56\_009 RQ56\_010 RQ62\_001 RQ62\_002 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_008 RQ63\_001\_1 RQ63\_004 RQ63\_006 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_017 |

##### 4.4.21.2.2 TC\_LPAd\_InitiateAuthentication\_ErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |

Test Sequence #01 Error: Invalid SM-DP+ Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_1\_3\_8) | LPAd aborts AddProfile procedure | RQ31\_034 RQ56\_008 RQ56\_011 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_034 RQ56\_008 RQ56\_011 |

Test Sequence #02 Error: Unsupported Security Configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_2\_3\_1) | LPAd aborts AddProfile procedure | RQ31\_035 RQ56\_008 RQ56\_011 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_035 RQ56\_008 RQ56\_011 |

Test Sequence #03 Error: Unsupported SVN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_3\_3\_1) | LPAd aborts AddProfile procedure | RQ56\_008 RQ56\_011 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_008 RQ56\_011 |

Test Sequence #04 Error: Unavailable SM-DP+ Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_4\_3\_7) | LPAd aborts AddProfile procedure | RQ31\_036 RQ56\_008 RQ56\_011 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_036 RQ56\_008 RQ56\_011 |

Test Sequence #05 Error: Invalid SM-DP+ Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_INV\_CERT) | LPAd aborts AddProfile procedure | RQ31\_052 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_052 |

Test Sequence #06 Error: Invalid SM-DP+ Signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_INV\_SIGN) | LPAd aborts AddProfile procedure | RQ31\_052 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_052 |

Test Sequence #07 Error: Invalid SM-DP+ Address sent by the SM-DP+

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_INV\_SMDP+\_ADDRESS) | LPAd informs the S\_EndUser and aborts the AddProfile procedure | RQ31\_045 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_045 |

Test Sequence #08 Error: Unsupported CI Key ID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_INV\_CI) | LPAd aborts AddProfile procedure | RQ31\_052 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_052 |

Test Sequence #09 Error: Invalid SM-DP+ OID

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated, #ACTIVATION\_CODE\_2 is provided to the LPAd on request from the S\_EndUser. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_2 (PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_INV\_OID) | LPAd informs the S\_EndUser and aborts the AddProfile procedure | RQ31\_075 |
| 2 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_075 |

### 4.4.22 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage

#### 4.4.22.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_112, RQ31\_113, RQ31\_141, RQ31\_146, RQ31\_147, RQ31\_148\_2

 RQ56\_015, RQ56\_018, RQ56\_022, RQ56\_024, RQ56\_025, RQ56\_026, RQ56\_027, RQ56\_028

 RQ65\_020

#### 4.4.22.2 Test Cases

##### 4.4.22.2.1 TC\_LPAd\_ES9+\_GetBoundProfilePackage\_Nominal

|  |  |  |
| --- | --- | --- |
| **General Initial Conditions** | | |
| **Entity** | **Description of the general initial condition** | |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. | |

Test Sequence #01 Nominal: Get BPP using S-ENC and S-MAC without Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC))  Verify:  • If <S\_TRANSACTION\_ID> is the same as in #R\_PREP\_DOWNLOAD\_NO\_CC  • <EUICC\_SIGNATURE2> using the #PK\_EUICC\_ECDSA | RQ31\_113 RQ31\_141 RQ31\_148\_2 RQ56\_024 RQ56\_026 RQ65\_020 |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_OK) | No error, see NOTE. | RQ56\_027 |
| NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested. | | | | |

Test Sequence #02 Nominal: Get BPP using S-ENC and S-MAC with Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT\_CC | | | |
| IC4 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_EndUser. | #CONFIRMATION\_CODE1 is provided by manual entry. |  |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_WITH\_CC))  Verify if:  • <S\_TRANSACTION\_ID> is the same as in #R\_PREP\_DOWNLOAD\_WITH\_CC  • <EUICC\_SIGNATURE2> using the #PK\_EUICC\_ECDSA  • <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | RQ31\_112 RQ31\_113 RQ31\_141 RQ31\_148\_2 RQ31\_146 RQ31\_147 RQ56\_015 RQ56\_024 |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_OK) | No error, see NOTE. | RQ56\_027 |
| NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested. | | | | |

Test Sequence #03 Nominal: Get BPP using PPK-ENC and PPK-MAC without Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC))  Verify:  • If <S\_TRANSACTION\_ID> is the same as in #R\_PREP\_DOWNLOAD\_NO\_CC  • <EUICC\_SIGNATURE2> using the #PK\_EUICC\_ECDSA | RQ31\_113 RQ31\_141 RQ31\_148\_2 RQ56\_024 RQ56\_026 RQ65\_020 |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_OK\_PPK) | No error, see NOTE. | RQ56\_027 |
| NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested. | | | | |

Test Sequence #04 Nominal: Get BPP using PPK-ENC and PPK-MAC with Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT\_CC | | | |
| IC4 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_EndUser. | #CONFIRMATION\_CODE1 is provided by manual entry. |  | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_WITH\_CC))  Verify if: • <S\_TRANSACTION\_ID> is the same as in #R\_PREP\_DOWNLOAD\_WITH\_CC  • <EUICC\_SIGNATURE2> using the #PK\_EUICC\_ECDSA  • <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | RQ31\_112 RQ31\_113 RQ31\_141 RQ31\_148\_2 RQ31\_146 RQ31\_147 RQ56\_015 RQ56\_024 RQ56\_026 | |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_OK\_PPK) | No error, see NOTE. | RQ56\_027 | |
| NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested. | | | | | |

##### 4.4.22.2.2 TC\_LPAd\_ES9+\_GetBoundProfilePackage\_Retry

|  |  |  |
| --- | --- | --- |
| **General Initial Conditions** | | |
| **Entity** | **Description of the general initial condition** | |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. | |

Test Sequence #01 Nominal: Get BPP Retry after incorrect Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT\_CC | | | |
| IC4 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_EndUser. | #CONFIRMATION\_CODE2 is provided by manual entry. |  |
| IC5 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_WITH\_CC))  Verify if: <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE2, <S\_TRANSACTION\_ID>) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_2\_7\_3\_8) | Continue to step 2 | RQ31\_148\_2 RQ56\_022 |
| 2 | S\_SM-DP+ closes TLS session (unless ,LPAd has already closed TLS session) | | | |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC | | | |
| 6 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_EndUser. | #CONFIRMATION\_CODE1 is provided by manual entry. | RQ31\_148\_3 RQ56\_022 |
| 7 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_WITH\_CC))  Verify if: • If <S\_TRANSACTION\_ID> is the same as in #R\_PREP\_DOWNLOAD\_WITH\_CC  • <EUICC\_SIGNATURE2> using the #PK\_EUICC\_ECDSA • <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | RQ31\_148\_3 RQ56\_022 RQ56\_026 |
| 8 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_OK) | No error, see NOTE 1. | RQ56\_024 RQ56\_027 |
| NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested. | | | | |

##### 4.4.22.2.3 TC\_LPAd\_ES9+\_GetBoundProfilePackage\_Error

|  |  |  |
| --- | --- | --- |
| General Initial Conditions | | |
| Entity | Description of the general initial condition | |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. | |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). | |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. | |

Test Sequence #01 Error: Wrong eUICC Signature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  | |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_1\_6\_1) | LPAd aborts AddProfile procedure  NOTE: The LPAd MAY retry by restarting the Profile download and installation procedure. | RQ56\_018 RQ56\_025 RQ56\_028 | |

Test Sequence #02 Error: BPP Not Available

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  | |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_2\_3\_7) | LPAd aborts AddProfile procedure  NOTE: the LPAd MAY retry by restarting the Profile download and installation procedure. | RQ56\_028 | |

Test Sequence #03 Error: Unknown TransactionID received by SM-DP+

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | LPAd → S\_SM-DP+ | Send ES9+.GetBundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  | |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_10\_1\_3\_9) | LPAd aborts AddProfile procedure  NOTE: the LPAd MAY retry by restarting the Profile download and installation procedure. | RQ56\_018 RQ56\_025 RQ56\_028 | |

Test Sequence #04 Error: Missing Confirmation Code

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_2\_7\_2\_2) | LPAd aborts AddProfile procedure  NOTE: the LPAd MAY retry by restarting the Profile download and installation procedure. | RQ56\_018 RQ56\_025 RQ56\_028 |

Test Sequence #05 Error: Download Order Expired

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_8\_5\_4\_10) | LPAd aborts AddProfile procedure  NOTE: The LPAd MAY retry by restarting the Profile download and installation procedure. | RQ56\_018 RQ56\_025 RQ56\_028 |

Test Sequence #06 Error: Wrong Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT\_CC | | | |
| IC4 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_EndUser. | #CONFIRMATION\_CODE2 is provided by manual entry. |  | |
| IC5 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_WITH\_CC)) |  | |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_2\_7\_3\_8) | LPAd aborts AddProfile procedure  NOTE: The LPAd MAY retry by restarting the Profile download and installation procedure | RQ56\_018 RQ56\_025 RQ56\_028 | |

Test Sequence #07 Error: Maximum number of Confirmation Code retries exceeded

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT\_CC | | | |
| IC4 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_EndUser. | #CONFIRMATION\_CODE2 is provided by manual entry. |  |
| IC5 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_WITH\_CC)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_2\_7\_6\_4) | LPAd aborts AddProfile procedure  The LPAd SHALL NOT retry by restarting the Profile download and installation procedure. | RQ56\_018 RQ56\_025 RQ56\_028 RQ31\_148\_2 |

### 4.4.23 ES9+ (LPA -- SM-DP+): AuthenticateClient

#### 4.4.23.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ21\_001, RQ21\_002

 RQ31\_032, RQ31\_033, RQ31\_043, RQ31\_046, RQ31\_055, RQ31\_056, RQ31\_057, RQ31\_060, RQ31\_061, RQ31\_073, RQ31\_076, RQ31\_083, RQ31\_085, RQ31\_090, RQ31\_091, RQ31\_095, RQ31\_136

 RQ42\_001, RQ42\_002, RQ42\_003, RQ42\_004, RQ42\_005, RQ42\_006, RQ42\_007, RQ42\_008, RQ42\_009, RQ42\_010, RQ42\_011, RQ42\_012, RQ42\_013, RQ42\_014, RQ42\_015, RQ42\_016, RQ42\_017, RQ42\_018, RQ42\_019, RQ42\_020, RQ43\_001

 RQ56\_001, RQ56\_004, RQ56\_005, RQ56\_009, RQ56\_010, RQ56\_029, RQ56\_030, RQ56\_031\_1, RQ56\_033, RQ56\_037, RQ56\_038, RQ56\_039, RQ56\_040, RQ56\_041, RQ56\_041\_1, RQ56\_041\_2

 RQ57\_031

 RQ62\_001, RQ62\_002, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007, RQ62\_008

 RQ63\_001\_1, RQ63\_004, RQ63\_006

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_019, RQ65\_022

#### 4.4.23.2 Test Cases

##### 4.4.23.2.1 TC\_LPAd\_AuthenticateClient\_Nominal

|  |  |  |
| --- | --- | --- |
| General Initial Conditions | | |
| Entity | Description of the general initial condition | |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. |

Test Sequence #01 Nominal: Authenticate Client without Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  <EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1))  • Extract <EUICC\_CHALLENGE> |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_OK) | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO))  Verify:  • if #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO used with the #MATCHING\_ID\_1 • If <S\_TRANSACTION\_ID> is the same as in #INITIATE\_AUTH\_OK • <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • if <S\_SMDP\_CHALLENGE> present in the #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO is the same as in <S\_SMDP\_SIGNED1> present in #INITIATE\_AUTH\_OK • for #DEVICE\_INFO: - The value of the TAC corresponds to the first 8 digits of #IUT\_IMEI and is represented as a string of 4 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] - if IMEI is present then its value corresponds to #IUT\_IMEI and is represented as a string of 8 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] except that the last octet contains the check digit (in high nibble) and an'F' filler (in low nibble)  - if O\_D\_GSM\_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT\_GSM\_GERAN\_REL. – if O\_D\_UMTS\_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT\_UMTS\_UTRAN\_REL. – if O\_D\_CDMA2000\_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_1X\_REL. – if O\_D\_CDMA2000\_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_HRPD\_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O\_D\_CDMA2000\_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_EHRPD\_REL. – if O\_D\_LTE then eutranSupportedRelease (or eutranEpcSupportedRelease if #IUT\_RSP\_VERSION is v2.2.2 or higher) is set to the highest release as defined in #IUT\_LTE\_EUTRAN\_REL. – if O\_D\_NFC\_TS26 then contactlessSupportedRelease is set to the highest release as defined in #IUT\_NFC\_REL. – if O\_D\_CRL then rspCrlSupportedVersion is set to the highest release as defined in #IUT\_RSP\_VERSION.  For each of the options O\_D\_GSM\_GERAN, O\_D\_UMTS\_UTRAN, O\_D\_CDMA2000\_1X, O\_D\_CDMA2000\_HRPD, O\_D\_CDMA2000\_EHRPD, O\_D\_LTE, O\_D\_NFC\_TS26 or O\_D\_CRL, if the option is not set, verify that the corresponding field in DeviceCapabilities is not present. | RQ21\_001 RQ21\_002 RQ31\_043 RQ31\_046 RQ31\_055 RQ31\_056 RQ31\_057 RQ31\_060 RQ31\_076 RQ42\_001 RQ42\_002 RQ42\_003 RQ42\_004 RQ42\_005 RQ42\_006 RQ42\_007 RQ42\_008 RQ42\_009 RQ42\_010 RQ42\_011 RQ42\_012 RQ42\_013 RQ42\_014 RQ42\_015 RQ42\_016 RQ42\_017 RQ42\_018 RQ42\_019 RQ42\_020 RQ43\_001 RQ56\_009 RQ56\_010 RQ56\_029 RQ56\_039 RQ62\_001 RQ62\_002 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_008 RQ63\_001\_1 RQ63\_004 RQ63\_006 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_019 RQ65\_022 |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_OK) | No Error | RQ31\_073 RQ31\_095 RQ56\_037 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 |

Test Sequence #02 Nominal: Authenticate Client with Confirmation Code

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  <EUICC\_CHALLENGE>,  #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1))  • Extract <EUICC\_CHALLENGE> |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_OK) | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO))  Verify:  • if #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO used with the #MATCHING\_ID\_3  • If <S\_TRANSACTION\_ID> is the same as in #INITIATE\_AUTH\_OK  • <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • if <S\_SMDP\_CHALLENGE> present in the #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO is the same as in <S\_SMDP\_SIGNED1> present in #INITIATE\_AUTH\_OK  • for #DEVICE\_INFO:  - The value of the TAC corresponds to the first 8 digits of #IUT\_IMEI and is represented as a string of 4 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12]  - if IMEI is present then its value corresponds to #IUT\_IMEI and is represented as a string of 8 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] except that the last octet contains the check digit (in high nibble) and an 'F' filler (in low nibble)  - if O\_D\_GSM\_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT\_GSM\_GERAN\_REL.  – if O\_D\_UMTS\_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT\_UMTS\_UTRAN\_REL.  – if O\_D\_CDMA2000\_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_1X\_REL.  – if O\_D\_CDMA2000\_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_HRPD\_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively.  – if O\_D\_CDMA2000\_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_EHRPD\_REL  – if O\_D\_LTE then eutranSupportedRelease (or eutranEpcSupportedRelease if #IUT\_RSP\_VERSION is v2.2.2 or higher) is set to the highest release as defined in #IUT\_LTE\_EUTRAN\_REL.  – if O\_D\_NFC\_TS26 then contactlessSupportedRelease is set to the highest release as defined in #IUT\_NFC\_REL.  – if O\_D\_CRL then rspCrlSupportedVersion is set to the highest release as defined in #IUT\_RSP\_VERSION.  For each of the options O\_D\_GSM\_GERAN, O\_D\_UMTS\_UTRAN, O\_D\_CDMA2000\_1X, O\_D\_CDMA2000\_HRPD, O\_D\_CDMA2000\_EHRPD, O\_D\_LTE, O\_D\_NFC\_TS26 or O\_D\_CRL, if the option is not set, verify that the corresponding field in DeviceCapabilities is not present. | RQ21\_001 RQ21\_002 RQ31\_043 RQ31\_046 RQ31\_055 RQ31\_056 RQ31\_057 RQ31\_060 RQ31\_076 RQ42\_001 RQ42\_002 RQ42\_003 RQ42\_004 RQ42\_005 RQ42\_006 RQ42\_007 RQ42\_008 RQ42\_009 RQ42\_010 RQ42\_011 RQ42\_012 RQ42\_013 RQ42\_014 RQ42\_015 RQ42\_016 RQ42\_017 RQ42\_018 RQ42\_019 RQ42\_020 RQ43\_001 RQ56\_009 RQ56\_010 RQ56\_029 RQ56\_039 RQ62\_001 RQ62\_002 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_008 RQ63\_001\_1 RQ63\_004 RQ63\_006 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_022 |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_OK\_CC) | No Error | RQ31\_073 RQ31\_095 RQ56\_037 RQ56\_040 RQ56\_041\_1 RQ56\_041\_2 |

Test Sequence #03 Nominal: Authenticate Client with Confirmation Code Retry

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT\_CC | | | | |
| IC4 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_EndUser. | #CONFIRMATION\_CODE2 is provided by manual entry. | |  |
| IC5 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | Verify if: <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE2, <S\_TRANSACTION\_ID>) | |  |
| IC6 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_2\_7\_3\_8) |  | |  |
| IC7 | Restart Add Profile procedure if O\_D\_CC\_RETRY not supported | | | | |
| IC8 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC9 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC10 | S\_SM-DP+ → LPAd | Send ES9+.AuthenticateClient method | | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_OK\_CC) | | No Error | RQ31\_091 |

##### 4.4.23.2.2 TC\_LPAd\_AuthenticateClient\_ErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| S\_SM-DP+ | There is a pending Profile download order for MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |
| Device | The protection of access to the LUI is disabled. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |

Test Sequence #01 Error: Invalid EUM Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | LPAd aborts AddProfile procedure | RQ31\_061 RQ56\_030 RQ56\_038 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #02 Error: Expired EUM Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_3) | LPAd aborts AddProfile procedure | RQ31\_061 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #03 Error: Invalid eUICC Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | LPAd aborts AddProfile procedure | RQ31\_061 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #04 Error: Expired eUICC Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_3) | LPAd aborts AddProfile procedure | RQ31\_061 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #05 Error: Invalid eUICC Signature or serverChallenge

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | LPAd aborts AddProfile procedure | RQ31\_061 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #06 Error: Insufficient Memory

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_4\_8) | LPAd aborts AddProfile procedure | RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #07 Error: Unknown CI Root Key

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_11\_1\_3\_9) | LPAd aborts AddProfile procedure | RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #08 Error: Profile not Allowed (Not in 'released' State)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_2) | LPAd aborts AddProfile procedure | RQ31\_083 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_033 RQ56\_041 |

Test Sequence #09 Error: Unknown TransactionID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | LPAd aborts AddProfile procedure | RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #10 Error: Refused MatchingID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_8) | LPAd aborts AddProfile procedure | RQ31\_083 RQ31\_090 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_033 RQ56\_041 |

Test Sequence #11 Error: Refused EID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_1\_3\_8) | LPAd aborts AddProfile procedure | RQ31\_083 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #12 Error: No Eligible Profile for this eUICC/Device

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_5\_4\_3) | LPAd aborts AddProfile procedure | RQ31\_090 RQ31\_083 RQ56\_030 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #13 Error: Expired Download Order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_5\_4\_10) | LPAd aborts AddProfile procedure | RQ31\_090 RQ56\_030 RQ56\_031 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_033 RQ56\_041 |

Test Sequence #14 Error: Maximum Number of Retries Exceeded

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_8\_5\_6\_4) | LPAd aborts AddProfile procedure | RQ31\_085 RQ56\_030 RQ56\_031\_1 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #15 Error: Invalid SM-DP+(pb) certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #AUTH\_CLIENT\_INV\_PB\_CERT) | LPAd aborts AddProfile procedure (See NOTE) | RQ31\_136 RQ57\_031 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_136 RQ57\_031 |
| NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S\_EndUser. In this case the S\_EndUser SHALL give the Confirmation. | | | | |

Test Sequence #16 Error: Different OID for SM-DP+ Certificates (CERT.DPpb.ECDSA and CERT.DPauth.ECDSA not belonging to the same entity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | AuthenticateClient | MTD\_HTTP\_REQ(#TEST\_DP\_ADDRESS1,  #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#AUTH\_CLIENT\_INV\_CI) | LPAd aborts AddProfile procedure (See NOTE) | RQ31\_136 RQ57\_031 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_136 RQ57\_031 |
| NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S\_EndUser. In this case the S\_EndUser SHALL give the Confirmation. | | | | |

Test Sequence #17 Error: Invalid SM-DP+ signature (smdpSignature2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #AUTH\_CLIENT\_INV\_SIGN) | LPAd aborts AddProfile procedure (See NOTE) | RQ31\_136 RQ57\_031 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_136 RQ57\_031 |
| NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S\_EndUser. In this case the S\_EndUser SHALL give the Confirmation. | | | | |

Test Sequence #18 Error: Invalid TransactionID sent by SM-DP+

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient Method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #AUTH\_CLIENT\_INV\_TRANSACTION\_ID) | LPAd aborts AddProfile procedure (See NOTE) | RQ31\_136 RQ57\_031 |
| 3 | LPAd → S\_SM-DP+ | No Profile download action | No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_136 RQ57\_031 |
| NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S\_EndUser. In this case the S\_EndUser SHALL give the Confirmation. | | | | |

### 4.4.24 ES9+ (LPA – SM-DP+): HandleNotification

#### 4.4.24.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_171, RQ31\_173, RQ31\_176

 RQ35\_008, RQ35\_012, RQ35\_013, RQ35\_014, RQ35\_014\_3, RQ35\_017, RQ35\_018, RQ35\_022

 RQ56\_042, RQ62\_003, RQ62\_009, RQ63\_005, RQ65\_024, RQC3\_003

#### 4.4.24.2 Test Cases

##### 4.4.24.2.1 TC\_LPAd\_ES9+\_HandleNotification\_Nominal

Throughout all the test cases the maximum number of Notifications simultaneously tested has been set as to two as there is not minimum defined in SGP.21 [3] or SGP.22 [2] for the number of Notifications that can be stored by the eUICC.

|  |  |  |
| --- | --- | --- |
| General Initial Conditions | | |
| Entity | Description of the general initial condition | |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). | |
| S\_SM-DP+ | S\_SM-DP+(1) is configured with #TEST\_DP\_ADDRESS1 and #CERT\_S\_SM\_DP\_TLS.  S\_SM-DP+(2) is configured with #TEST\_DP\_ADDRESS2 and #CERT\_S\_SM\_DP2\_TLS. | |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. |

Test Sequence #01 Nominal: Successful PIR and Install Notifications to the Same SM-DP+ Address

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1 for PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  (s NOTE 1) | | | |
| 1 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#R\_PIR\_OK)) • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ31\_171 RQ31\_176 RQ35\_008 RQ35\_013 RQ35\_017 RQ35\_018 RQ62\_003 RQ65\_024 RQC3\_003 |
| 2 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd.  The LPAd MAY inform the End User of the success status indicated by the Profile Installation Result. | RQ35\_008 RQ35\_014 RQ35\_017 RQ56\_042 RQ62\_003 RQ62\_009 RQ63\_005 |
| 3 | LPAd → S\_SM-DP+(1) | Establish an HTTPs connection if previously closed | | |
| 4 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(  #PENDING\_NOTIF\_INST1))  sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_008 RQ35\_013 RQ35\_014 RQ35\_022 RQ35\_018 RQ62\_003 RQ65\_024 RQC3\_003 |
| 5 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_022 RQ56\_042 RQ62\_003 RQ62\_009 RQ63\_005 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session.  NOTE 2: The timeout SHALL start after the PIR is received.  NOTE 3: In case the AddProfile initiation was combined with “Enable” (i.e. O\_D\_ADD\_ENABLE\_SEPARATED is not supported), any subsequent Enable Notification is not part of the test sequence. | | | | |

Test Sequence #02 Nominal: Successful PIR and Enable Notifications to the Same SM-DP+ Address

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1 for PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_EN instead of #METADATA\_OP\_PROF1. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  (s. NOTE 1) | | | |
| 1 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#R\_PIR\_OK))  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ31\_171 RQ31\_176 RQ35\_008 RQ35\_013 RQ35\_017 RQ35\_018 | |
| 2 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd.  The LPAd MAY inform the End User of the success status indicated by the Profile Installation Result. | RQ35\_008 RQ35\_014 RQ35\_017 | |
| 3 | S\_EndUser → LPAd | If PROFILE\_OPERATIONAL1 is not already enabled (see NOTE 3), initiate the Enable Profile operation for PROFILE\_OPERATIONAL1. | PROFILE\_OPERATIONAL1 is enabled |  | |
| 4 | LPAd → S\_SM-DP+(1) | Establish an HTTPs connection if previously closed | | | |
| 5 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF( #PENDING\_NOTIF\_EN1))  sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_008 RQ35\_013 RQ35\_014 RQ35\_022 RQ35\_018 | |
| 6 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_022 | |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session.  NOTE 2: The timeout SHALL start after the initiation of the Enable Profile operation.  NOTE 3: PROFILE\_OPERATIONAL1 is expected to be already enabled only in the case that the device supports only O\_D\_ADD\_ENABLE\_COMBINED. | | | | | |

Test Sequence #03 Nominal: Disable and Delete Notifications to the Same SM-DP+ Address

|  |
| --- |
| Initial Conditions |
| **Entity** | Description of the initial condition |
| eUICC | PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | PROFILE\_OPERATIONAL1 is in the Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Disable Profile operation for PROFILE\_OPERATIONAL1 | PROFILE\_OPERATIONAL1 is disabled | RQ32\_001 |
| 2 | LPAd → S\_SM-DP+(1) | Establish an HTTPs connection if previously closed | | |
| 3 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS1))  sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT (see NOTE 1)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_008 RQ35\_013 RQ35\_017 RQ35\_018 |
| 4 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_014 RQ35\_017 |
| 5 | LPAd → S\_SM-DP+(1) | Establish an HTTPs connection if previously closed | | |
| 6 | S\_EndUser → LPAd | Initiate the Delete Profile operation for PROFILE\_OPERATIONAL1 | Successful End User Intent verified  PROFILE\_OPERATIONAL1 is deleted |  |
| 7 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL1)) sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT (see NOTE 2)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_008 RQ35\_013 RQ35\_014 RQ35\_022 RQ35\_018 |
| 8 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_022 |
| NOTE 1: The timeout SHALL start after the initiation of the Disable Profile operation.  NOTE 2: The timeout SHALL start after the End User Intent verification. | | | | |

Test Sequence #04 Nominal: Enable and Disable Notifications with Different SM-DP+ Addresses

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| eUICC | PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | PROFILE\_OPERATIONAL2 is installed on the eUICC. |
| eUICC | PROFILE\_OPERATIONAL1 is in the Enabled state. |
| eUICC | PROFILE\_OPERATIONAL2 is in the Disabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Enable Profile operation for PROFILE\_OPERATIONAL2 | PROFILE\_OPERATIONAL2 is enabled | RQ32\_001 |
| 2 | LPAd → S\_SM-DP+(1) | Establish an HTTPs connection if previously closed | | |
| 3 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS1)) sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_012 RQ35\_008 RQ35\_013 RQ35\_014\_3 RQ35\_017 RQ35\_018 |
| 4 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_014 RQ35\_017 |
| 5 | LPAd → S\_SM-DP+(2) | Establish an HTTPs connection | | |
| 6 | LPAd → S\_SM-DP+(2) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS2,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_EN2)) sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_012 RQ35\_008 RQ35\_013 RQ35\_014 RQ35\_014\_3 RQ35\_022 RQ35\_018 |
| 7 | S\_SM-DP+(2) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_022 |
| NOTE 1: Steps 2,3 and 4 can be executed in parallel to the steps 5, 6 and 7.  NOTE 2: The timeout SHALL start after the initiation of the Enable Profile operation. | | | | |

Test Sequence #05 Nominal: Different SM-DP+ Addresses in PIR and Install Notifications

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1 for PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_INST\_DIFF instead of #METADATA\_OP\_PROF1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | |
| IC4 | PROC\_ES9+\_GET\_BPP(s. NOTE 1) | | | |
| 1 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#R\_PIR\_OK)) | RQ35\_012 RQ35\_008 RQ35\_013 RQ35\_017 RQ35\_018 |
| 2 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd.  The LPAd MAY inform the End User of the success status indicated by the Profile Installation Result. | RQ35\_008 RQ35\_014 RQ35\_017 |
| 3 | LPAd → S\_SM-DP+(2) | Establish an HTTPs connection | | |
| 4 | LPAd → S\_SM-DP+(2) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS2,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_INST\_ADDRESS2)) sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_012 RQ35\_008 RQ35\_013 RQ35\_014 RQ35\_022 RQ35\_018 |
| 5 | S\_SM-DP+(2) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_022 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session.  NOTE 2: Steps 1 and 2 can be executed in parallel to the steps 3,4 and 5.  NOTE 3: The timeout SHALL start after the End User Intent verification. | | | | |

Test Sequence #06 Nominal: Profile Download with PIR Failed

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1 for PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT | | | | |
| IC4 | LPAd → S\_SM-DP+(1) | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  | |
| IC5 | S\_SM-DP+(1) → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_INV) | No error exhibited by the LPAd, s. note 1. |  | |
| 1 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#R\_PIR\_SECU\_INVALID))  • Verify the euiccSignPIR <EUICC\_SIGN\_PIR> using the #PK\_EUICC\_ECDSA | RQ31\_171 RQ31\_173 RQ31\_176 RQ35\_008 RQ35\_012 RQ35\_013 RQ35\_014 | |
| 2 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd.  The LPAd MAY inform the End User of the error status indicated by the Profile Installation Result. | RQ35\_008 | |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session. | | | | |

Test Sequence #07 Nominal: Successful PIR and Install Notifications after Connectivity Interruption

This Test Sequence is FFS.

Test Sequence #08 Nominal: No Acknowledge for Successful PIR results in No Further Notifications

The purpose of this test case is to verify that the next Notification of a group is not sent until LPA receives a successful response from the SM-DP+ for the previous Notification.

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1 for PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  (s. NOTE 1) | | | |
| 1 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method initiated | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#R\_PIR\_OK)) |  | |
| 2 | LPAd → S\_SM-DP+(1) | No ES9+.HandleNotification method sent | No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  OR  TLS Session closed independent of timeout. | RQ35\_014 | |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session.  NOTE 2: The timeout in Step 3 SHALL start after the End User Intent verification. | | | | |

Test Sequence #09 Nominal: Disable and Delete Notifications to the Same SM-DP+ Address using Delete Operation

|  |
| --- |
| Initial Conditions |
| **Entity** | Description of the initial condition |
| eUICC | PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | PROFILE\_OPERATIONAL1 is in the Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Delete Profile operation for PROFILE\_OPERATIONAL1 and provide requested Confirmation | PROFILE\_OPERATIONAL1 is deleted | RQ32\_001 |
| 2 | LPAd → S\_SM-DP+(1) | Establish an HTTPs connection if previously closed | | |
| 3 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS1))  sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT (see NOTE)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_008 RQ35\_013 RQ35\_017 RQ35\_018 |
| 4 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_014 RQ35\_017 |
| 5 | LPAd → S\_SM-DP+(1) | Establish an HTTPs connection if previously closed | | |
| 6 | LPAd → S\_SM-DP+(1) | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL1)) sent within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT (see NOTE)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_008 RQ35\_013 RQ35\_014 RQ35\_022 RQ35\_018 |
| 7 | S\_SM-DP+(1) → LPAd | #R\_HTTP\_204\_OK | No error exhibited by the LPAd | RQ35\_008 RQ35\_022 |
| NOTE: The timeout SHALL start after the End User Intent verification. | | | | |

### 4.4.25 ES9+ (LPA – SM-DP+): CancelSession

#### 4.4.25.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ29\_011, RQ29\_012, RQ29\_013, RQ29\_014, RQ29\_018, RQ29\_007\_1, RQ29\_008, RQ29\_008\_1, RQ29\_009, RQ29\_015

 RQ31\_096, RQ31\_099, RQ31\_100, RQ31\_101, RQ31\_102, RQ31\_103, RQ31\_105, RQ31\_111, RQ31\_114, RQ31\_117, RQ31\_118, RQ31\_120, RQ31\_121, RQ31\_123, RQ31\_123\_1, RQ31\_124, RQ31\_129, RQ31\_159, RQ31\_160, RQ31\_162\_1, RQ31\_186\_1

 RQ56\_044, RQ56\_047

 RQ65\_025

#### 4.4.25.2 Test Cases

##### 4.4.25.2.1 TC\_LPAd\_ES9+\_CancelSession\_Nominal

|  |  |  |
| --- | --- | --- |
| **General Initial Conditions** | | |
| **Entity** | **Description of the general initial condition** | |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. | |

Test Sequence #01 Nominal: Profile Download with PPR1 not allowed due to Operational Profile already present

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_4. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_4 (associated with PROFILE\_OPERATIONAL4). |
| S\_SM-DP+ | The S\_SM-DP+ is configured to ignore the forbidden PPR during the eligibility check. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | | |
| IC4 | PROC\_ES9+\_GET\_BPP with #METADATA\_OP\_PROF4 used in #GET\_BPP\_OK  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_OK\_PPR\_NOT\_ALLOWED))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | RQ31\_099 RQ56\_044 RQ56\_047 RQ65\_025 RQ31\_114 RQ31\_117 RQ31\_118 RQ31\_120 | |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_SUCCESS) | If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.  OR  If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | RQ31\_099 | | |

Test Sequence #02 Nominal: End User rejection

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download. | RQ31\_096 |
| 2 | S\_EndUser → LPAd | End User Rejection (or failed confirmation) is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_REJ))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | RQ56\_044 RQ56\_047 RQ65\_025 RQ31\_114 RQ31\_117 RQ31\_118 RQ31\_120 |
| 4 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_SUCCESS) | If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.  OR  If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | RQ31\_114 |

Test Sequence #03 Nominal: Load BPP Error

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #GET\_BPP\_LOAD\_ERROR) | Continue to step 2 (End User Confirmation) if requested, otherwise continue with Step 3 |  |
| 2 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | End User Intent successfully verified. |  |
| 3 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_OK\_EU\_LOAD\_BPP\_ERROR))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | RQ31\_129 RQ56\_044 RQ56\_047 RQ65\_025 RQ31\_114 RQ31\_117 RQ31\_118 RQ31\_120 RQ31\_162\_1 |
| 4 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_SUCCESS) | No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | RQ31\_129 |

Test Sequence #04 Nominal: End User Timeout

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation of the Profile Download. | RQ31\_096 RQ31\_159 | |
| 2 | S\_EndUser → LPAd | No End User Rejection or Confirmation is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  | |
| 3 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_TIMEOUT))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | RQ56\_044 RQ56\_047RQ65\_025 RQ31\_114 RQ31\_124 RQ56\_044 RQ56\_047 RQ65\_025 RQ31\_117 RQ31\_118 RQ31\_120 RQ31\_111 | |
| 4 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_SUCCESS) | If Step 1 was performed directly after IC3:  No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.  OR  If Step 1 was performed after IC4:  No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | RQ31\_114 | |

Test Sequence #05 Nominal: Load BPP Error due to unknown TAG

|  |
| --- |
| **Initial Conditions** |
| **Entity** | **Description of the initial condition** |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  |
| 1 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_LOAD\_ERROR\_UNKNOWN\_TAG) | Continue to step 2 (End User Confirmation) if requested, otherwise continue with Step 3 |  |
| 2 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | End User Intent successfully verified. |  |
| 3 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_OK\_EU\_LOAD\_BPP\_ERROR))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | RQ31\_129 RQ56\_044 RQ56\_047 RQ65\_025 RQ31\_114 RQ31\_186\_1 RQ31\_162\_1 |
| 4 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_SUCCESS) | No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | RQ31\_129 |

##### 4.4.25.2.2 TC\_LPAd\_ES9+\_CancelSession\_EndUserPostponed\_Nominal

|  |  |  |
| --- | --- | --- |
| **General Initial Conditions** | | |
| **Entity** | **Description of the general initial condition** | |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. | |

Test Sequence #01 Nominal: End User Postponed

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download. | | RQ31\_096 |
| 2 | S\_EndUser → LPAd | End User Postpone is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  | |  |
| 3 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_POSTPONED))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | | RQ56\_044 RQ56\_047 RQ65\_025 RQ31\_114 |
| 4 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_SUCCESS) | If Step 1 was performed directly after IC3:  No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.  OR  If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | | RQ31\_114 |

##### 4.4.25.2.3 TC\_LPAd\_ES9+\_CancelSession\_Error

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. |

Test Sequence #01 Error: Unknown TransactionID after End User Rejection/Postpone

|  |
| --- |
| Initial Conditions |
| **Entity** | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID>  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | |
| IC5 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download. |  |
| IC6 | S\_EndUser → LPAd | End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_REJ))  OR  MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_POSTPONED)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | No error after receiving the HTTPs response. (See NOTE) | RQ56\_044 RQ56\_047 RQ56\_049 RQ31\_121 |
| NOTE: The LPA MAY either stop or retry sending ES9+.CancelSession method. | | | | |

Test Sequence #02 Error: Invalid eUICC Signature after End User Rejection/Postpone

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID>  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | |
| IC5 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download. |  |
| IC6 | S\_EndUser → LPAd | End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_REJ))  OR  MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_POSTPONED)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_1\_6\_1) | No error after receiving the HTTPs response.  The LPA SHALL stop the procedure: no ES9+.CancelSession requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.. | RQ56\_044 RQ56\_047 RQ56\_049 RQ31\_123 |

Test Sequence #03 Error: Invalid SM-DP+ OID after End User Rejection/Postpone

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID>  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | |
| IC5 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download. |  |
| IC6 | S\_EndUser → LPAd | End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_REJ))  OR  MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_POSTPONED)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_8\_3\_10) | No error after receiving the HTTPs response.  The LPA SHALL stop the procedure: no ES9+.CancelSession requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.. | RQ56\_044 RQ56\_047 RQ56\_049 RQ31\_123\_1 |

##### 4.4.25.2.4 TC\_LPAd\_ES9+\_CancelSession\_PPRs

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. |

Test Sequence #01 Nominal: End User rejection/postpone after PPR1 consent requested

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed and End User Consent is required for #MCC\_MNC4 with gid1 and gid2 absent. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_4. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_4 (associated with PROFILE\_OPERATIONAL4). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP with #METADATA\_OP\_PROF4 used in #GET\_BPP\_OK  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download.  For LPAd supporting SGP.22 v2.2.2 or earlier:  Relevant information about PPRs is shown, including consequences for the End User, and the End User consent is requested if not requested before.  For LPAd supporting SGP.22 v2.3 or later:  Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download. | RQ31\_102 RQ31\_103 RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_015 RQ29\_011 RQ29\_013 RQ29\_018 | |
| 2 | S\_EndUser → LPAd | End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  | |
| 3 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_REJ))  OR  MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_POSTPONED))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | RQ31\_100 RQ31\_105 RQ31\_160 | |
| 4 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_SUCCESS) | If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.  OR  If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | RQ31\_100 RQ31\_160 | |

Test Sequence #02 Nominal: End User rejection/posptone after PPR2 consent requested

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR2 is allowed and End User Consent is required for #MCC\_MNC2 with gid1 and gid2 absent. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3\_NO\_CC. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL3). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP with #METADATA\_OP\_PROF3 used in #GET\_BPP\_OK  This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method) | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download.  For LPAd supporting SGP.22 v2.2.2 or earlier:  Relevant information about PPRs is shown, including consequences for the End User, and the End User consent is requested if not requested before.  For LPAd supporting SGP.22 v2.3 or later:  Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download. | RQ31\_102 RQ31\_103 RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_015 RQ29\_011 RQ29\_013 RQ29\_018 |
| 2 | S\_EndUser → LPAd | End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | LPAd → S\_SM-DP+ | Send ES9+.CancelSession method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_OK\_EU\_REJ))  OR  MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,   #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>, #CS\_OK\_EU\_POSTPONED))  Verify: •<EUICC\_CANCEL\_SESSION\_SIGNATURE> with the #PK\_EUICC\_ECDSA •<S\_TRANSACTION\_ID> is the same as in IC3 | RQ31\_100 RQ31\_105 RQ31\_160 |
| 4 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#R\_SUCCESS) | If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT.  OR  If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT. | RQ31\_100 |

### 4.4.26 ES9+ (LPA – SM-DP+): HTTPS

#### 4.4.26.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ21\_001

 RQ26\_023, RQ26\_024, RQ26\_026, RQ26\_027, RQ26\_029

 RQ31\_032, RQ31\_032\_1

 RQ45\_026, RQ45\_031

 RQ56\_001, RQ56\_003

 RQ60\_001, RQ60\_002, RQ60\_004

 RQ61\_001

#### 4.4.26.2 Test Cases

##### 4.4.26.2.1 TC\_LPAd\_HTTPS\_Nominal

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| eUICC | There is no default SM-DP+ address configured. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |

Test Sequence #01 Nominal: HTTPS Session Establishment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DP+ | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>)  Verify the following: • #IUT\_TLS\_VERSION SHALL be 1.2 or higher • <TLS\_CIPHER\_SUITES> SHALL contain at least TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 or TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256  • <EXT\_SHA256\_ECDSA> SHALL have at least the 'supported\_signature\_algorithms' extension set with HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03). | RQ26\_023 RQ26\_024 RQ26\_026 RQ31\_032 RQ56\_001 |
| 2 | S\_SM-DP+ → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DP\_TLS) | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>) | RQ26\_027 RQ31\_032 RQ45\_026 RQ56\_003 |
| 3 | S\_SM-DP+ → LPAd | Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages) | HTTPS connection established | RQ31\_032 RQ56\_001 RQ60\_001 RQ60\_002 RQ61\_001 |

Test Sequence #02 Nominal: non-reuse of session keys

The purpose of this test sequence is to verify that the LPAd is not reusing ephemeral keys from the previous session.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+  Extract <CLIENT\_TLS\_EPHEM\_KEY>  Extract <SESSION\_ID\_CLIENT> and <S\_SESSION\_ID\_SERVER> | | | |
| IC2 | Terminate TLS session and restart “Add Profile” Procedure as define in the initial conditions. | | | |
| 1 | LPAd → S\_SM-DP+ | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLEINT>, <EXT\_SHA256\_ECDSA>)  Verify the following: • #IUT\_TLS\_VERSION SHALL be 1.2 or higher • <TLS\_CIPHER\_SUITES> SHALL be at least TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 • if <SESSION\_ID\_CLIENT> is non-empty then it SHALL be different from <SESSION\_ID\_CLIENT> and <S\_SESSION\_ID\_SERVER> extracted in IC1. • <EXT\_SHA256\_ECDSA> SHALL have at least the 'supported\_signature\_algorithms' extension set with HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03). | RQ31\_032 |
| 2 | S\_SM-DP+ → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DP\_TLS) | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>)  Verify if  • <CLIENT\_TLS\_EPHEM\_KEY> is different from the one used by LPAd in IC1 | RQ31\_032 |
| 3 | S\_SM-DP+ → LPAd | Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages) | HTTPS connection established | RQ31\_032 RQ60\_001 RQ60\_002 RQ60\_004 RQ61\_001 |

##### 4.4.26.2.2 TC\_LPAd\_HTTPS\_ErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |

Test Sequence #01 Error: Invalid (SM-DP+) TLS Certificate signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DP+ | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2,  #S\_TLS\_CIPHER\_SUITE,  <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DP\_TLS\_INV\_SIG)  Note: if the LPAd sends an Alert during or after any of the messages sent by the S\_SM-DP+ in MTD\_TLS\_SERVER\_HELLO\_ETC, then the S\_SM-DP+ might not send the messages specified in MTD\_TLS\_SERVER\_HELLO\_ETC which occur after the Alert. | LPAd MAY send a TLS Alert.  LPAd aborts AddProfile procedure | RQ31\_032 RQ45\_026 |
| 3 | LPDd → S\_SM-DP+ | TLS 1.2 close | The TLS connection is rejected. | RQ26\_023 RQ56\_003 |

Test Sequence #02 Error: Expired TLS Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DP+ | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2,  #S\_TLS\_CIPHER\_SUITE,  <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DP\_TLS\_EXPIRED)  Note: if the LPAd sends an Alert during or after any of the messages sent by the S\_SM-DP+ in MTD\_TLS\_SERVER\_HELLO\_ETC, then the S\_SM-DP+ might not send the messages specified in MTD\_TLS\_SERVER\_HELLO\_ETC which occur after the Alert. | LPAd MAY send a TLS Alert.  LPAd aborts AddProfile procedure | RQ31\_032 RQ45\_026 |
| 3 | LPDd → S\_SM-DP+ | TLS 1.2 close | The TLS connection is rejected. | RQ26\_023 RQ56\_003 |

Test Sequence #03 Error: VOIDTest Sequence #04 Error: VOID

Test Sequence #05 Error: VOID

Test Sequence #06 Error: VOID

Test Sequence #07 Error: Invalid TLS Certificate based on Invalid CI (Invalid Curve)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Power-on the Device | | | |
| 1 | LPAd → S\_SM-DP+ | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DP\_TLS\_INV\_CURVE)  Note: if the LPAd sends an Alert during or after any of the messages sent by the S\_SM-DP+ in MTD\_TLS\_SERVER\_HELLO\_ETC, then the S\_SM-DP+ might not send the messages specified in MTD\_TLS\_SERVER\_HELLO\_ETC which occur after the Alert. | LPAd MAY send a TLS Alert.  LPAd aborts AddProfile procedure | RQ31\_032 RQ45\_031 |
| 3 | LPDd → S\_SM-DP+ | TLS 1.2 close | The TLS connection is rejected. | RQ26\_029 RQ56\_003 |

### 4.4.27 ES11 (LPA – SM-DS): InitiateAuthentication

#### 4.4.27.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_033, RQ31\_034, RQ31\_035, RQ31\_036, RQ31\_043, RQ31\_045, RQ31\_048, RQ31\_052, RQ31\_075

 RQ58\_013, RQ58\_020

 RQ65\_026

#### 4.4.27.2 Test Cases

##### 4.4.27.2.1 TC\_LPAd\_ES11\_InitiateAuthentication\_Nominal

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The Profile Download is initiated using SM-DS (see section 2.2.4.1). |
| S\_SM-DS | S\_SM-DP+ (#TEST\_DP\_ADDRESS1) performed Profile download Event Registration to the S\_SM-DS (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1. |
| eUICC | There is no default SM-DP+ address configured. |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #EVENT\_ID\_1 (PROFILE\_OPERATIONAL1) (see NOTE). |
| NOTE: In order to avoid potentially misleading errors on LUI, the S\_SM-DP+ SHALL be available to the LPAd for profile download during test sequence execution. The test tool SHALL NOT check the ES9+ communication. | |

Test Sequence #01 Nominal: Initiate Authentication

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| 1 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS))  • Extract <EUICC\_CHALLENGE> | RQ31\_033 |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_DS\_OK) | No error: Next step of common mutual authentication procedure is performed. | RQ31\_043 RQ58\_013 RQ58\_020 RQ65\_026 |

##### 4.4.27.2.2 TC\_LPAd\_ES11\_InitiateAuthentication\_ErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The Profile Download is initiated using SM-DS (see section 2.2.4.1). |
| S\_SM-DS | S\_SM-DP+ (#TEST\_DP\_ADDRESS1) performed Profile download Event Registration to the S\_SM-DS (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1 (see NOTE). |
| eUICC | There is no default SM-DP+ address configured. |
| NOTE: The S\_SM\_DP+ does not need to be available to the LPAd for profile download during test sequence execution, as the LPAd is not expected to receive the smdpAddress. | |

Test Sequence #01 Error: Invalid SM-DS Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_9\_1\_3\_8) | LPAd aborts AddProfile procedure | RQ31\_034 RQ58\_020 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No ES11.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_034 RQ58\_020 |

Test Sequence #02 Error: Unsupported Security Configuration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_9\_2\_3\_1) | LPAd aborts AddProfile procedure | RQ31\_035 RQ58\_020 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No ES11.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_035 RQ58\_020 |

Test Sequence #03 Error: Unsupported SVN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_9\_3\_3\_1) | LPAd aborts AddProfile procedure | RQ58\_020 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No ES11.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ58\_020 |

Test Sequence #04 Error: Unavailable SM-DS Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_9\_4\_3\_7) | LPAd aborts AddProfile procedure | RQ31\_036 RQ58\_020 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No ES11.InitiateAuthentication requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_036 RQ58\_020 |

Test Sequence #05 Error: Invalid SM-DS Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_INV\_CERT\_DS) | LPAd aborts AddProfile procedure | RQ31\_052 RQ58\_013 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_052 RQ58\_013 |

Test Sequence #06 Error: Invalid SM-DS Signature

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | | |
| IC2 | LPAd → S\_SM-DS | | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 1 | S\_SM-DS → LPAd | | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_INV\_SIGN\_DS) | LPAd aborts AddProfile procedure | RQ31\_052 RQ58\_013 |
| 2 | LPAd → S\_SM-DS | | No Profile download action | No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_052 RQ58\_013 |

Test Sequence #07 Error: Invalid SM-DS Address sent by the SM-DS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_INV\_SMDS\_ADDRESS) | LPAd informs the S\_EndUser and aborts the AddProfile procedure | RQ31\_045 RQ31\_052 RQ58\_013 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ31\_045 RQ31\_052 RQ58\_013 |

Test Sequence #08 Error: Unsupported CI Key ID

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>, #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) | | |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_INV\_CI\_DS) | LPAd aborts AddProfile procedure | | | RQ31\_048 RQ31\_052 RQ58\_013 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | | | RQ31\_048 RQ31\_052 RQ58\_013 |

### 4.4.28 ES11 (LPA – SM-DS): AuthenticateClient

#### 4.4.28.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_046, RQ31\_056, RQ31\_057, RQ31\_061, RQ31\_062, RQ31\_065, RQ31\_078, RQ31\_083, RQ31\_085, RQ31\_090, RQ31\_095, RQ31\_136, RQ36\_018, RQ36\_019, RQ36\_020

 RQ42\_001, RQ42\_002, RQ42\_003, RQ42\_004, RQ42\_005, RQ42\_006, RQ42\_007, RQ42\_008, RQ42\_009, RQ42\_010, RQ42\_011, RQ42\_012, RQ42\_013, RQ42\_014, RQ42\_015, RQ42\_016, RQ42\_017, RQ42\_018, RQ42\_019, RQ42\_020

 RQ58\_021, RQ58\_030, RQ58\_036, RQ58\_037, RQ58\_038, RQ58\_039

 RQ62\_001, RQ62\_002, RQ62\_003, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007, RQ62\_008, RQ62\_009

 RQ63\_001\_1, RQ63\_004, RQ63\_005, RQ63\_006

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_004, RQ65\_005, RQ65\_006, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_022, RQ65\_028

#### 4.4.28.2 Test Cases

##### 4.4.28.2.1 TC\_LPAd\_ES11\_AuthenticateClient\_Nominal

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The Profile Download is initiated using SM-DS (see section 2.2.4.1). |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| eUICC | There is no default SM-DP+ address configured. |

Test Sequence #01 Nominal: Authenticate Client with empty MatchingID

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| S\_SM-DS | S\_SM-DP+ (#TEST\_DP\_ADDRESS1) performed Profile download Event Registration to the root S\_SM-DS (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1 for #EID1. |
| S\_SM-DP+ | There is a pending Profile download order for #EVENT\_ID\_1 (PROFILE\_OPERATIONAL1) (see NOTE). |
| NOTE: In order to avoid potentially misleading errors on LUI, the S\_SM-DP+ SHALL be available to the LPAd for profile download during test sequence execution. The test tool SHALL NOT check the ES9+ communication. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>,  #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS))  • Extract <EUICC\_CHALLENGE> |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_DS\_OK) | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO))  Verify:  • If <S\_TRANSACTION\_ID> is the same as in #INITIATE\_AUTH\_DS\_OK • <EUICC\_SIGNATURE1> using the #PK\_EUICC\_ECDSA  • if matchingId field in #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO is missing OR matchingId field is present and <MATCHING\_ID> is empty • if <S\_SMDS\_CHALLENGE> present in the #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO is the same as in <S\_SMDS\_SIGNED1> present in #INITIATE\_AUTH\_DS\_OK • for #DEVICE\_INFO: - The value of the TAC corresponds to the first 8 digits of #IUT\_IMEI and is represented as a string of 4 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] - if IMEI is present then its value corresponds to #IUT\_IMEI and is represented as a string of 8 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] except that the last octet contains the check digit (in high nibble) and an 'F' filler (in low nibble)  - if O\_D\_GSM\_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT\_GSM\_GERAN\_REL. – if O\_D\_UMTS\_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT\_UMTS\_UTRAN\_REL. – if O\_D\_CDMA2000\_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_1X\_REL. – if O\_D\_CDMA2000\_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_HRPD\_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O\_D\_CDMA2000\_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT\_CDMA2000\_EHRPD\_REL. – if O\_D\_LTE then eutranSupportedRelease (or eutranEpcSupportedRelease if #IUT\_RSP\_VERSION is v2.2.2 or higher) is set to the highest release as defined in #IUT\_LTE\_EUTRAN\_REL. – if O\_D\_NFC\_TS26 then contactlessSupportedRelease is set to the highest release as defined in #IUT\_NFC\_REL. – if O\_D\_CRL then rspCrlSupportedVersion is set to the highest release as defined in #IUT\_RSP\_VERSION .  For each of the options O\_D\_GSM\_GERAN, O\_D\_UMTS\_UTRAN, O\_D\_CDMA2000\_1X, O\_D\_CDMA2000\_HRPD, O\_D\_CDMA2000\_EHRPD, O\_D\_LTE, O\_D\_NFC\_TS26 or O\_D\_CRL, if the option is not set, verify that the corresponding field in DeviceCapabilities is not present. | RQ31\_046 RQ31\_056 RQ31\_057 RQ31\_078 RQ36\_018, RQ36\_019 RQ42\_001 RQ42\_002 RQ42\_003 RQ42\_004 RQ42\_005 RQ42\_006 RQ42\_007 RQ42\_008 RQ42\_009 RQ42\_010 RQ42\_011 RQ42\_012 RQ42\_013 RQ42\_014 RQ42\_015 RQ42\_016 RQ42\_017 RQ42\_018 RQ42\_019 RQ42\_020 RQ58\_021 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ62\_003 RQ62\_004 RQ62\_005 RQ62\_006 RQ62\_007 RQ62\_008 RQ62\_009 RQ63\_001\_1 RQ63\_004 RQ63\_005 RQ63\_006 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_004 RQ65\_005 RQ65\_006 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_022 RQ65\_028 |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_DS\_OK1) | No Error | RQ31\_062, RQ31\_065, RQ31\_095 |

Test Sequence #02 Nominal: Authenticate Client with MatchingID set to EventID

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| S\_SM-DS | The Alternative S\_SM-DS(2) (#TEST\_DS\_ADDRESS1) performed Profile download Event Registration to the root S\_SM-DS(1) (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1 for #EID1. |
| S\_SM-DS | S\_SM-DP+ (#TEST\_DP\_ADDRESS1) performed Profile download Event Registration to the Alternative S\_SM-DS(2) (#TEST\_DS\_ADDRESS1) with #EVENT\_ID\_2 for #EID1. |
| S\_SM-DP+ | There is a pending Profile download order for #EVENT\_ID\_2 (PROFILE\_OPERATIONAL1) (see NOTE). |
| NOTE: In order to avoid potentially misleading errors on LUI, the S\_SM-DP+ SHALL be available to the LPAd for profile download during test sequence execution. The test tool SHALL NOT check the ES9+ communication. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence/ Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS(1) | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(<EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS))  • Extract <EUICC\_CHALLENGE> |  |
| 1 | S\_SM-DS(1) → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_DS\_OK) | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT( <S\_TRANSACTION\_ID>, #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO))  Verify:• if matchingId field in #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO is missing OR matchingId field is present and <MATCHING\_ID> is empty | RQ31\_078 |
| 2 | S\_SM-DS(1) → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_DS\_OK\_DSADDR1) | No Error | RQ31\_062 RQ31\_065 RQ31\_095 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11  with #TEST\_DS\_ADDRESS1 and #CERT\_S\_SM\_DS2\_TLS | | | |
| 4 | LPAd → S\_SM-DS(2) | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DS\_ADDRESS1,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(<EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_DS\_ADDRESS1))  • Extract <EUICC\_CHALLENGE> |  |
| 5 | S\_SM-DS(2) → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_DS\_OK\_1) | MTD\_HTTP\_REQ(#TEST\_DS\_ADDRESS1 , #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>, #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO\_1))  Verify:  • if <MATCHING\_ID> is set to #EVENT\_ID\_1 | RQ31\_078 RQ36\_018 RQ36\_020 |
| 6 | S\_SM-DS(2) → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_DS\_OK2) | No Error | RQ31\_062 RQ31\_065 RQ31\_095 |

##### 4.4.28.2.2 TC\_LPAd\_ES11\_AuthenticateClient\_ErrorCases

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The Profile Download is initiated using SM-DS (see section 2.2.4.1). |
| S\_SM-DP+ | There is a pending Profile download order for #EVENT\_ID\_1 (PROFILE\_OPERATIONAL1) (see NOTE). |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| eUICC | There is no default SM-DP+ address configured. |
| NOTE: The S\_SM\_DP+ does not need to be available to the LPAd for profile download during test sequence execution, as the LPAd is not expected to receive the smdpAddress. | |

Test Sequence #01 Error: Invalid EUM Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | PROC\_ES11\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DS | Send ES11.AuthenticateClient method | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_1\_2\_6\_1) | LPAd aborts AddProfile procedure | RQ31\_061 |
| 3 | LPAd → S\_SM-DS | No Profile download action | No requests are sent on ES11 within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ58\_030 RQ58\_039 |

Test Sequence #02 Error: Expired EUM Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | PROC\_ES11\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DS | Send ES11.AuthenticateClient method | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_1\_2\_6\_3) | LPAd aborts AddProfile procedure | RQ31\_061 |
| 3 | LPAd → S\_SM-DS | No Profile download action | No requests are sent on ES11 within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ58\_030 RQ58\_039 |

Test Sequence #03 Error: Invalid eUICC Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | PROC\_ES11\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DS | Send ES11.AuthenticateClient method | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_1\_3\_6\_1) | LPAd aborts AddProfile procedure | RQ31\_061 |
| 3 | LPAd → S\_SM-DS | No Profile download action | No requests are sent on ES11 within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ58\_030 RQ58\_039 |

Test Sequence #04 Error: Expired eUICC Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | PROC\_ES11\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DS | Send ES11.AuthenticateClient method | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_1\_3\_6\_3) | LPAd aborts AddProfile procedure | RQ31\_061 |
| 3 | LPAd → S\_SM-DS | No Profile download action | No requests are sent on ES11 within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ58\_030 RQ58\_039 |

Test Sequence #05 Error: Invalid eUICC signature or serverChallenge

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | | |
| IC2 | PROC\_ES11\_INIT\_AUTH | | | | |
| 1 | LPAd → S\_SM-DS | Send ES11.AuthenticateClient method | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_1\_6\_1) | LPAd aborts AddProfile procedure | RQ58\_030 RQ58\_039 |
| 3 | LPAd → S\_SM-DS | No Profile download action | No requests are sent on ES11 within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ58\_030 RQ58\_039 |

Test Sequence #06 Error: Unknown TransactionID

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | PROC\_ES11\_INIT\_AUTH | | | |
| 1 | LPAd → S\_SM-DS | Send ES11.AuthenticateClient method | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_10\_1\_3\_9) | LPAd aborts AddProfile procedure | RQ56\_030 |
| 3 | LPAd → S\_SM-DS | No Profile download action | No requests are sent on ES11 within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ56\_030 RQ56\_041 |

Test Sequence #07 Error: Unknown Event Record

|  |
| --- |
| **Initial Conditions** |
| **Entity** | **Description of the initial condition** |
| S\_SM-DS | The Alternative S\_SM-DS (#TEST\_DS\_ADDRESS1) performed Profile download Event Registration to the root S\_SM-DS (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1 for #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | |
| IC2 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( <EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS))  • Extract <EUICC\_CHALLENGE> |  |
| IC3 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_DS\_OK) | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| IC4 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_DS\_OK\_DSADDR1) | No Error |  |
| IC5 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11  with #TEST\_DS\_ADDRESS1 and #CERT\_S\_SM\_DS2\_TLS | | | |
| IC6 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_DS\_ADDRESS1,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(<EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_DS\_ADDRESS1))  • Extract <EUICC\_CHALLENGE> |  |
| IC7 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#INITIATE\_AUTH\_DS\_OK\_1) | MTD\_HTTP\_REQ(#TEST\_DS\_ADDRESS1 , #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>, #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO\_1)) |  |
| 1 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP(#R\_ERROR\_8\_9\_5\_3\_9) | LPAd aborts AddProfile procedure | RQ31\_090 RQ31\_083 |
| 2 | LPAd → S\_SM-DS | No Profile download action | No requests are sent on ES11 within the timeout #IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT in Annex F. | RQ58\_035 |

### 4.4.29 ES11 (LPA -- SM-DS): HTTPS

#### 4.4.29.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_023, RQ26\_024, RQ26\_026, RQ26\_027, RQ26\_029

* RQ31\_032
* RQ36\_017

 RQ45\_026, RQ45\_028, RQ45\_033

 RQ58\_001, RQ58\_002

 RQ60\_001, RQ60\_002, RQ61\_001

#### 4.4.29.2 Test Cases

##### 4.4.29.2.1 TC\_LPAd\_ES11\_HTTPS\_Nominal

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The Profile Download is initiated using SM-DS (see section 2.2.4.1). |
| S\_SM-DS | S\_SM-DP+ (#TEST\_DP\_ADDRESS1) performed Profile download Event Registration to the S\_SM-DS (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1. |
| eUICC | There is no default SM-DP+ address configured. |

Test Sequence #01 Nominal: HTTPS Session Establishment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DS | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(#IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>)  Verify the following: • #IUT\_TLS\_VERSION SHALL be 1.2 or higher • <TLS\_CIPHER\_SUITES> SHALL contain at least TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256  • <EXT\_SHA256\_ECDSA> SHALL have at least the'supported\_signature\_algorithms' extension set with HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03). | RQ26\_023 RQ26\_024 RQ26\_026 RQ31\_032 RQ58\_001 |
| 2 | S\_SM-DS → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DS\_TLS) | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>) | RQ26\_027 RQ31\_032 RQ36\_017 RQ45\_026 RQ45\_028 RQ45\_033 RQ58\_002 |
| 3 | S\_SM-DS → LPAd | Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages) | HTTPS connection established | RQ31\_032 RQ58\_001 RQ60\_001 RQ60\_002 RQ61\_001 |

Test Sequence #02 Nominal: non-reuse of session keys

The purpose of this test sequence is to verify that the LPAd is not reusing ephemeral keys from the previous session.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11  Extract <CLIENT\_TLS\_EPHEM\_KEY>  Extract <SESSION\_ID\_CLIENT> and <S\_SESSION\_ID\_SERVER> | | | |
| IC2 | Terminate TLS session and Initiate Profile Download using SM-DS (see section 2.2.4.1). | | | |
| 1 | LPAd → S\_SM-DS | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>)  Verify the following: • #IUT\_TLS\_VERSION SHALL be 1.2 or higher • <TLS\_CIPHER\_SUITES> SHALL be at least TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 • if <SESSION\_ID\_CLIENT> is non-empty then it SHALL be different from <SESSION\_ID\_CLIENT> and <S\_SESSION\_ID\_SERVER> extracted in IC1. • <EXT\_SHA256\_ECDSA> SHALL have at least the'supported\_signature\_algorithms' extension set with HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03). | RQ31\_032 |
| 2 | S\_SM-DS → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DS\_TLS) | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>)  Verify if  • <CLIENT\_TLS\_EPHEM\_KEY> is different from the one used by LPAd in IC1 | RQ31\_032 |
| 3 | S\_SM-DS → LPAd | Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages) | HTTPS connection established | RQ31\_032  RQ58\_001RQ60\_001 RQ60\_002 RQ61\_001 |

##### 4.4.29.2.2 TC\_LPAd\_ES11\_HTTPS\_Error

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The Profile Download is initiated using SM-DS (see section 2.2.4.1). |
| S\_SM-DS | S\_SM-DP+ (#TEST\_DP\_ADDRESS1) performed Profile download Event Registration to the S\_SM-DS (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1. |
| eUICC | There is no default SM-DP+ address configured. |

Test Sequence #01 Error: Invalid (SM-DS) TLS Certificate signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DS | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>) |  |
| 2 | S\_SM-DS → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DS\_TLS\_INV\_SIG)  Note: if the LPAd sends an Alert during or after any of the messages sent by the S\_SM-DS in MTD\_TLS\_SERVER\_HELLO\_ETC, then the S\_SM-DS might not send the messages specified in MTD\_TLS\_SERVER\_HELLO\_ETC which occur after the Alert. | LPAd MAY send a TLS Alert.  LPAd aborts AddProfile procedure | RQ31\_032 RQ45\_026 RQ45\_028 |
| 3 | LPDd → S\_SM-DS | TLS 1.2 close | The TLS connection is rejected. | RQ26\_023 RQ58\_002 |

Test Sequence #02 Error: Expired TLS Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DS | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>) |  |
| 2 | S\_SM-DS → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DS\_TLS\_EXPIRED)  Note: if the LPAd sends an Alert during or after any of the messages sent by the S\_SM-DS in MTD\_TLS\_SERVER\_HELLO\_ETC, then the S\_SM-DS might not send the messages specified in MTD\_TLS\_SERVER\_HELLO\_ETC which occur after the Alert. | LPAd MAY send a TLS Alert.  LPAd aborts AddProfile procedure | RQ31\_032 RQ45\_026 |
| 3 | LPDd → S\_SM-DS | TLS 1.2 close | The TLS connection is rejected. | RQ26\_023 RQ58\_002 |

Test Sequence #03 Error: VOIDTest Sequence #04 Error: VOID

Test Sequence #05 Error: VOID

Test Sequence #06 Error: VOID

Test Sequence #07 Error: Invalid TLS Certificate based on Invalid CI (Invalid Curve)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DS | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>) |  |
| 2 | S\_SM-DS → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SM\_DS\_TLS\_INV\_CURVE)  Note: if the LPAd sends an Alert during or after any of the messages sent by the S\_SM-DS in MTD\_TLS\_SERVER\_HELLO\_ETC, then the S\_SM-DS might not send the messages specified in MTD\_TLS\_SERVER\_HELLO\_ETC which occur after the Alert. | LPAd MAY send a TLS Alert.  LPAd aborts AddProfile procedure | RQ26\_029 RQ31\_032 RQ45\_033 |
| 3 | LPDd → S\_SM-DS | TLS 1.2 close | The TLS connection is rejected. | RQ26\_023 RQ58\_002 |

## 4.5 SM-DS Interfaces

### 4.5.1 ES12 (SM-DP+ -- SM-DS): RegisterEvent

#### 4.5.1.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ36\_004, RQ36\_005, RQ36\_006, RQ36\_007, RQ36\_008, RQ36\_009, RQ36\_010, RQ36\_011, RQ36\_012, RQ36\_013

 RQ59\_003, RQ59\_004, RQ59\_005, RQ59\_006, RQ59\_007, RQ59\_009, RQ59\_010, RQ59\_011, RQ59\_012, RQ59\_013, RQ59\_014, RQ59\_015

 RQ62\_001, RQ62\_002, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_030

#### 4.5.1.2 Test Cases

##### 4.5.1.2.1 TC\_ROOT\_SM\_DS\_ES12.RegisterEvent

|  |  |
| --- | --- |
| General Initial Conditions | |
| **Entity** | Description of the general initial condition |
| Root SM-DS | * No TLS connections are established between the Root SM-DS and any of the simulator test tools. |

Test Sequence #01 Nominal: EventID Registration to SM-DS without Event forwarding

The purpose of this test is to verify that the SM-DS can perform Event Registration without Event forwarding set.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is not already used by the Root SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | | |
| 1 | S\_SM-DP+ →  Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12, #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  FALSE)) | MTD\_HTTP\_RESP(#R\_SUCCESS) | RQ36\_004 RQ36\_005 RQ59\_004 RQ59\_006 RQ59\_009 RQ59\_011 RQ59\_013 RQ59\_014 RQ62\_001 RQ62\_002 RQ62\_005 RQ62\_006 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_030 |
| 2 | S\_LPAd →  Root SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID | | RQ36\_004 RQ59\_006 |

Test Sequence #02 Nominal: EventID Registration to SM-DS with Event forwarding

The purpose of this test is to verify that the SM-DS ignores the ForwardingIndicator and successfully performs Event Registration with Event forwarding set.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is not already used by the Root SM-DS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | | |
| 1 | S\_SM-DP+ →  Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  TRUE)) | MTD\_HTTP\_RESP(#R\_SUCCESS) | RQ59\_003 RQ59\_012 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd →  Root SM-DS | PROC**\_**ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID | | RQ36\_004 RQ59\_006 |

Test Sequence #03 Error: Event Record Already Exists without Event Forwarding (Subject Code 8.9.5 Reason Code 3.3)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is already used by the Root SM-DS for #EID2, registered with S\_SM\_DP+\_OID.. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | | |
| 1 | S\_SM-DP+ →  Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  FALSE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_3) | RQ59\_005 RQ59\_010 RQ59\_015 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd →  Root SM-DS | PROC**\_**ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID\_ERROR | | RQ59\_005 |

##### 4.5.1.2.2 TC\_ALT\_SM\_DS\_ES12.RegisterEvent

The test sequences in this section test the Alternative SM-DS acting as a Server on ES12 and a Client on ES15.

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Alt. SM-DS | * No TLS connections are established between the Alternative SM-DS and any of the simulator test tools. |

Test Sequence #01 Nominal: EventID Registration on Alternative SM-DS with Event forwarding

The purpose of this test is to verify that Alternative SM-DS can perform Event Registration with Event forwarding set.

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| Alt. SM-DS | #EVENT\_ID\_1 is not already used by the Alternative SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  TRUE)) |  |  |
| 2 | Alt. SM-DS → S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | |
| 3 | Alt. SM-DS → S\_SM-DS | Call ES15.RegisterEvent | MTD\_HTTP\_REQ(  #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  <FUNCTION\_REQ\_ID>,  <FUNCTION\_CALL\_ID>,  #EID1,   #IUT\_SM\_DS\_ADDRESS\_ES11,  <EVENT\_ID\_R>,  FALSE)) | RQ36\_007 RQ36\_008 RQ36\_009 RQ36\_010 RQ36\_011 RQ36\_012 RQ36\_013RQ59\_002 RQ59\_004 RQ59\_006 RQ59\_011 RQ62\_001 RQ62\_002 RQ62\_004 RQ62\_006 RQ62\_007RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_030 |
| 4 | S\_SM-DS → Alt. SM-DS | MTD\_HTTP\_RESP(#R\_SUCCESS) on ES15 | No Error |  |
| 5 | Alt. SM-DS → S\_SM-DP+ | Successful result is sent to the S\_SM-DP+ | MTD\_HTTP\_RESP(#R\_SUCCESS) on ES12 | RQ36\_007 RQ36\_008 RQ36\_009 RQ36\_010 RQ36\_011 RQ36\_012 RQ36\_013RQ59\_009 RQ59\_013 RQ59\_014RQ62\_001 RQ62\_002 RQ62\_005 RQ62\_006RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_030 |
| 6 | S\_LPAd → Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID | | RQ36\_009 RQ59\_006 |

Test Sequence #02 Nominal: Uniqueness of EventID Registration by Alternative SM-DS with Event forwarding

The purpose of this test is to verify that Alternative SM-DS can perform Event Registration using a unique EventID2 value with Event forwarding set.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 is not already used by the Alternative SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  TRUE)) |  |  |
| 2 | Alt. SM-DS → S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | |
| 3 | Alt. SM-DS → S\_SM-DS | Call ES15.RegisterEvent | MTD\_HTTP\_REQ(  #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  <FUNCTION\_REQ\_ID>,  <FUNCTION\_CALL\_ID>,  #EID1,   #IUT\_SM\_DS\_ADDRESS\_ES11,  <EVENT\_ID\_R>,  FALSE))  Extract the value of <EVENT\_ID\_R> | RQ36\_006 RQ62\_001 RQ62\_002 |
| 4 | S\_SM-DS → Alt. SM-DS | MTD\_HTTP\_RESP(#R\_SUCCESS) on ES15 | No Error |  |
| 5 | Alt. SM-DS → S\_SM-DP+ | Successful result is sent to the S\_SM-DP+ | MTD\_HTTP\_RESP(#R\_SUCCESS) on ES12 | RQ36\_006 RQ62\_001 RQ62\_002 |
| 6 | S\_SM-DP+ → Alt. SM-DS | Close TLS session on ES12 (unless Alternative SM-DS has already closed TLS session) | |  |
| 7 | S\_SM-DP+ →  Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | |  |
| 8 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_2,  TRUE)) |  |  |
| 9 | Alt. SM-DS → S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | |
| 10 | Alt. SM-DS → S\_SM-DS | Call ES15.RegisterEvent | MTD\_HTTP\_REQ(  #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  <FUNCTION\_REQ\_ID>,  <FUNCTION\_CALL\_ID>,  #EID1,   #IUT\_SM\_DS\_ADDRESS\_ES11,  <EVENT\_ID\_R>,  FALSE))  Verify that <EVENT\_ID\_R> in step 3 is not equal to <EVENT\_ID\_R> | RQ36\_006 RQ62\_001 RQ62\_002 |

Test Sequence #03 Error: SM-DS registration failed, Root SM-DS unavailable (Subject Code 8.9 Reason Code 5.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 is not already used by the Alternative SM-DS. * S\_SM\_DS (Root SM-DS simulator) is not available – it will not respond to any client attempts to connect. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_1) | RQ59\_005 RQ59\_007 RQ59\_010 RQ59\_015 RQ62\_001 RQ62\_002 | |
| 2 | S\_LPAd →  Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID\_ERROR | | | RQ59\_005 |

Test Sequence #04 Error: SM-DS registration failed, Root SM-DS error (Subject Code 8.9 Reason Code 4.2)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 is not already used by the Alternative SM-DS for #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  TRUE)) |  |  |
| 2 | Alt. SM-DS → S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | |
| 3 | Alt. SM-DS → S\_SM-DS | Call ES15.RegisterEvent | MTD\_HTTP\_REQ(  #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  <FUNCTION\_REQ\_ID>,  <FUNCTION\_CALL\_ID>,  #EID1,   #IUT\_SM\_DS\_ADDRESS\_ES11,  <EVENT\_ID\_R>,  FALSE)) | RQ36\_007 RQ36\_008 RQ36\_009 RQ36\_010 RQ36\_011 RQ36\_012 RQ36\_013 RQ59\_002 RQ59\_004 RQ59\_006 RQ59\_011 RQ62\_001 RQ62\_002 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_030 |
| 4 | S\_SM-DS → Alt. SM-DS | MTD\_HTTP\_RESP( #R\_ERROR\_1\_2\_4\_2) | No Error |  |
| 5 | Alt. SM-DS → S\_SM-DP+ | SM-DS forwards error response back to S\_SM-DP+ | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_4\_2) | RQ59\_005 RQ59\_007 RQ59\_010 RQ59\_015 RQ62\_001 RQ62\_002 |
| 6 | S\_LPAd → Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID\_ERROR | | RQ59\_005 |

Test Sequence #05 Error: Event Record Already Exists on Alternative SM-DS (Subject Code 8.9.5 Reason Code 3.3)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 is already used by the Alternative SM-DS, registered with S\_SM\_DP+\_OID for #EID2. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_DP\_ADDRESS1,  #EVENT\_ID\_1,  TRUE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_3) | RQ59\_005 RQ59\_007 RQ59\_010 RQ59\_015 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd →  Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID\_ERROR | | RQ59\_005 |

### 4.5.2 ES12 (SM-DS -- SM-DP+): DeleteEvent

#### 4.5.2.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ36\_024, RQ36\_025, RQ36\_025\_1, RQ36\_027, RQ36\_028, RQ36\_029, RQ36\_030, RQ36\_031, RQ36\_032

* RQ510\_019, RQ510\_020

 RQ59\_016, RQ59\_016\_1, RQ59\_017, RQ59\_017\_1, RQ59\_017\_2, RQ59\_018, RQ59\_019, RQ59\_021, RQ59\_022, RQ59\_023, RQ59\_024, RQ59\_025

 RQ62\_001, RQ62\_002, RQ62\_004, RQ62\_005, RQ62\_006, RQ62\_007

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_031

#### 4.5.2.2 Test Cases

##### 4.5.2.2.1 TC\_ROOT\_SM\_DS\_ES12.DeleteEvent

Test Sequence #01 Nominal: Event Deletion

The purpose of this test is to verify that the Root SM-DS can perform Event Deletion.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 was registered for #EID1 and #TEST\_DP\_ADDRESS1, registered with S\_SM\_DP+\_OID |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | | |
| 1 | S\_SM-DP+ → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP(#R\_SUCCESS) | RQ36\_024 RQ36\_025 RQ36\_025\_1 RQ36\_029 RQ36\_030RQ59\_016 RQ59\_021 RQ59\_023 RQ59\_024RQ510\_019RQ62\_001 RQ62\_002 RQ62\_005 RQ62\_006RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_031 |
| 2 | S\_LPAd → Root SM-DS | PROC**\_**ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID\_ERROR | | RQ36\_025 RQ36\_029 RQ59\_017\_1 |

Test Sequence #02 Error: Event Record Does Not Exist (Subject Code 8.9.5 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is not registered. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | | |
| 1 | S\_SM-DP+ → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | RQ59\_016\_1 RQ59\_022 RQ59\_025 RQ510\_020 RQ62\_001 RQ62\_002 |

Test Sequence #03 Error: Event Record Does Not Match OID (Subject Code 8.9.5 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 was registered for #EID1 and #TEST\_DP\_ADDRESS1, registered with S\_SM\_DP+\_OID. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Root SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH\_INV\_OID on ES12 | | |
| 1 | S\_SM-DP+ → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | RQ59\_016\_1 RQ59\_022 RQ59\_025 RQ510\_020 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd → Root SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID | | RQ59\_016\_1 |

##### 4.5.2.2.2 TC\_ALT\_SM\_DS\_ES12.DeleteEvent

The test sequences in this section test the Alternative SM-DS acting as a Server on ES12 and a Client on ES15.

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Alt. SM-DS | * No TLS connections are established between the Alternative SM-DS and any of the simulator test tools. |

Test Sequence #01 Nominal: Cascaded Event Deletion on Alternative SM-DS

The purpose of this test is to verify that Alternative SM-DS can perform cascaded Event Deletion.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 registration for #EID1 and #TEST\_DP\_ADDRESS1 (registered with S\_SM\_DP+\_OID) was cascaded using <EVENT\_ID\_R> to the Root SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) |  |  |
| 2 | Alt. SM-DS → S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | |
| 3 | Alt. SM-DS → S\_SM-DS | Call ES15.DeleteEvent | MTD\_HTTP\_REQ(  #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  <FUNCTION\_REQ\_ID>,  <FUNCTION\_CALL\_ID>,  #EID1,  <EVENT\_ID\_D>))  Verify that <EVENT\_ID\_D> is equal to <EVENT\_ID\_R> | RQ36\_027 RQ36\_028 RQ36\_031 RQ36\_032 RQ59\_016 RQ59\_017 RQ59\_017\_2 RQ59\_023 RQ62\_001 RQ62\_002 RQ62\_004 RQ62\_006 RQ62\_007RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_031 |
| 4 | S\_SM-DS → Alt. SM-DS | MTD\_HTTP\_RESP( #R\_SUCCESS) on ES15 | No Error | RQ510\_019 |
| 5 | Alt. SM-DS → S\_SM-DP+ | SM-DS sends response back to S\_SM-DP+ | MTD\_HTTP\_RESP( #R\_SUCCESS) on ES12 | RQ36\_027 RQ36\_028 RQ36\_031 RQ36\_032 RQ59\_016 RQ59\_021 RQ59\_024RQ510\_019 RQ62\_001 RQ62\_002 RQ62\_005 RQ62\_006RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_031 |
| 6 | S\_LPAd → Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID\_ERROR | | RQ36\_031 RQ59\_019 RQ510\_020 RQ62\_001 RQ62\_002 |

Test Sequence #02 Nominal: Cascaded Event Deletion, Event Record not found on Root SM-DS

The purpose of this test is to verify that if cascaded deletion fails because the Event Record was not found in the Root SM-DS the Alternative SM-DS can ignore this error case and continue.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 registration for #EID1 and #TEST\_DP\_ADDRESS1 (registered with S\_SM\_DP+\_OID) was cascaded using <EVENT\_ID\_R> to the Root SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) |  |  |
| 2 | Alt. SM-DS →  S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | |
| 3 | Alt. SM-DS → S\_SM-DS | Call ES15.DeleteEvent | MTD\_HTTP\_REQ(  #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  <FUNCTION\_REQ\_ID>,  <FUNCTION\_CALL\_ID>,  #EID1,  <EVENT\_ID\_D>))  Verify that <EVENT\_ID\_D> is equal to <EVENT\_ID\_R> | RQ36\_027 RQ36\_028 RQ36\_031 RQ36\_032 RQ59\_016 RQ59\_017 RQ59\_017\_2 RQ59\_023 RQ62\_001 RQ62\_002 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_031 |
| 4 | S\_SM-DS → Alt. SM-DS | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | No Error | RQ510\_020 RQ62\_001 RQ62\_002 |
| 5 | Alt. SM-DS → S\_SM-DP+ | SM-DS sends response back to S\_SM-DP+ | MTD\_HTTP\_RESP( #R\_SUCCESS) on ES12 | RQ59\_021 RQ59\_024 RQ510\_019 RQ62\_001 RQ62\_002 |
| 6 | S\_LPAd → Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID\_ERROR | | RQ36\_031 RQ59\_018 RQ510\_020 RQ62\_001 RQ62\_002 |

Test Sequence #03 Error: Cascaded Event Deletion failed, Root SM-DS Unavailable (Subject Code 8.9 Reason Code 5.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 registration for #EID1 and #TEST\_DP\_ADDRESS1 (registered with S\_SM\_DP+\_OID) was cascaded using <EVENT\_ID\_R> to the Root SM-DS. * S\_SM\_DS (Root SM-DS simulator) is not available – it will not respond to any client attempts to connect. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_1) | RQ59\_016\_1 RQ59\_018 RQ59\_022 RQ59\_025 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd →  Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID | | RQ59\_018 |

Test Sequence #04 Error: Cascaded Event Deletion failed, Root SM-DS execution error (Subject Code 8.9 Reason Code 4.2)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 registration for #EID1 and #TEST\_DP\_ADDRESS1 (registered with S\_SM\_DP+\_OID) was cascaded using <EVENT\_ID\_R> to the Root SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) |  |  |
| 2 | Alt. SM-DS → S\_SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | |
| 3 | Alt. SM-DS → S\_SM-DS | Call ES15.DeleteEvent | MTD\_HTTP\_REQ(  #TEST\_ROOT\_DS\_ADDRESS,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  <FUNCTION\_REQ\_ID>,  <FUNCTION\_CALL\_ID>,  #EID1,  <EVENT\_ID\_D>)) | RQ36\_027 RQ36\_028 RQ36\_031 RQ36\_032 RQ59\_016 RQ59\_017 RQ59\_017\_2 RQ59\_023 RQ62\_001 RQ62\_002 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_031 |
| 4 | S\_SM-DS →  Alt. SM-DS | MTD\_HTTP\_RESP( #R\_ERROR\_1\_2\_4\_2) | No Error | RQ510\_020 RQ62\_001 RQ62\_002 |
| 5 | Alt. SM-DS → S\_SM-DP+ | SM-DS sends response back to S\_SM-DP+ | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_4\_2) | RQ59\_018 RQ59\_022 RQ59\_025 RQ510\_020 RQ62\_001 RQ62\_002 |
| 6 | S\_LPAd → Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID | | RQ59\_018 |

***Test Sequence #05 Error: Event Record Does Not Match OID (Subject Code 8.9.5 Reason Code 3.***9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 registration for #EID1 and #TEST\_DP\_ADDRESS1 (registered with S\_SM\_DP+\_OID) was cascaded using <EVENT\_ID\_R> to the Root SM-DS. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH\_INV\_OID on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | RQ59\_016\_1 RQ59\_022 RQ59\_025 RQ510\_020 RQ62\_001 RQ62\_002 | |
| 2 | S\_LPAd → Alt. SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID | | RQ59\_016\_1 |

##### 4.5.2.2.3 TC\_ALT\_SM\_DS\_ES12.DeleteEvent\_Error\_Nonexistant\_EventID

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Alt. SM-DS | * No TLS connections are established between the Alternative SM-DS and any of the simulator test tools. |

Test Sequence #01 Error: Event Record Does Not Exist (Subject Code 8.9.5 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Alt. SM-DS | * #EVENT\_ID\_1 is not registered. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DP+ → Alt. SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES12 | | |
| 1 | S\_SM-DP+ → Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DP+\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | RQ59\_016\_1 RQ59\_022 RQ59\_025 RQ510\_020 RQ62\_001 RQ62\_002 |

### 4.5.3 ES15 (SM-DS -- SM-DS): RegisterEvent

#### 4.5.3.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ36\_005, RQ36\_010, RQ36\_011, RQ36\_012

 RQ62\_001, RQ62\_002, RQ62\_005, RQ62\_006

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_030

 RQ510\_003, RQ510\_004, RQ510\_005, RQ510\_006, RQ510\_009, RQ510\_010, RQ510\_011, RQ510\_012, RQ510\_013, RQ510\_014, RQ510\_015

#### 4.5.3.2 Test Cases

##### 4.5.3.2.1 TC\_ROOT\_SM\_DS\_ES15.RegisterEvent

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Root SM-DS | * No TLS connections are established between the Root SM-DS and any of the simulator test tools. |

Test Sequence #01 Nominal: EventID Registration to SM-DS with Event forwarding

The purpose of this test is to verify that the Root SM-DS ignores the ForwardingIndicator and successfully performs Event Registration with Event forwarding set.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is not already used by the Root SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | | |
| 1 | S\_SM-DS → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES15,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DS\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_ALT\_DS\_ADDRESS,  #EVENT\_ID\_1,  TRUE)) | MTD\_HTTP\_RESP(#R\_SUCCESS) | RQ510\_003 RQ510\_012 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd → Root SM-DS | PROC**\_**ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID using R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_ALT\_DS\_OK instead of R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK | | RQ36\_011 |

Test Sequence #02 Nominal: EventID Registration to SM-DS without Event forwarding

The purpose of this test is to verify that the Root SM-DS successfully performs Event Registration with Event without Event forwarding set.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is not already used by the Root SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | | |
| 1 | S\_SM-DS → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES15,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DS\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_ALT\_DS\_ADDRESS,  #EVENT\_ID\_1,  FALSE)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ36\_010 RQ36\_011 RQ36\_012 RQ510\_004 RQ510\_006 RQ510\_009 RQ510\_011 RQ510\_013 RQ510\_014 RQ62\_001 RQ62\_002 RQ62\_005 RQ62\_006 RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_030 |
| 2 | S\_LPAd → Root SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID using R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_ALT\_DS\_OK instead of R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK | | RQ36\_011 |

Test Sequence #03 Error: Event Record Already Exists without Event Forwarding (Subject Code 8.9.5 Reason Code 3.3)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is already used by the Root SM-DS for #EID2, registered with S\_SM\_DS\_OID. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | | |
| 1 | S\_SM-DS → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES15,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #S\_SM\_DS\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,   #TEST\_ALT\_DS\_ADDRESS,  #EVENT\_ID\_1,  FALSE)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_3) | RQ510\_005 RQ510\_010 RQ510\_015 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd → Root SM-DS | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID\_ERROR | | RQ36\_005 |

### 4.5.4 ES15 (SM-DS -- SM-DS): DeleteEvent

#### 4.5.4.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ36\_028, RQ36\_029, RQ36\_030, RQ36\_031

 RQ62\_001, RQ62\_002, RQ62\_005, RQ62\_006

 RQ65\_001, RQ65\_002, RQ65\_003, RQ65\_005, RQ65\_007, RQ65\_008, RQ65\_009, RQ65\_031

 RQ510\_016, RQ510\_016\_1, RQ510\_021, RQ510\_022, RQ510\_023, RQ510\_024, RQ510\_025

#### 4.5.4.2 Test Cases

##### 4.5.4.2.1 TC\_ROOT\_SM\_DS\_ES15.DeleteEvent

|  |  |
| --- | --- |
| General Initial Conditions | |
| **Entity** | Description of the general initial condition |
| Root SM-DS | * No TLS connections are established between the Alternative SM-DS and any of the simulator test tools. |

Test Sequence #01 Nominal: Event Deletion

The purpose of this test is to verify that the Root SM-DS can perform Event Deletion.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 was registered for #EID1 and #TEST\_ALT\_DS\_ADDRESS, registered with S\_SM\_DS\_OID. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | | |
| 1 | S\_SM-DS → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES15,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DS\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_SUCCESS) | RQ36\_028 RQ36\_029 RQ36\_030 RQ62\_001 RQ62\_002 RQ62\_005 RQ62\_006RQ65\_001 RQ65\_002 RQ65\_003 RQ65\_005 RQ65\_007 RQ65\_008 RQ65\_009 RQ65\_031 RQ510\_016 RQ510\_021 RQ510\_023 RQ510\_024 |
| 2 | S\_LPAd → Root SM-DS | PROC**\_**ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID\_ERROR | | RQ36\_031 |

Test Sequence #02 Error: Event Record Does Not Exist (Subject Code 8.9.5 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | * #EVENT\_ID\_1 is not registered. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH on ES15 | | | |
| 1 | S\_SM-DS → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES15,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DS\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | RQ510\_016\_1 RQ510\_022 RQ510\_025 RQ62\_001 RQ62\_002 |

Test Sequence #03 Error: Event Record Does Not Match OID (Subject Code 8.9.5 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| Root SM-DS | #EVENT\_ID\_1 was registered for #EID1 and #TEST\_ALT\_DS\_ADDRESS, registered with S\_SM\_DS\_OID. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_SM-DS → Root SM-DS | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH\_INV\_OID on ES15 | | |
| 1 | S\_SM-DS → Root SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES15,  #PATH\_DELETE\_EVENT,  MTD\_DELETE\_EVENT(  #S\_SM\_DS\_F\_REQ\_ID,  #FUNCTION\_CALL\_ID\_1,  #EID1,  #EVENT\_ID\_1)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | RQ510\_016\_1 RQ510\_022 RQ510\_025 RQ62\_001 RQ62\_002 |
| 2 | S\_LPAd → Root SM-DS | PROC**\_**ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID using #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_ALT\_DS\_OK instead of #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK | | RQ510\_016\_1 |

### 4.5.5 ES11 (LPA -- SM-DS): InitiateAuthentication

#### 4.5.5.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_033

* RQ31\_030, RQ31\_033, RQ31\_034, RQ31\_035, RQ31\_036, RQ31\_037, RQ31\_038, RQ31\_039, RQ31\_041, RQ31\_042, RQ31\_043, RQ31\_073

 RQ57\_106

 RQ58\_003, RQ58\_004, RQ58\_005, RQ58\_006, RQ58\_007, RQ58\_008, RQ58\_010, RQ58\_011, RQ58\_012, RQ58\_013, RQ58\_014, RQ58\_015, RQ58\_016, RQ58\_017, RQ58\_018, RQ58\_019, RQ58\_020

 RQ62\_001, RQ62\_002

* RQ65\_018

#### 4.5.5.2 Test Cases

##### 4.5.5.2.1 TC\_SM\_DS\_ES11.InitiateAuthenticationNIST

|  |  |
| --- | --- |
| General Initial Conditions for SM-DS testing | |
| Entity | Description of the general initial condition |
| SM-DS | SM-DS is configured with the #CERT\_SM\_DSauth\_ECDSA for NIST. |

Perform all test sequences defined in 4.3.12.2.1 with the following variables:

 Test Environment = TE\_S1

 SERVER = SM-DS

o CERT\_SM\_XXauth\_ECDSA = CERT\_SM\_DSauth\_ECDSA

o PK\_SM\_XXauth\_ECDSA = PK\_SM\_DSauth\_ECDSA

##### 4.5.5.2.2 TC\_SM\_DS\_ES11.InitiateAuthenticationBRP

|  |  |
| --- | --- |
| General Initial Conditions for SM-DS testing | |
| Entity | Description of the general initial condition |
| SM-DS | SM-DS is configured with the #CERT\_SM\_DSauth\_ECDSA for BRP. |

Perform all test sequences defined in 4.3.12.2.3 with the following variables:

 Test Environment = TE\_S1

 SERVER = SM-DS

o CERT\_SM\_XXauth\_ECDSA = CERT\_SM\_DSauth\_ECDSA

o PK\_SM\_XXauth\_ECDSA = PK\_SM\_DSauth\_ECDSA

### 4.5.6 ES11 (LPA -- SM-DS): Authenticate Client

#### 4.5.6.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_005, RQ26\_006, RQ26\_012, RQ26\_014

 RQ31\_058, RQ31\_059, RQ31\_060

 RQ36\_017, RQ36\_021, RQ36\_022

 RQ45\_006, RQ45\_026, RQ45\_026\_1, RQ45\_027, RQ45\_028, RQ45\_029

 RQ57\_037, RQ57\_108

 RQ58\_025, RQ58\_026, RQ58\_027, RQ58\_028, RQ58\_029, RQ58\_031, RQ58\_036, RQ58\_036\_1, RQ58\_037, RQ58\_038, RQ58\_039

 RQ62\_001, RQ62\_002

* RQ65\_27, RQ65\_028, RQ65\_029

#### 4.5.6.2 Test Cases

##### 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DS | SM-DS is configured with the #CERT\_SM\_DSauth\_ECDSA for NIST. |

Test Sequence #01 Nominal Matching ID Empty for one pending Event

The purpose of this test is to verify that common mutual authentication between the SM-DS and the S\_LPAd is performed successfully with an empty Matching ID, and that Event Retrieval occurs for one pending Event.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EMPTY)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ36\_021 RQ36\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_025 RQ58\_026 RQ58\_027 RQ58\_028 RQ58\_029 RQ58\_031 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #02 Nominal Matching ID Empty for two pending Events

The purpose of this test is to verify that common mutual authentication between the SM-DS and the S\_LPAd is performed successfully with an empty Matching ID, and that Event Retrieval occurs for any pending Events.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. * #EVENT\_ID\_2 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS2. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EMPTY)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_MULTI\_OK) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ36\_021 RQ36\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_025 RQ58\_026 RQ58\_027 RQ58\_028 RQ58\_029 RQ58\_031 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #03 Nominal Matching ID Empty for no pending Events

The purpose of this test is to verify that common mutual authentication between the SM-DS and the S\_LPAd is performed successfully with an empty Matching ID, and that Event Retrieval returns no pending Events.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * No Events have been registered in the SM-DS for #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EMPTY)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_EMPTY\_OK) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ36\_021 RQ36\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_025 RQ58\_026 RQ58\_027 RQ58\_028 RQ58\_029 RQ58\_031 RQ58\_033 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #04 Nominal Matching ID Omitted for one pending Event

The purpose of this test is to verify that common mutual authentication between the SM-DS and the S\_LPAd is performed successfully with the Matching ID omitted, and that Event Retrieval occurs for one pending Event.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_OMITTED)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ36\_021 RQ36\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_025 RQ58\_026 RQ58\_027 RQ58\_028 RQ58\_029 RQ58\_031 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #05 Nominal Matching ID Omitted for two pending Events

The purpose of this test is to verify that common mutual authentication between the SM-DS and the S\_LPAd is performed successfully with the Matching ID omitted, and that Event Retrieval occurs for any pending Events.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. * #EVENT\_ID\_2 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS2. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_OMITTED)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_MULTI\_OK) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ36\_021 RQ36\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_025 RQ58\_026 RQ58\_027 RQ58\_028 RQ58\_029 RQ58\_031 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #06 Nominal Matching ID Omitted for no pending Events

The purpose of this test is to verify that common mutual authentication between the SM-DS and the S\_LPAd is performed successfully with the Matching ID omitted, and that Event Retrieval returns no pending Events.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * No Events have been registered in the SM-DS for #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_OMITTED)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_EMPTY\_OK) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ36\_021 RQ36\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_025 RQ58\_026 RQ58\_027 RQ58\_028 RQ58\_029 RQ58\_031 RQ58\_033 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #07 Alt. Nominal Matching ID containing EventID with one pending Event

The purpose of this test is to verify that common mutual authentication between the Alternative SM-DS and the S\_LPAd is performed successfully with a Matching ID containing an EventID, and that Event Retrieval occurs for the requested pending Event.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the Alternative SM-DS with #EID1 and #TEST\_DP\_ADDRESS1 and was cascaded using <EVENT\_ID\_R> to the Root SM-DS. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  Alt. SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EVENT\_ID\_R)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ36\_021 RQ36\_022 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_025 RQ58\_026 RQ58\_027 RQ58\_028 RQ58\_029 RQ58\_034 RQ58\_036 RQ58\_037 RQ58\_038 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #08 Alt Nominal Matching ID containing EventID with two pending Events

The purpose of this test is to verify that common mutual authentication between the Alternative SM-DS and the S\_LPAd is performed successfully with a Matching ID containing an EventID, and that Event Retrieval occurs for only the requested pending Event.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the Alternative SM-DS with #EID1 and #TEST\_DP\_ADDRESS1 and was cascaded using <EVENT\_ID\_R> to the Root SM-DS. * #EVENT\_ID\_2 has been registered in the Alternative SM-DS with #EID1 and #TEST\_DP\_ADDRESS2 and was cascaded, using an EventID which is different from <EVENT\_ID\_R>, to the Root SM-DS. |

Repeat Test Sequence #07 Nominal Matching ID containing one Event with one pending Event.

Test Sequence #09 Error: Invalid EUM Certificate (Subject Code 8.1.2 Reason Code 6.1)

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_SIG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |
| 2 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | |  |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 4 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 5 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_KU)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ65\_028 RQ62\_001 RQ62\_002 RQ65\_029 |
| 6 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | |  |
| 7 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 8 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 9 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_CP)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |
| 10 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | |  |
| 11 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 12 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 13 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_BC\_cA)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |
| 14 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | | |
| 15 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 16 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 17 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_BC\_PLC)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ65\_028 RQ62\_001 RQ62\_002 RQ65\_029 |

Test Sequence #10 Error: Expired EUM Certificate (Subject Code 8.1.2 Reason Code 6.3)

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_3)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_2\_6\_3) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #11 Error: Invalid eUICC Certificate (Subject Code 8.1.3 Reason Code 6.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_SIG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ65\_028 RQ62\_001 RQ62\_002 RQ65\_029 | |
| 2 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | |  | |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | | |
| 4 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 5 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,   #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_EX\_KU)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 | |
| 6 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | |  | |
| 7 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | | |
| 8 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 9 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_EX\_CP)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 | |
| 10 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | |  | |
| 11 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 12 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 13 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_SUB\_ORG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 | |
| 14 | S\_LPAd → SM-DS | Close TLS session (unless SM-DS has already closed TLS session) | |  | |
| 15 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | |  | |
| 16 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 17 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_SUB\_SN)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 | |

Test Sequence #12 Error: Expired eUICC Certificate (Subject Code 8.1.3 Reason Code 6.3)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_3)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_3\_6\_3) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #13 Error: Invalid eUICC Signature (Subject Code 8.1 Reason Code 6.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_6\_1\_SIG)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #14 Error: Invalid Server Challenge (Subject Code 8.1 Reason Code 6.1)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_1\_6\_1\_CHA)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_6\_1) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #15 Error: Unknown Transaction ID in JSON transport layer (Subject Code 8.10.1 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT( <INVALID\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EMPTY)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #16 Error: Unknown Transaction ID in ASN.1 euiccSigned1 payload (Subject Code 8.10.1 Reason Code 3.9)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * #EVENT\_ID\_1 has been registered in the SM-DS with #EID1 and #TEST\_DP\_ADDRESS1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT( <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_8\_10\_1\_3\_9)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_10\_1\_3\_9) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

Test Sequence #17 Error: Matching ID containing EventID with no pending Event

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DS | * No Events have been registered in the SM-DS for #EID1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| IC2 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd →  SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EVENT\_ID)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) | RQ26\_005 RQ26\_006 RQ26\_012 RQ26\_014 RQ31\_058 RQ31\_059 RQ31\_060 RQ36\_017 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ45\_027 RQ45\_028 RQ45\_029 RQ57\_037 RQ57\_108 RQ58\_030 RQ58\_036\_1 RQ58\_037 RQ58\_039 RQ62\_001 RQ62\_002 RQ65\_028 RQ65\_029 |

##### 4.5.6.2.2 TC\_SM-DS\_ES11.AuthenticateClientBRP

|  |  |
| --- | --- |
| General Initial Conditions | |
| **Entity** | Description of the general initial condition |
| SM-DS | SM-DS is configured with the #CERT\_SM\_DSauth\_ECDSA for BrainpoolP256r1. |

Test Sequence #01 Nominal Matching ID Empty for one pending Event

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #02 Nominal Matching ID Empty for two pending Events

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #03 Nominal Matching ID Empty for no pending Events

This test sequence SHALL be the same as the Test Sequence #03 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #04 Nominal Matching ID Omitted for one pending Event

This test sequence SHALL be the same as the Test Sequence #04 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #05 Nominal Matching ID Omitted for two pending Events

This test sequence SHALL be the same as the Test Sequence #05 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #06 Nominal Matching ID Omitted for no pending Events

This test sequence SHALL be the same as the Test Sequence #06 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #07 Alt. Nominal Matching ID containing EventID with one pending Event

This test sequence SHALL be the same as the Test Sequence #07 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

Test Sequence #08 Alt. Nominal Matching ID containing EventID with two pending Events

This test sequence SHALL be the same as the Test Sequence #08 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all auth keys and certificates SHALL be based on BrainpoolP256r1.

### 4.5.7 ES15 (SM-DS -- SM-DS): TLS, Mutual Authentication, Client, Session Establishment

#### 4.5.7.1 TC\_ALT\_SM-DS\_ES15\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST

Perform all test sequences defined in section 4.6.1.2.1 with the following variables set as follows:

 CLIENT = Alternative SM-DS under test

o CERT\_CLIENT\_TLS = #CERT\_SM\_DS\_TLS for NIST

 SERVER = Root S\_SM-DS

o CERT\_S\_SERVER\_TLS = #CERT\_S\_SM\_DS\_TLS for NIST

#### 4.5.7.2 TC\_ALT\_SM-DS\_ES15\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP

Perform all test sequences defined in section 4.6.1.2.2 with the following variables set as follows:

 CLIENT = Alternative SM-DS under test

o CERT\_CLIENT\_TLS = #CERT\_SM\_DS\_TLS for BRP

 SERVER = Root S\_SM-DS

o CERT\_S\_SERVER\_TLS = #CERT\_S\_SM\_DS\_TLS for BRP

### 4.5.8 ES12 (SM-DS -- SM-DP+): TLS, Mutual Authentication, Server, Session Establishment

#### 4.5.8.1 TC\_SM-DS\_ES12\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST

Perform all test sequences defined in section 4.6.2.2.1 with the following variables set as follows:

 CLIENT = S\_SM-DP+

o CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DP\_TLS for NIST

 SERVER = Alternative or Root SM-DS under test.

o CERT\_SERVER\_TLS = CERT\_SM\_DS\_TLS for NIST

#### 4.5.8.2 TC\_SM-DS\_ES12\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP

Perform all test sequences defined in section 4.6.2.2.2 with the following variables set as follows:

 CLIENT = S\_SM-DP+

o CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DP\_TLS for BRP

 SERVER = Alternative or Root SM-DS under test.

o CERT\_SERVER\_TLS = CERT\_SM\_DS\_TLS for BRP

### 4.5.9 ES15 (SM-DS -- SM-DS): TLS, Mutual Authentication, Server, Session Establishment

#### 4.5.9.1 TC\_ROOT\_SM-DS\_ES15\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST

Perform all test sequences defined in section 4.6.2.2.1 with the following variables set as follows:

 CLIENT = Alternative S\_SM-DS

o CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DS\_TLS for NIST

 SERVER = Root SM-DS under test.

o CERT\_SERVER\_TLS = CERT\_SM\_DS\_TLS for NIST

#### 4.5.9.2 TC\_ROOT\_SM-DS\_ES15\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP

Perform all test sequences defined in section 4.6.2.2.2 with the following variables set as follows:

 CLIENT = Alternative S\_SM-DS

o CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DS\_TLS for BRP

 SERVER = Root SM-DS under test.

o CERT\_SERVER\_TLS = CERT\_SM\_DS\_TLS for BRP

### 4.5.10 ES11 (LPA -- SM-DS): TLS, Server Authentication, Session Establishment

#### 4.5.10.1 TC\_SM-DS\_ES11\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST

Perform all test sequences defined in section 4.6.3.2.1 with the following variables set as follows:

 CLIENT = S\_LPAd

 SERVER = SM-DS under test.

o CERT\_SERVER\_TLS = #CERT\_SM\_DS\_TLS for NIST

#### 4.5.10.2 TC\_SM-DS\_ES11\_Server\_Authentication\_for\_HTTPS\_EstablishmentBRP

Perform all test sequences defined in section 4.6.3.2.2 with the following variables set as follows:

 CLIENT = S\_LPAd

 SERVER = SM-DS under test.

o CERT\_SERVER\_TLS = #CERT\_SM\_DS\_TLS for BRP

## 4.6 TLS Interface

### 4.6.1 TLS, Mutual Authentication, Client, TLS Establishment

#### 4.6.1.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_023, RQ26\_024, RQ26\_025, RQ26\_025\_1, RQ26\_026, RQ26\_027, RQ26\_028

 RQ31\_032

 RQ45\_006, RQ45\_026, RQ45\_026\_1

 RQ56\_001, RQ56\_002, RQ56\_003

 RQ58\_001, RQ58\_002

 RQ59\_001

* RQ60\_002, RQ60\_003

 RQ61\_001

 RQ63\_006

 RQ510\_001

#### 4.6.1.2 Test Cases

##### 4.6.1.2.1 TC\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST

|  |  |
| --- | --- |
| General Initial Conditions for SM-DP+ as Client under test | |
| Entity | Description of the initial condition |
| SM-DP+ | * The SM-DP+ is ready to execute a download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration (for example, a suitable profile is available).There is currently no TLS connection established to the S\_SM-DS. |

|  |  |
| --- | --- |
| General Initial Conditions for SM-DS as Client under test | |
| **Entity** | Description of the initial condition |
| SM-DS | * EventID to be used by the S\_SM-DP+ is not already used in the SM-DS. * There is currently no TLS connection established to the S\_SM-DS. |

Test Sequence #01 Nominal: HTTPS Session Establishment

The purpose of this test is to verify that the Client correctly establishes an HTTPS Session with the Server using Mutual Authentication.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1)  • <EXT\_SHA256\_ECDSA> SHALL have at least the 'supported\_signature\_algorithms' extension set with HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03). | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,  #CERT\_S\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI) | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_CLIENT\_TLS, <CLIENT\_TLS\_EPHEM\_KEY>) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 3 | S\_SERVER → CLIENT | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,  <SERVER\_FINISHED>) | HTTPS connection established | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

Test Sequence #02 Nominal: Non-reuse of session keys

The purpose of this test sequence is to verify that the Client is not reusing ephemeral keys from the previous session.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | * The SM-DP+ is ready to execute a further download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration (for example, a further suitable profile is available). |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | | Expected result | | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | | | | |
| IC2 | CLIENT → S\_SERVER | | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(    #IUT\_CLIENT\_TLS\_VER,   <TLS\_CIPHER\_SUITES>,   <SESSION\_ID\_CLIENT>,   <EXT\_SHA256\_ECDSA>)  Extract <SESSION\_ID\_CLIENT>. | |  | |
| IC3 | S\_SERVER → CLIENT | | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(     #TLS\_VERSION\_1\_2, <S\_SEL\_TLS\_CIPHER\_SUITE>,   <SESSION\_ID\_RANDOM>,   #CERT\_S\_SERVER\_TLS,   <SERVER\_TLS\_EPHEM\_KEY>,   #CLIENT\_CERT\_TYPE,   #S\_SAH\_SHA256\_ECDSA,   #DIST\_NAME\_CI) | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(   #CERT\_CLIENT\_TLS, <CLIENT\_TLS\_EPHEM\_KEY>)  Extract <SESSION\_ID\_RANDOM>.  Extract <CLIENT\_TLS\_EPHEM\_KEY> from the ClientKeyExchange message. | |  | |
| IC4 | S\_SERVER → CLIENT | | MTD\_TLS\_SERVER\_END(    #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | HTTPS connection established | |  | |
| IC5 | S\_SERVER  → CLIENT | Close TLS session (unless CLIENT has already closed TLS session) | | | | | |
| IC6 | When the Client under test is the SM-DP+, initiate a further download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, repeat IC1 | | | | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | | | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  <SESSION\_ID\_CLIENT> SHALL be different from any non-empty value of session\_id (in ClientHello or ServerHello) used in IC2.  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see Note 1) | |  |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,  #CERT\_S\_SERVER\_TLS, <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI) | | | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_CLIENT\_TLS, <CLIENT\_TLS\_EPHEM\_KEY>)  Verify that in the ClientKeyExchange message:  • <CLIENT\_TLS\_EPHEM\_KEY> is different from the one used by the CLIENT in IC1 | | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 3 | S\_SERVER → CLIENT | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,  <SERVER\_FINISHED>) | | | HTTPS connection established | | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | | | | |

Test Sequence #03 Error: Invalid Server TLS Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(  #TLS\_VERSION\_1\_1,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,  #CERT\_S\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI)  Note: if the Client sends an Alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC, then the S\_SERVER might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC which occur after the Alert. | Client sends a TLS Fatal-alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ510\_001 RQ59\_001 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

Test Sequence #04 Error: Invalid Server TLS Certificate Signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>, #CERT\_S\_SERVER\_TLS\_INV\_SIG,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI)  Note: if the Client sends an Alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC, then the S\_SERVER might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC which occur after the Alert. | Client sends a TLS Fatal-alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

Test Sequence #05 Error: Expired Server TLS Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(  #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>, #CERT\_S\_SERVER\_TLS\_EXPIRED,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI)  Note: if the Client sends an Alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC, then the S\_SERVER might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC which occur after the Alert. | Client sends a TLS Fatal-alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

Test Sequence #06 Error: Invalid Server TLS Certificate with critical extension not set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(  #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>, #CERT\_S\_SERVER\_TLS\_INV\_CRITICAL\_EXT,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI)  Note: if the Client sends an Alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC, then the S\_SERVER might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC which occur after the Alert. | Client sends a TLS Fatal-alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

Test Sequence #07 Error: Invalid Server TLS Certificate with invalid 'key usage' extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(  #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>, #CERT\_S\_SERVER\_TLS\_INV\_KEY\_USAGE,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI)  Note: if the Client sends an Alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC, then the S\_SERVER might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC which occur after the Alert. | Client sends a TLS Fatal-alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

Test Sequence #08 Error: Invalid TLS Certificate with invalid 'extended key usage' extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(  #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>, #CERT\_S\_SERVER\_TLS\_INV\_EXT\_KEY\_USAGE,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI)  Note: if the Client sends an Alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC, then the S\_SERVER might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC which occur after the Alert. | Client sends a TLS Fatal-alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

Test Sequence #09 Error: Invalid Client TLS Certificate with invalid 'Certificate Policies' extensions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | When the Client under test is the SM-DP+, initiate the download order procedure (see SGP.22 [2] section 3.1.1) for the SM-DS use case with smdsAddress #TEST\_ROOT\_DS\_ADDRESS to be used for Event Registration.  When the Client under test is the SM-DS, the S\_SM-DP+ calls ES12.RegisterEvent configured as follows:  MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES12,  #PATH\_REGISTER\_EVENT,  MTD\_REGISTER\_EVENT(  #EID1,  #TEST\_DP\_ADDRESS1,  <EVENT\_ID>,  TRUE) | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_CLIENT\_TLS\_VER,  <TLS\_CIPHER\_SUITES>,  <SESSION\_ID\_CLIENT>,  <EXT\_SHA256\_ECDSA>)  Verify that: • <TLS\_CIPHER\_SUITES> SHALL contain at least one of TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 orTLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (see note 1) | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(  #TLS\_VERSION\_1\_2,  <S\_SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>, #CERT\_S\_SERVER\_TLS\_INV\_CERT\_POL,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  #S\_SAH\_SHA256\_ECDSA,  #DIST\_NAME\_CI)  Note: if the Client sends an Alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC, then the S\_SERVER might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC which occur after the Alert. | Client sends a TLS Fatal-alert during or after any of the messages sent by the S\_SERVER in MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| Note 1: if the verification fails, the test tool cannot continue execution while remaining compliant with both SGP.22 [2] and RFC 5246 [27]. | | | | |

##### 4.6.1.2.2 TC\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP

Test Sequence #01 Nominal: HTTPS Session Establishment

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.6.1.2.1 TC\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

Test Sequence #02 Nominal: Non-reuse of session keys

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.6.1.2.1 TC\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

### 4.6.2 TLS, Mutual Authentication, Server, TLS Establishment

#### 4.6.2.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_023, RQ26\_024, RQ26\_025, RQ26\_026, RQ26\_027, RQ26\_028

 RQ45\_006, RQ45\_026, RQ45\_026\_1

 RQ56\_002

 RQ59\_001

 RQ60\_003

 RQ61\_001

#### 4.6.2.2 Test Cases

##### 4.6.2.2.1 TC\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST

Test Sequence #01 Nominal: HTTPS Session Establishment

The purpose of this test is to verify that the Server correctly establishes an HTTPS Session with the Client using Mutual Authentication.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS,  <CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | RQ26\_023 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #02 Nominal: Non-reuse of session keys

The purpose of this test sequence is to verify that the Server is not reusing ephemeral keys from the previous session.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH  Extract <SERVER\_TLS\_EPHEM\_KEY> from the ServerKeyExchange message | | | |
| IC2 | Terminate the TLS session | | | |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the ServerKeyExchange message: •<SERVER\_TLS\_EPHEM\_KEY> is different from the <SERVER\_TLS\_EPHEM\_KEY> value used in IC1. | RQ26\_025 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS,  <CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | RQ26\_023 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #03 Nominal: HTTPS Session Establishment with supported and unsupported Cipher Suites

The purpose of this test is to verify that the Server correctly establishes an HTTPS Session with the Client when supported and unsupported Cipher Suites are offered by the Client.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #PROP\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message:  •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS,  <CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | RQ26\_023 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #04 Error: Invalid TLS Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_1,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  NO\_PARAM) | Server sends a TLS Fatal-alert | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ510\_001 RQ59\_001 RQ61\_001 |

Test Sequence #05 Error: Unsupported Cipher Suites and Extensions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #UNSUP\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #EXT\_SHA256\_RSA) | Server sends a TLS Fatal-alert | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ510\_001 RQ59\_001 RQ61\_001 |

Test Sequence #06 Error: Invalid Client TLS Certificate Signature

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS\_INV\_SIG,  <CLIENT\_TLS\_EPHEM\_KEY>)  Note: if the Server sends an Alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH, then the S\_CLIENT might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH which occur after the Alert. | Server sends a TLS Fatal-alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #07 Error: Expired Client TLS Certificate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS\_EXPIRED,  <CLIENT\_TLS\_EPHEM\_KEY>)  Note: if the Server sends an Alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH, then the S\_CLIENT might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH which occur after the Alert. | Server sends a TLS Fatal-alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #08 Error: Invalid Client TLS Certificate with critical extension not set

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH( #CERT\_S\_CLIENT\_TLS\_INV\_CRITICAL\_EXT,  <CLIENT\_TLS\_EPHEM\_KEY>)  Note: if the Server sends an Alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH, then the S\_CLIENT might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH which occur after the Alert. | Server sends a TLS Fatal-alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #09 Error: Invalid Client TLS Certificate with invalid 'key usage' extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS\_INV\_KEY\_USAGE,  <CLIENT\_TLS\_EPHEM\_KEY>)  Note: if the Server sends an Alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH, then the S\_CLIENT might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH which occur after the Alert. | Server sends a TLS Fatal-alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #10 Error: Invalid TLS Certificate with invalid 'extended key usage' extension

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS\_INV\_EXT\_KEY\_USAGE,  <CLIENT\_TLS\_EPHEM\_KEY>)  Note: if the Server sends an Alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH, then the S\_CLIENT might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH which occur after the Alert. | Server sends a TLS Fatal-alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #11 Error: Invalid Client TLS Certificate with invalid 'Certificate Policies' extensions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS\_INV\_CERT\_POL,  <CLIENT\_TLS\_EPHEM\_KEY>)  Note: if the Server sends an Alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH, then the S\_CLIENT might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH which occur after the Alert. | Server sends a TLS Fatal-alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_006 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

Test Sequence #12 Error: No suitable Client certificate available

The purpose of this test is to verify that the Server does not establish an HTTPS Session with the Client using Mutual Authentication when the CERT.CLIENT.TLS certificate of the S\_CLIENT certificate message contains no certificates (the certificate\_list structure has a length of zero).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI)  Verify that in the Server Hello message: •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  NO\_PARAM,  <CLIENT\_TLS\_EPHEM\_KEY>)  Note: if the Server sends an Alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH, then the S\_CLIENT might not send the messages specified in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH which occur after the Alert. | Server sends a TLS Fatal-alert during or after any of the messages sent by the S\_CLIENT in MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ26\_028 RQ45\_026 RQ45\_026\_1 RQ510\_001 RQ56\_002 RQ59\_001 RQ60\_003 RQ61\_001 |

##### 4.6.2.2.2 TC\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP

Test Sequence #01 Nominal: HTTPS Session Establishment

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.6.2.2.1 TC\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

Test Sequence #02 Nominal: Non-reuse of session keys

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.6.2.2.1 TC\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

Test Sequence #03 Nominal: HTTPS Session Establishment with supported and unsupported Cipher Suites

This test sequence SHALL be the same as the Test Sequence #03 defined in section 4.6.2.2.1 TC\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

### 4.6.3 TLS, Server Authentication, TLS Establishment

#### 4.6.3.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_023, RQ26\_024, RQ26\_025, RQ26\_025\_1, RQ26\_026, RQ26\_027, RQ26\_028

 RQ31\_032

 RQ45\_026, RQ45\_026\_1

 RQ56\_001, RQ56\_002, RQ56\_003

 RQ58\_001, RQ58\_002

 RQ60\_002

 RQ61\_001

 RQ63\_006

#### 4.6.3.2 Test Cases

##### 4.6.3.2.1 TC\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST

Test Sequence #01 Nominal: HTTPS Session Establishment

The purpose of this test is to verify that the Server correctly establishes an HTTPS Session with the Client.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2, <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>)  Verify that in the Server Hello message:  •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_028 RQ31\_032 RQ31\_032\_1 RQ45\_026 RQ45\_026\_1 RQ56\_001 RQ56\_002 RQ56\_003 RQ58\_001 RQ58\_002 RQ60\_002 RQ61\_001 |
| 2 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | RQ26\_023 RQ26\_026 RQ26\_027 RQ31\_032 RQ45\_026 RQ45\_026\_1 RQ56\_001 RQ58\_001 RQ60\_002 RQ61\_001 |

Test Sequence #02 Nominal: Non-reuse of session keys

The purpose of this test sequence is to verify that the Server is not reusing ephemeral keys from the previous session.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH  Extract <SERVER\_TLS\_EPHEM\_KEY> from the ServerKeyExchange message | | | |
| IC2 | Terminate the TLS session | | | |
| 1 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2, <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS, <SERVER\_TLS\_EPHEM\_KEY>)  Verify that in the ServerKeyExchange message: •<SERVER\_TLS\_EPHEM\_KEY> is different from the <SERVER\_TLS\_EPHEM\_KEY> value used in IC1. | RQ26\_025 RQ31\_032 |
| 2 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | RQ26\_023 RQ26\_026 RQ26\_027 RQ31\_032 RQ45\_026 RQ45\_026\_1 RQ56\_001 RQ58\_001 RQ60\_001 RQ60\_002 RQ61\_001 |

Test Sequence #03 Nominal: HTTPS Session Establishment with supported and unsupported Cipher Suites

The purpose of this test is to verify that the Server correctly establishes an HTTPS Session with the Client when supported and unsupported Cipher Suites are offered by the Client.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #PROP\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2, <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  <SERVER\_TLS\_EPHEM\_KEY>)  Verify that in the Server Hello message:  •<SEL\_TLS\_CIPHER\_SUITE> SHALL contain either TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 OR TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_028 RQ31\_032 RQ31\_032\_1 RQ45\_026 RQ45\_026\_1 RQ56\_001 RQ56\_002 RQ56\_003 RQ58\_001 RQ58\_002 RQ60\_002 RQ61\_001 |
| 2 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | RQ26\_023 RQ26\_026 RQ26\_027 RQ31\_032 RQ45\_026 RQ45\_026\_1 RQ56\_001 RQ58\_001 RQ60\_002 RQ61\_001 |

Test Sequence #04 Error: Invalid TLS Version

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_1,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  NO\_PARAM) | Server sends a TLS Fatal-alert | RQ26\_023 RQ26\_025 RQ26\_026 RQ26\_027 RQ31\_032 RQ45\_026 RQ45\_026\_1 RQ56\_001 RQ58\_001 RQ60\_002 RQ61\_001 |

Test Sequence #05 Error: Unsupported Cipher Suites and Extensions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #UNSUP\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #EXT\_SHA256\_RSA) | Server sends a TLS Fatal-alert | RQ26\_023 RQ26\_024 RQ26\_025 RQ26\_026 RQ26\_027 RQ31\_032 RQ45\_026 RQ45\_026\_1 RQ56\_001 RQ58\_001 RQ60\_002 RQ61\_001 |

##### 4.6.3.2.2 TC\_Server\_Authentication\_for\_HTTPS\_EstablishmentBRP

Test Sequence #01 Nominal: HTTPS Session Establishment

This test sequence SHALL be the same as the Test Sequence #01 defined in section 4.6.3.2.1 TC\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

Test Sequence #02 Nominal: Non-reuse of session keys

This test sequence SHALL be the same as the Test Sequence #02 defined in section 4.6.3.2.1 TC\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

Test Sequence #03 Nominal: HTTPS Session Establishment with supported and unsupported Cipher Suites

This test sequence SHALL be the same as the Test Sequence #03 defined in section 4.6.3.2.1 TC\_Server\_Authentication\_for\_HTTPS\_EstablishmentNIST, except that the brainpoolP256r1 curve is used.

## 4.7 LPAe Interfaces

This section is defined as FFS.

# 5 Procedure - Behaviour Testing

## 5.1 General Overview

## 5.2 eUICC Behaviour

### 5.2.1 Retry mechanism

#### 5.2.1.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ26\_029, RQ26\_030

 RQ31\_130, RQ31\_131, RQ31\_132, RQ31\_133, RQ31\_134, RQ31\_135, RQ31\_137, RQ31\_139, RQ31\_140, RQ31\_141

 RQ57\_025, RQ57\_026, RQ57\_027, RQ57\_028, RQ57\_029, RQ57\_030, RQ57\_033, RQ57\_034, RQ57\_035, RQ57\_036, RQ57\_037, RQ57\_038, RQ57\_039, RQ57\_047, RQ57\_112

#### 5.2.1.2 Test Cases

##### 5.2.1.2.1 TC\_eUICC\_PrepareDownload\_Retry\_ReuseOTKeys

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification |

Test Sequence #01 Nominal: Confirmation Code retry mechanism by reusing previous One-Time key pair

The purpose of this test is to check the Confirmation Code retry mechanism. The S\_LPAd simulates that an incorrect Confirmation Code has been filled by the End User. Then, the S\_LPAd sends another ES10b.PrepareDownload function with a correct Confirmation Code value. In this case, the eUICC does not have to generate a new one-time key pair and uses the previous one given by the SM-DP+.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | | Sequence / Description | | Expected result | REQ |
| IC1 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | | | | | |
| 1 | S\_LPAd → eUICC | | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_WITH\_CC) | | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Extract the <OTPK\_EUICC\_ECKA> and reuse the same value in step 4 | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ26\_029 RQ26\_030 |
| 2 | Execute the Common Mutual Authentication procedure between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP are sent to the eUICC * the same GSMA CI as for the first attempt has been chosen for signing and for verification | | | | | RQ57\_047 |
| 3 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE2, <S\_TRANSACTION\_ID>) | | | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_RETRY\_CC) | | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <OTPK\_EUICC\_ECKA> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC. | | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ31\_137 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ57\_033 RQ26\_029 RQ26\_030 |

Test Sequence #02 Nominal: Retry after a CancelSession Reason “Postponed”

The purpose of this test is to check that the eUICC can reuse the one-time key pair generated during a previous attempt. In this case, the S\_LPAd simulates that the End User has postponed the download of the Profile.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | | Sequence / Description | Expected result | REQ |
| IC1 | | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | | | |
| 1 | | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_WITH\_CC) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Extract the <OTPK\_EUICC\_ECKA> and reuse the same value in step 4 | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ26\_029 RQ26\_030 |
| 2 | | S\_LPAd → eUICC | MTD\_STORE\_DATA( #CANCEL\_SESSION\_POSTPONED) | #R\_CANCEL\_SESSION\_POSTPONED  SW = 0x9000 | RQ57\_112 |
| 3 | | Execute the Common Mutual Authentication procedure between the eUICC and the S\_SM-DP+   #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP are sent to the eUICC   the same GSMA CI as for the first attempt has been chosen for signing and for verification | | | RQ57\_047 |
| 4 | | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_RETRY\_C) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <OTPK\_EUICC\_ECKA> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC. | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ31\_137 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ57\_033 RQ26\_029 RQ26\_030 |

Test Sequence #03 Nominal: Retry after a CancelSession Reason “Timeout”

The purpose of this test is to check that the eUICC can reuse the one-time key pair generated during a previous attempt. In this case, the S\_LPAd simulates that the End User does not confirm the download of the Profile within the timeout interval.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_WITH\_CC) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Extract the <OTPK\_EUICC\_ECKA> and reuse the same value in step 4 | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ26\_029 RQ26\_030 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #CANCEL\_SESSION\_TIMEOUT) | #R\_CANCEL\_SESSION\_TIMEOUT  SW = 0x9000 | RQ57\_112 |
| 3 | Execute the Common Mutual Authentication procedure between the eUICC and the S\_SM-DP+   #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP are sent to the eUICC   the same GSMA CI as for the first attempt has been chosen for signing and for verification | | | RQ57\_047 |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_RETRY\_CC) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <OTPK\_EUICC\_ECKA> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC. | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ31\_137 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ57\_033 RQ26\_029 RQ26\_030 |

##### 5.2.1.2.2 TC\_eUICC\_PrepareDownload\_Retry\_NewOTKeys

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is not loaded on the eUICC. |
| eUICC | The communication between the S\_Device and the eUICC has been initialized and the S\_LPAd has selected the ISD-R.  Common Mutual Authentication procedure has been successfully executed between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP have been sent to the eUICC * the same GSMA CI has been chosen for signing and for verification |

Test Sequence #01 Nominal: Confirmation Code retry mechanism by not reusing previous One-Time key pair

The purpose of this test is to check the Confirmation Code retry mechanism. The S\_LPAd simulates that an incorrect Confirmation Code has been filled by the End User. Then, the S\_LPAd sends another ES10b.PrepareDownload function with a correct Confirmation Code value. In this case, the eUICC does not support the storage of unused one-time key pair or the eUICC has discarded the previous one-time public key: we expect the eUICC to generate a new set of keys.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE1, <S\_TRANSACTION\_ID>) | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_WITH\_CC) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_WITH\_CC.  Extract the <OTPK\_EUICC\_ECKA> and reuse the same value in step 4 | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ26\_029 RQ26\_030 |
| 2 | Execute the Common Mutual Authentication procedure between the eUICC and the S\_SM-DP+   * #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP are sent to the eUICC * the same GSMA CI as for the first attempt has been chosen for signing and for verification | | | RQ57\_047 |
| 3 | <S\_HASHED\_CC> = MTD\_GENERATE\_HASHED\_CC(#CONFIRMATION\_CODE2, <S\_TRANSACTION\_ID>) | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_RETRY\_CC) | #R\_PREP\_DOWNLOAD\_WITH\_CC  SW=0x9000  The <EUICC\_SIGNATURE2> SHALL be verified with the #PK\_EUICC\_ECDSA.  Verify that the <S\_TRANSACTION\_ID> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <S\_HASHED\_CC> present in the euiccSigned2 is the same as in #PREP\_DOWNLOAD\_RETRY\_CC.  Verify that the <OTPK\_EUICC\_ECKA> present in the euiccSigned2 is NOT the same as in #PREP\_DOWNLOAD\_RETRY\_CC. | RQ31\_130 RQ31\_131 RQ31\_132 RQ31\_133 RQ31\_134 RQ31\_135 RQ31\_139 RQ31\_140 RQ31\_141 RQ31\_137 RQ57\_025 RQ57\_026 RQ57\_027 RQ57\_028 RQ57\_029 RQ57\_030 RQ57\_034 RQ57\_035 RQ57\_036 RQ57\_037 RQ57\_038 RQ57\_039 RQ57\_033 RQ26\_029 RQ26\_030 |

### 5.2.2 Forbidden PPRs

#### 5.2.2.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ57\_053, RQ57\_054

 RQ57\_056, RQ57\_057

 RQ25\_025, RQ25\_023

 RQ55\_032

#### 5.2.2.2 Test Cases

##### 5.2.2.2.1 TC\_eUICC\_ForbiddenPPRs

|  |  |
| --- | --- |
| General Initial Conditions | |
| **Entity** | Description of the general initial condition |
| eUICC | There is no Profile installed in the eUICC. |

Test Sequence #01 Nominal: PPR1 management and handling when Operational Profile is installed

The purpose of this test is to verify that the eUICC automatically sets PPR1 in the forbiddenProfilePolicyRules of EUICCInfo2 when an Operational Profile is installed. Any Operational Profile with PPR1 SHALL be rejected by the eUICC once an Operational Profile has been installed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | forbiddenProfilePolicyRules in EUICCInfo2 does not contain ppr1 | RQ57\_053RQ57\_054RQ57\_056 RQ57\_057 |
| 2 | Install PROFILE\_OPERATIONAL1 | | | RQ57\_057 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | forbiddenProfilePolicyRules in EUICCInfo2 contains ppr1(1) | RQ57\_053RQ57\_054 RQ57\_056 |
| 4 | Execute the Common Mutual Authentication procedure between the eUICC and the S\_SM-DP+   #GET\_EUICC\_INFO1, #GET\_EUICC\_CHALLENGE and #AUTHENTICATE\_SMDP are sent to the eUICC   the same GSMA CI is chosen for signing and for verification | | | |
| 5 | Execute the Sub-procedure Profile Download and Installation – End User Confirmation between the eUICC and the S\_SM-DP+   #PREP\_DOWNLOAD\_NO\_CC is sent to the eUICC | | | |
| 6 | Generate the <OTPK\_S\_SM\_DP+\_ECKA> and <OT\_SK\_S\_SM\_DP+\_ECKA> | | | |
| 7 | <BPP> = MTD\_GENERATE\_BPP(  #S\_INIT\_SC\_PROF1,  #CONF\_ISDP\_EMPTY,  #METADATA\_OP\_PROF4,  NO\_PARAM,  #UPP\_OP\_PROF4) | | | |
| 8 | Split the <BPP> into several segments arrays named:   <BPP\_SEG\_INIT>   <BPP\_SEG\_A0>   <BPP\_SEG\_A1>   <BPP\_SEG\_A3> | | | |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_INIT>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 10 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A0>) | SW=0x9000 without response data for all STORE DATA commands |  |
| 11 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  <BPP\_SEG\_A1>) | SW=0x9000 with the response data #R\_PIR\_PPR\_NOT\_ALLOWED | RQ25\_025RQ25\_023RQ57\_056 RQ55\_032 RQ57\_057 |
| 12 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE\_INFO1\_DISABLED  }  SW=0x9000 |  |
| 13 | Delete PROFILE\_OPERATIONAL1 | | | |
| 14 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | forbiddenProfilePolicyRules in EUICCInfo2 does not contain ppr1 | RQ57\_053RQ57\_054RQ57\_056 |

### 5.2.3 eUICC's RAT

#### 5.2.3.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ31\_097, RQ31\_097, RQ31\_098, RQ31\_130

 RQ32\_057

 RQ57\_117, RQ57\_118, RQ57\_119, RQ57\_123, RQ57\_179, RQ57\_180, RQ57\_181, RQ57\_182, RQ57\_184

#### 5.2.3.2 Test Cases

##### 5.2.3.2.1 TC\_eUICC\_GetProfilesInfo\_GetRAT\_RSPSession

Test Sequence #01 Nominal: GetProfilesInfo and GetRAT during RSP session

The purpose of this test is to ensure that the eUICC can be requested during a RSP session context to retrieve the list of installed Profiles and the Rules Authorization Table.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The eUICC's RAT is configured as detailed SGP.21 Annex H:   * one PPAR authorizing PPR1 and PPR2 for all MNOs with End User consent required (i.e. #PPRS\_ALLOWED) * no additional rules |
| eUICC | The PROFILE\_OPERATIONAL1 is installed and Enabled on the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO1) | #R\_EUICC\_INFO1  SW = 0x9000  Extract the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION> from response data and verify if they contain at least one same GSMA CI Key ID |  |
| IC4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_CHALLENGE) | #R\_CHALLENGE  SW = 0x9000  Extract the <EUICC\_CHALLENGE> |  |
| IC5 | The following inputs are required for Step IC6 as described in the InitiateAuthentication function:   * <S\_TRANSACTION\_ID> * <EUICC\_CHALLENGE> * <S\_SMDP\_CHALLENGE> * <S\_SMDP\_SIGNATURE1> * Set the <EUICC\_CI\_PK\_ID\_TO\_BE\_USED> to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> * Choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | | | |
| IC6 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTHENTICATE\_SMDP) | #R\_AUTHENTICATE\_SMDP  SW = 0x9000 |  |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_RAT) | #R\_DEFAULT\_RAT with exact same structure and order  SW = 0x9000 | RQ57\_179 RQ57\_180 RQ57\_181 RQ57\_182 RQ57\_184 RQ31\_097 |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_PROFILES\_INFO\_ALL) | response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE\_INFO1  }  SW = 0x9000 | RQ32\_057 RQ57\_117 RQ57\_118 RQ57\_119 RQ57\_123 RQ31\_098 |
| 3 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_NO\_CC) | #R\_PREP\_DOWNLOAD\_NO\_CC  SW=0x9000 | RQ31\_130 |

### 5.2.4 eUICC File Structure

#### 5.2.4.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ34\_003, RQ34\_005\_1, Q34\_010, RQ34\_011, RQ34\_004\_1

#### 5.2.4.2 Test Cases

##### 5.2.4.2.1 TC\_eUICC\_Default\_FileSystem

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | There is no Profile installed in the eUICC. |

Test Sequence #01 Nominal: Default file system available

The purpose of this test is to verify that if there is no Profile on the eUICC, the eUICC still ensures a file system to the Device.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_Device → eUICC | RESET | ATR present | RQ34\_003 RQ34\_004\_1 |
| 2 | S\_Device → eUICC | [SELECT\_MF] | FCP Template present with tag 0xA5 (Proprietary Information) containing 0x87 01 01 (Supported system commands = TERMINAL CAPABILITY)  SW=0x9000 | RQ34\_010 RQ34\_011 RQ34\_005\_1 RQ34\_003 RQ34\_004\_1 |
| 3 | S\_Device → eUICC | [TERMINAL\_CAPABILITY\_LPAd] | SW=0x9000 | RQ34\_005\_1 RQ34\_003 RQ34\_004\_1 |
| 4 | S\_Device → eUICC | [TERMINAL\_PROFILE] | Toolkit initialization THEN SW=0x9000 | RQ34\_003 RQ34\_004\_1 |

### 5.2.5 eUICC Delete Profile Process

#### 5.2.5.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_020

 RQ31\_027, RQ31\_028, RQ31\_183

 RQ57\_051, RQ57\_052, RQ57\_054

#### 5.2.5.2 Test Cases

##### 5.2.5.2.1 TC\_eUICC\_DeleteProfile\_ISDP\_And\_Components

Test Sequence #01 Nominal: ISD-P and Profile Components Deletion

The purpose of this test is to verify that when a Profile is deleted, the eUICC removes the ISD-P and all Profile Components related to it. In order to do so, we are checking the eUICC Non-Volatile Memory variation.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | There is no Profile installed on the eUICC. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | |
| IC3 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | Retrieve free non-volatile memory value (tag 0x82) from <EXT\_CARD\_RESOURCE> in EUICCInfo2 as <FREE\_MEMORY\_NO\_PROFILE> |  | |
| IC4 | Install PROFILE\_OPERATIONAL1 | | | |
| IC5 | Remove all Install Notifications from eUICC | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | Retrieve free non-volatile memory value (tag 0x82) from <EXT\_CARD\_RESOURCE> in EUICCInfo2 as <FREE\_MEM\_OP\_PROF1\_INSTALLED>  Verify that <FREE\_MEM\_OP\_PROF1\_INSTALLED> is lower than <FREE\_MEMORY\_NO\_PROFILE> | RQ31\_027 RQ31\_028 RQ57\_051 RQ57\_052 RQ57\_054 RQ31\_183 | |
| 2 | Delete PROFILE\_OPERATIONAL1 | | | |
| 3 | Remove the Delete Notification from eUICC | | | |
| 4 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #GET\_EUICC\_INFO2) | Retrieve free non-volatile memory value (tag 0x82) from <EXT\_CARD\_RESOURCE> in EUICCInfo2 as <FREE\_MEM\_OP\_PROF1\_DELETED>  Verify that <FREE\_MEM\_OP\_PROF1\_DELETED> is higher than <FREE\_MEM\_OP\_PROF1\_INSTALLED> | RQ31\_027 RQ31\_028 RQ57\_051 RQ57\_052 RQ57\_054 RQ24\_020 | |

### 5.2.6 eUICC Enable Profile Process

#### 5.2.6.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ35\_001, RQ35\_002, RQ35\_007

 RQ55\_048\_1

 RQ57\_135\_5

#### 5.2.6.2 Test Cases

##### 5.2.6.2.1 TC\_eUICC\_EnableProfile\_Twice\_Notifications

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed and Enabled on the eUICC. |
| eUICC | No Notification is stored in the eUICC's Pending Notifications List. |

Test Sequence #01 Nominal: Notifications generation

The purpose of this test is to verify that when an Enable Profile operation is performed and the current Enabled Profile is implicitly Disabled, both Notifications are generated. The eUICC automatically increments its sequence number each time a Notification is generated across all Profiles.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ | |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | | |
| IC3 | Install PROFILE\_OPERATIONAL2  The default Profile downloading procedure defined in section 2.2.3.1 SHALL be used with the following exceptions:   #CERT\_S\_SM\_DP2auth\_ECDSA SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #CERT\_S\_SM\_DPauth\_ECDSA   #TEST\_DP\_ADDRESS2 SHALL be set in #AUTH\_SMDP\_MATCH\_ID rather than #TEST\_DP\_ADDRESS1   #CERT\_S\_SM\_DP2pb\_ECDSA SHALL be set in #PREP\_DOWNLOAD\_NO\_CC rather than #CERT\_S\_SM\_DPpb\_ECDSA | | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_IN2\_PIR\_IN2  SW = 0x9000  Verify that <NOTIF\_SEQ\_NO\_IN2\_PIR> and <NOTIF\_SEQ\_NO\_IN2> follow this order in an incremental sequence (see NOTE) | | | RQ35\_001 RQ35\_002 RQ55\_048\_1 |
| 2 | Remove the ProfileInstallationResult and OtherSignedNotification for Install | | | | | |
| 3 | Enable PROFILE\_OPERATIONAL2 | | | | | |
| 4 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | | |
| 5 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | | |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | | #R\_LIST\_NOTIF\_DI1\_EN2 SW = 0x9000  Verify that  <NOTIF\_SEQ\_NO\_IN2> is lower than <NOTIF\_SEQ\_NO\_DI1>.  Verify that  <NOTIF\_SEQ\_NO\_DI1> and <NOTIF\_SEQ\_NO\_EN2> follow this order in an incremental sequence | RQ35\_001 RQ35\_002 RQ35\_007 RQ57\_135\_5 | |
| NOTE: In order to compare the sequence numbers, the test tool can retrieve the <NOTIF\_SEQ\_NO\_IN2\_PIR> value through the PIR returned at the end of the step IC3. | | | | | | |

### 5.2.7 eUICC Disable Profile Process

#### 5.2.7.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ24\_026

#### 5.2.7.2 Test Cases

##### 5.2.7.2.1 TC\_eUICC\_DisableProfile\_ApplicationManagement

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| eUICC | PROFILE\_OPERATIONAL1 is installed and Enabled. |

Test Sequence #01 Nominal: Application Selection/Deletion not available on Disabled Profile

The purpose of this test is to verify that when a Profile is Disabled, the eUICC does not allow the selection or deletion of any application within the Profile.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| IC2 | S\_Device → eUICC | [SELECT\_USIM] | FCP Template present SW=0x9000 |  |
| IC3 | S\_Device → eUICC | MTD\_SELECT(  #SSD\_AID) | SSD is selected  SW=0x9000 |  |
| IC4 | Disable PROFILE\_OPERATIONAL1 | | | |
| 1 | S\_Device → eUICC | [SELECT\_USIM] | USIM is not found  SW=0x6A82 | RQ24\_026 |
| 2 | S\_Device → eUICC | MTD\_SELECT(  #SSD\_AID) | SSD is not found  SW=0x6A82 | RQ24\_026 |
| 3 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | |
| 4 | S\_Device → eUICC | MTD\_SEND\_SMS\_PP(  [DELETE\_SSD]) | SW=0x91XX  or SW=0x9000 (i.e. envelope rejected, see NOTE)  or any error SW (i.e. envelope rejected,see NOTE) | RQ24\_026 |
| 5 | S\_Device 🡪eUICC | FETCH 'XX' | SMS POR received  SCP80 response status code equal to 0x06 (Unidentified security error) or 0x09 (TAR unknown) | RQ24\_026 |
| 6 | S\_Device 🡪 eUICC | TERMINAL RESPONSE | SW=0x9000 |  |
| 7 | Enable PROFILE\_OPERATIONAL1 | | | |
| 8 | S\_Device → eUICC | MTD\_SELECT(  #SSD\_AID) | SSD is selected  SW=0x9000 | RQ24\_026 |
| NOTE: Depending on the implementation, the eUICC MAY decide to not send back a POR (e.g. SW=0x9000 on the ENVELOPE command). Therefore, the steps 5 and 6 SHALL only be executed in case SW=0x91XX. | | | | |

### 5.2.8 eUICC Notifications

#### 5.2.8.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ57\_135\_6, RQ57\_142\_17, RQ57\_158\_1

#### 5.2.8.2 Test Cases

##### 5.2.8.2.1 TC\_eUICC\_Enable\_Disable\_Delete\_Notifications

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 with #METADATA\_EN\_DI\_DE\_NOTIFS is loaded on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | No Notification is stored in the eUICC's Pending Notifications List. |

Test Sequence #01 Nominal: Multiple Enable, Disable and Delete Notifications

The purpose of this test is to verify that when a Local Profile Management Operation (i.e. Enable, Disable and Delete Profile) is performed, all Notifications configured in the notificationConfigurationInfo are generated by the eUICC.

NOTE: In this sequence, the maximum number of Notifications simultaneously tested has been set as to two as there is not minimum defined in SGP.21 or SGP.22 [2] for the number of Notifications that can be stored by the eUICC.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ |
| IC1 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC2 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| IC3 | Enable PROFILE\_OPERATIONAL1 | | | | |
| IC4 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| IC5 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | #R\_LIST\_NOTIF\_EN1\_EN1  SW = 0x9000 | | RQ57\_135\_6 |
| 2 | Remove all the pending notifications | | | | |
| 3 | Disable PROFILE\_OPERATIONAL1 | | | | |
| 4 | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | | | |
| 5 | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | | | |
| 6 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | | #R\_LIST\_NOTIF\_DI1\_DI1 SW = 0x9000 | RQ57\_142\_17 |
| 7 | Remove all the pending notifications | | | | |
| 8 | Delete PROFILE\_OPERATIONAL1 | | | | |
| 9 | S\_LPAd → eUICC | MTD\_STORE\_DATA(  #LIST\_NOTIF\_ALL) | | #R\_LIST\_NOTIF\_DE1\_DE1 SW = 0x9000 | RQ57\_158\_1 |

## 5.3 Platform Procedures

### 5.3.1 Profile Download and Installation Procedure

This section is defined as FFS and not applicable for this version of test specification.

### 5.3.2 Common Mutual Authentication Process

This section is defined as FFS and not applicable for this version of test specification.

### 5.3.3 Profile Download and Installation Process

#### 5.3.3.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ44\_002

 RQ55\_033\_1

#### 5.3.3.2 Test Cases

##### 5.3.3.2.1 TC\_SM\_DP+\_ProfileMetadata

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| SM-DP+ | * SM-DP+ is configured with the #CERT\_SM\_DPauth\_ECDSA for NIST. * PROFILE\_OPERATIONAL1 (configured with metadata as specified in each sequence) is securely loaded as a Protected Profile Package using <PPK\_ENC> and <PPK\_MAC>. * Pending Profile PROFILE\_OPERATIONAL1 is in the 'Released' state with an empty MatchingID. * EID #EID1 is known to the SM-DP+ and associated to PROFILE\_OPERATIONAL1. * Confirmation Code is not provided by the Operator to the SM-DP+.   NOTE: the Profile Metadata for PROFILE\_OPERATIONAL1 SHALL be specified in the Initial Conditions for each individual sequence. |

Test Sequence #01 Nominal: all elements present

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | SM-DP+ is configured with #SMDP\_METADATA\_ALL for the pending Profile PROFILE\_OPERATIONAL1. |

Run the sequence below with the following parameter assignments:

 PARAM\_R\_AUTH\_CLIENT = #R\_AUTH\_CLIENT\_META\_ALL

 PARAM\_METADATA = #SMDP\_METADATA\_ALL

The sequence below has the following parameters:

 PARAM\_R\_AUTH\_CLIENT

 PARAM\_METADATA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,   #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP(PARAM\_R\_AUTH\_CLIENT) | RQ44\_002 |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK)  Construct the complete metadata element from the <SMDP\_METADATA\_SEG\_MAC> segment(s) and verify that it matches PARAM\_METADATA | RQ44\_002 |

Test Sequence #02 Nominal: optional elements missing

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | SM-DP+ is configured with #SMDP\_METADATA\_ABS for the pending Profile PROFILE\_OPERATIONAL1. |

This test sequence SHALL be the same as the Test Sequence #01 defined in the current section, with the following parameter assignments:

 PARAM\_R\_AUTH\_CLIENT = #R\_AUTH\_CLIENT\_META\_ABS

 PARAM\_METADATA = #SMDP\_METADATA\_ABS

Test Sequence #03 Nominal: large icon

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | SM-DP+ is configured with #SMDP\_METADATA\_OP\_PROF1\_2\_SEG for the pending Profile PROFILE\_OPERATIONAL1. |

This test sequence SHALL be the same as the Test Sequence #01 defined in the current section, with the following parameter assignments:

 PARAM\_R\_AUTH\_CLIENT = #R\_AUTH\_CLIENT\_META\_LARGE\_ICON

 PARAM\_METADATA = #SMDP\_METADATA\_OP\_PROF1\_2\_SEG

Test Sequence #04 Nominal: long Service Provider name

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | SM-DP+ is configured with #SMDP\_METADATA\_SPN\_LONG for the pending Profile PROFILE\_OPERATIONAL1. |

This test sequence SHALL be the same as the Test Sequence #01 defined in the current section, with the following parameter assignments:

 PARAM\_R\_AUTH\_CLIENT = #R\_AUTH\_CLIENT\_META\_SPN\_LONG

 PARAM\_METADATA = #SMDP\_METADATA\_SPN\_LONG

Test Sequence #05 Nominal: long Profile name

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | SM-DP+ is configured with #SMDP\_METADATA\_PN\_LONG for the pending Profile PROFILE\_OPERATIONAL1. |

This test sequence SHALL be the same as the Test Sequence #01 defined in the current section, with the following parameter assignments:

 PARAM\_R\_AUTH\_CLIENT = #R\_AUTH\_CLIENT\_META\_PN\_LONG

 PARAM\_METADATA = #SMDP\_METADATA\_PN\_LONG

Test Sequence #06 Nominal: non-ASCII characters

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | SM-DP+ is configured with #SMDP\_METADATA\_NON\_ASCII for the pending Profile PROFILE\_OPERATIONAL1. |

This test sequence SHALL be the same as the Test Sequence #01 defined in the current section, with the following parameter assignments:

 PARAM\_R\_AUTH\_CLIENT = #R\_AUTH\_CLIENT\_META\_NON\_ASCII

 PARAM\_METADATA = #SMDP\_METADATA\_NON\_ASCII

Test Sequence #07 Nominal: multiple notificationConfigurationInfo elements

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| SM-DP+ | SM-DP+ is configured with #SMDP\_METADATA\_NOTIF\_MULTI for the pending Profile PROFILE\_OPERATIONAL1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,   #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_META\_NOTIF\_MULTI) | RQ44\_002 RQ55\_033\_1 |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK)  Construct the complete metadata element from the response and verify that it matches #SMDP\_METADATA\_NOTIF\_MULTI | RQ44\_002 RQ55\_033\_1 |

## 5.4 Device Procedures

### 5.4.1 Local Profile Management - Add Profile

#### 5.4.1.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ29\_007\_1, RQ29\_008, RQ29\_009, RQ29\_011, RQ29\_013, RQ29\_015

 RQ31\_062, RQ31\_064, RQ31\_071, RQ31\_072, RQ31\_077, RQ31\_079, RQ31\_096, RQ31\_100, RQ31\_102, RQ31\_108, RQ31\_112, RQ31\_161

 RQ32\_001, RQ32\_002, RQ32\_004, RQ32\_062, RQ32\_066, RQ32\_068, RQ32\_069, RQ32\_070, RQ32\_071

 RQ41\_001, RQ41\_005, RQ44\_001

 RQC1\_006, RQC1\_008, RQC1\_009, RQC3\_014

#### 5.4.1.2 Test Cases

##### 5.4.1.2.1 TC\_LPAd\_AddProfile\_Manual\_Entry

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (manual entry)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ | |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 | |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by manual entry | | No error | RQ31\_064 RQ31\_077 RQ41\_001 | |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. | | | RQ31\_062 RQ32\_001 RQ32\_002 RQC1\_008 RQC1\_014 |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | | | |
| 9 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL1 is displayed in Disabled state | | | RQ32\_069 RQ32\_070 RQ32\_071 RQ44\_001 RQC3\_006 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | | | | |

Test Sequence #02 Nominal: Add a new Operational Profile by using Activation Code (manual entry) with Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (PROFILE\_OPERATIONAL1) associated with #CONFIRMATION\_CODE1. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | LPAd requests the Activation Code from the S\_End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_3 by manual entry | No error | RQ31\_064 RQ31\_077 RQ41\_001 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_3 as <MATCHING\_ID> | | | |
| 6 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_End User. | CONFIRMATION\_CODE1 is provided by manual entry. | RQ31\_108 RQ31\_112 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | |
| 8 | LPAd → S\_EndUser | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified, if not verified before.  (see NOTE 2) | RQ31\_062 RQC1\_008 RQC3\_014 |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 10 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071 RQ44\_001 RQC1\_006 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

##### 5.4.1.2.2 TC\_LPAd\_AddProfile\_QRCode\_scanning

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (QR code scanning)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by scanning the QR code | No error | RQ41\_001 RQ41\_005 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. | RQC1\_008 RQC3\_014 |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 9 | S\_EndUser→ LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071 RQ44\_001 RQC1\_006 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

##### 5.4.1.2.3 TC\_LPAd\_AddProfile\_ActivationCode\_InvalidFormat\_QRCode

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| eUICC | The PROFILE\_OPERATIONAL1 is not installed on the eUICC. |

Test Sequence #01 Error: Add a new Operational Profile by using wrongly formatted Activation Code (QR code scanning)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | Activation Code is requested from the End User by LPAd | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_INVALID\_FORMAT by scanning the QR code | LPAd provides an error message to the EndUser | RQ31\_072 |
| 3 | S\_EndUser→ LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is not displayed | RQ31\_072 |

##### 5.4.1.2.4 TC\_LPAd\_AddProfile\_ActivationCode\_InvalidFormat\_ManualEntry

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Add a new Operational Profile by using wrongly formatted Activation Code (Manual entry)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | Activation Code is requested from the End User by LPAd | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_INVALID\_FORMAT by manual entry | LPAd provides an error message to the EndUser | RQ31\_072 |
| 3 | S\_EndUser→ LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is not displayed | RQ31\_072 |

##### 5.4.1.2.5 TC\_LPAd\_AddProfile\_ConfirmationCode\_smdpSigned2\_QR

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (QR code scanning) with confirmation code indicated only in smdpSigned2

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1) which requires confirmation code. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | Activation Code is requested from the End User by LPAd | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by scanning the QR code | No error | RQ41\_001 RQ41\_005 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | |
| 6 | LPAd → S\_EndUser | Request the Confirmation Code from the S\_End User. | #CONFIRMATION\_CODE1 is provided by manual entry. | RQ31\_108 RQ31\_112 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | |
| 8 | LPAd → S\_EndUser | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified, if not verified before.  (see NOTE 2) |  |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 10 | S\_EndUser→ LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

##### 5.4.1.2.6 TC\_LPAd\_AddProfile\_ConfirmationCode\_smdpSigned2\_Manual\_Entry

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (manual entry) with confirmation code indicated only in smdpSigned2

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1) which requires confirmation code. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | Activation Code is requested from the End User by LPAd | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by manual entry | No error | RQ41\_001 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9 | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | |
| 6 | LPAd → S\_EndUser | Request the Confirmation Code from the S\_End User. | #CONFIRMATION\_CODE1 is provided by manual entry. | RQ31\_108 RQ31\_112 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | |
| 8 | LPAd → S\_EndUser | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified, if not verified before.  (see NOTE 2) |  |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 10 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

##### 5.4.1.2.7 TC\_LPAd\_AddProfile\_default\_SM-DP+\_address

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a new Operational Profile by using the default SM-DP+ Address

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for PROFILE\_OPERATIONAL1 linked to the EID of the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation  See NOTE1 | No error | RQ31\_079 RQ32\_062 RQ32\_068 |
| 2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 3 | PROC\_ES9+\_INIT\_AUTH | | | |
| 4 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> or missing MatchingID data object | | | |
| 5 | PROC\_ES9+\_GET\_BPP  (see NOTE 2) | | | |
| 6 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. |  |
| 7 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 8 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The Profile download by default SM-DP+ address MAY be implemented in different ways (e.g. some Device MAY implement a separate LUI menu for this function, some Device MAY request first the activation code, etc.). In order to enforce that the default SM-DP+ address is used the user SHALL not enter the Activation Code in case it is requested.  NOTE 2: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

##### 5.4.1.2.8 TC\_LPAd\_AddProfile\_QRCode\_with\_ConfirmationCode

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (QR code scanning) with confirmation code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | | Sequence / Description | Expected result. | | | REQ |
| 1 | S\_EndUser→ LPAd | | Initiate Add Profile operation | Activation Code is requested from the End User by LPAd | | | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | | Provide#ACTIVATION\_CODE\_3 by scanning the QR code |  | | | RQ41\_001 RQ41\_005 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_3 as <MATCHING\_ID> | | | | | | |
| 6 | LPAd → S\_EndUser | | Request the Confirmation Code from the S\_End User. | #CONFIRMATION\_CODE1 is provided by manual entry. | | | RQ31\_108 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | | | | |
| 8 | LPAd → S\_EndUser | | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified, if not verified before.  (see NOTE 2) | |  | |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | | | | |
| 10 | S\_EndUser→ LPAd | Initiate List Profile operation | | | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071  RQ44\_001 | |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | | | | |

##### 5.4.1.2.9 TC\_LPAd\_AddProfile\_PPRs

Test Sequence #01 Nominal: End User Confirmation after PPR1 consent requested

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed and End User Consent is required for #MCC\_MNC4 with gid1 and gid2 absent. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_4. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_4 (associated with PROFILE\_OPERATIONAL4). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP with #METADATA\_OP\_PROF4 used in #GET\_BPP\_OK | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure.  For LPAd supporting SGP.22 v2.2.2 or earlier:Relevant information about PPRs is shown and the End User consent is requested either at this point or at a previous point of the procedure.  (See NOTE)  For LPAd supporting SGP.22 v2.3 or later:  Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download.  (See NOTE) | RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_015 RQ31\_096 RQ31\_100 RQ31\_102 RQ29\_011 RQ29\_013 |
| 2 | S\_EndUser → LPAd | End User Confirmation is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 4 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL4 is displayed in Disabled state | RQ31\_161 |
| NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt. | | | | |

Test Sequence #02 Nominal: End User Confirmation after PPR2 consent requested

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR2 is allowed and End User Consent is required for #MCC\_MNC2 with gid1 and gid2 absent. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_3\_NO\_CC. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL3). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | | |
| IC4 | PROC\_ES9+\_GET\_BPP with #METADATA\_OP\_PROF3 used in #GET\_BPP\_OK | | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure  For LPAd supporting SGP.22 v2.2.2 or earlier:  Relevant information about PPRs is shown and the End User consent is requested either at this point or at a previous point of the procedure.  (See NOTE)  For LPAd supporting SGP.22 v2.3 or later:  Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download.  (See NOTE) | RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_015 RQ31\_096 RQ31\_100 RQ31\_102 RQ29\_011 RQ29\_013 |
| 2 | S\_EndUser → LPAd | End User Confirmation is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 4 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL3 is displayed in Disabled state | RQ31\_161 |
| NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt. | | | | |

Test Sequence #03 Nominal: Profile with PPR1 already present

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed for #MCC\_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required). |
| eUICC | The PROFILE\_OPERATIONAL4 with PPR1 is installed and enabled on the eUICC. |
| LPAd | Add Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download.  End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before. | RQ31\_071 |
| 2 | S\_EndUser → LPAd | End User Confirmation is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 4 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ31\_161 |

##### 5.4.1.2.10 VOID

##### 5.4.1.2.11 TC\_LPAd\_AddProfile\_Security\_Errors

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| eUICC | The PROFILE\_OPERATIONAL1 is not installed on the eUICC. |

Test Sequence #01 Error: Stop Add Profile Operation if No Confirmation Provided

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ | |
| IC1 | S\_EndUser→ LPAd | Initiate Add Profile operation | | LPAd requests the Activation Code from the End User |  | |
| IC2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 | | No error |  | |
| IC3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ (See NOTE 2) | | | | | |
| IC4 | PROC\_ES9+\_INIT\_AUTH | | | | | |
| IC5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | | | |
| IC6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. The End User SHALL not provide Confirmation. | The LPAd stops the Add Profile procedure automatically or provide means to stop the procedure by the End User. | | | RQ32\_001 RQ32\_002 RQ32\_004 |
| 2 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL1 is not displayed | | | RQ32\_004 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction if there is a way to abort it in step 1.  NOTE 2: Step IC6 is conditional – occurs only if Step 1 (Request for Confirmation) was not executed before. | | | | | | | |

##### 5.4.1.2.12 TC\_LPAd\_AddProfile\_Empty\_MatchingID

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a new Operational Profile by using empty MatchingID (QR code entry)

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for PROFILE\_OPERATIONAL1 linked to the EID of the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Direction** | **Sequence / Description** | **Expected result** | **REQ** |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_5 by scanning the QR code | No error | RQ31\_079 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. |  |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 9 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1:The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

Test Sequence #02 Nominal: Add a new Operational Profile by using empty MatchingID (manual entry)

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for PROFILE\_OPERATIONAL1 linked to the EID of the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Direction** | **Sequence / Description** | **Expected result** | **REQ** |
| 1 | S\_EndUser→ LPAd | Initiate Add Profile operation | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_5 by manual entry | No error | RQ31\_079 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. |  |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 9 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

##### 5.4.1.2.13 TC\_LPAd\_AddEnableProfile\_Manual\_Entry

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual entry)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | | Expected result | REQ | |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations | | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 | |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by manual entry | | No error | RQ31\_064 RQ31\_077 RQ41\_001 | |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified,if not verified before. | | | RQ31\_062 RQ32\_001 RQ32\_002 RQC1\_008 RQC1\_014 |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | | | |
| 9 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL1 is displayed in Enabled state | | | RQ32\_069 RQ32\_070 RQ32\_071 RQ44\_001 RQC3\_006 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | | | | |

Test Sequence #02 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual entry) with Confirmation Code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (PROFILE\_OPERATIONAL1) associated with #CONFIRMATION\_CODE1. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations | LPAd requests the Activation Code from the S\_End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_3 by manual entry | No error | RQ31\_064 RQ31\_077 RQ41\_001 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_3 as <MATCHING\_ID> | | | |
| 6 | LPAd → S\_EndUser | LPAd requests the Confirmation Code from the S\_End User. | CONFIRMATION\_CODE1 is provided by manual entry. | RQ31\_108 RQ31\_112 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | |
| 8 | LPAd → S\_EndUser | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified if not verified before.  (see NOTE 2) | RQ31\_062 RQC1\_008 RQC3\_014 |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 10 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Enabled state | RQ32\_069 RQ32\_070 RQ32\_071 RQ44\_001 RQC1\_006 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

##### 5.4.1.2.14 TC\_LPAd\_AddEnableProfile\_QRCode\_scanning

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (QR code scanning)

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by scanning the QR code | No error | RQ41\_001 RQ41\_005 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified,if not verified before. | RQC1\_008 RQC3\_014 |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 9 | S\_EndUser→ LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Enabled state | RQ32\_069 RQ32\_070 RQ32\_071 RQ44\_001 RQC1\_006 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

##### 5.4.1.2.15 TC\_LPAd\_AddEnableProfile\_ConfirmationCode\_smdpSigned2\_QR

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (QR code scanning) with confirmation code indicated only in smdpSigned2

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1) which requires confirmation code. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations | Activation Code is requested from the End User by LPAd | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by scanning the QR code | No error | RQ41\_001 RQ41\_005 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | |
| 6 | LPAd → S\_EndUser | Request the Confirmation Code from the S\_End User. | #CONFIRMATION\_CODE1 is provided by manual entry. | RQ31\_108 RQ31\_112 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | |
| 8 | LPAd → S\_EndUser | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified if not verified before.  (see NOTE 2) |  |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 10 | S\_EndUser→ LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Enabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

##### 5.4.1.2.16 TC\_LPAd\_AddEnableProfile\_ConfirmationCode\_smdpSigned2\_Manual\_Entry

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual entry) with confirmation code indicated only in smdpSigned2

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (PROFILE\_OPERATIONAL1) which requires confirmation code. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations | Activation Code is requested from the End User by LPAd | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_1 by manual entry | No error | RQ41\_001 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9 | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_1 as <MATCHING\_ID> | | | |
| 6 | LPAd → S\_EndUser | Request the Confirmation Code from the S\_End User. | #CONFIRMATION\_CODE1 is provided by manual entry. | RQ31\_108 RQ31\_112 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | |
| 8 | LPAd → S\_EndUser | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified, if not verified before.  (see NOTE 2) |  |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 10 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Enabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

##### 5.4.1.2.17 TC\_LPAd\_AddEnableProfile\_default\_SM-DP+\_address

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using the default SM-DP+ Address

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for PROFILE\_OPERATIONAL1 linked to the EID of the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations  See NOTE1 | No error | RQ31\_079 RQ32\_062 RQ32\_068 |
| 2 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 3 | PROC\_ES9+\_INIT\_AUTH | | | |
| 4 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> or missing MatchingID data object | | | |
| 5 | PROC\_ES9+\_GET\_BPP  (see NOTE 2) | | | |
| 6 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. |  |
| 7 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 8 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Enable state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The Profile download by default SM-DP+ address MAY be implemented in different ways (e.g. some Device MAY implement a separate LUI menu for this function, some Device MAY request first the activation code, etc.). In order to enforce that the default SM-DP+ address is used the user SHALL not enter the Activation Code in case it is requested.  NOTE 2: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

##### 5.4.1.2.18 TC\_LPAd\_AddEnableProfile\_QRCode\_with\_ConfirmationCode

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (QR code scanning) with confirmation code

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (PROFILE\_OPERATIONAL1). |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Step | Direction | | Sequence / Description | Expected result. | | | REQ |
| 1 | S\_EndUser→ LPAd | | Initiate combined Add and Enable Profile operations | Activation Code is requested from the End User by LPAd | | | RQ32\_062 RQ32\_066 |
| 2 | S\_EndUser→ LPAd | | Provide#ACTIVATION\_CODE\_3 by scanning the QR code |  | | | RQ41\_001 RQ41\_005 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT\_CC with #MATCHING\_ID\_3 as <MATCHING\_ID> | | | | | | |
| 6 | LPAd → S\_EndUser | | Request the Confirmation Code from the S\_End User. | #CONFIRMATION\_CODE1 is provided by manual entry. | | | RQ31\_108 |
| 7 | PROC\_ES9+\_GET\_BPP\_CC  (see NOTE 1) | | | | | | |
| 8 | LPAd → S\_EndUser | | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation, if not requested before.  (see NOTE 2) | For LPAd supporting SGP.22 v2.2.2 or earlier:  End User Intent successfully verified,if not verified before.  (see NOTE 2) | |  | |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | | | | |
| 10 | S\_EndUser→ LPAd | Initiate List Profile operation | | | PROFILE\_OPERATIONAL1 is displayed in Enabled state | RQ32\_069 RQ32\_070 RQ32\_071  RQ44\_001 | |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction.  NOTE 2: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | | | | |

##### 5.4.1.2.19 TC\_LPAd\_AddEnableProfile\_PPRs

Test Sequence #01 Nominal: End User Confirmation after PPR1 consent requested

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed and End User Consent is required for #MCC\_MNC4 with gid1 and gid2 absent. |
| LPAd | Add and Enable Profile operation is initiated by using #ACTIVATION\_CODE\_4. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_4 (associated with PROFILE\_OPERATIONAL4). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP with #METADATA\_OP\_PROF4 used in #GET\_BPP\_OK | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure   For LPAd supporting SGP.22 v2.2.2 or earlier:  Relevant information about PPRs is shown and the End User consent is requested either at this point or at a previous point of the procedure.  (See NOTE)  For LPAd supporting SGP.22 v2.3 or later:  Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download.  (See NOTE) | RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_015 RQ31\_096 RQ31\_100 RQ31\_102 RQ29\_011 RQ29\_013 |
| 2 | S\_EndUser → LPAd | End User Confirmation is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 4 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL4 is displayed in Enabled state | RQ31\_161 |
| NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt. | | | | |

Test Sequence #02 Nominal: End User Confirmation after PPR2 consent requested

|  |  |
| --- | --- |
| Initial Conditions | |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR2 is allowed and End User Consent is required for #MCC\_MNC2 with gid1 and gid2 absent. |
| LPAd | Add and Enable Profile operation is initiated by using #ACTIVATION\_CODE\_3\_NO\_CC. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_3 (associated with PROFILE\_OPERATIONAL3). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | | |
| IC4 | PROC\_ES9+\_GET\_BPP with #METADATA\_OP\_PROF3 used in #GET\_BPP\_OK | | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure    For LPAd supporting SGP.22 v2.2.2 or earlier:  Relevant information about PPRs is shown and the End User consent is requested either at this point or at a previous point of the procedure.  (See NOTE)  For LPAd supporting SGP.22 v2.3 or later:  Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download.  (See NOTE) | RQ29\_007\_1 RQ29\_008 RQ29\_009 RQ29\_015 RQ31\_096 RQ31\_100 RQ31\_102 RQ29\_011 RQ29\_013 |
| 2 | S\_EndUser → LPAd | End User Confirmation is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 4 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL3 is displayed in Enabled state | RQ31\_161 |
| NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt. | | | | |

Test Sequence #03 Nominal: Profile with PPR1 already present

|  |
| --- |
| Initial Conditions |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed for #MCC\_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required). |
| eUICC | The PROFILE\_OPERATIONAL4 with PPR1 is installed and enabled on the eUICC. |
| LPAd | Add and Enable Profile operation is initiated by using #ACTIVATION\_CODE\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #MATCHING\_ID\_1 (associated with PROFILE\_OPERATIONAL1). |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| IC2 | PROC\_ES9+\_INIT\_AUTH | | | |
| IC3 | PROC\_ES9+\_AUTH\_CLIENT  Extract <S\_TRANSACTION\_ID> | | | |
| IC4 | PROC\_ES9+\_GET\_BPP | | | |
| 1 | LPAd → S\_EndUser | Request for Confirmation if not requested before. | The LPA provides means for the End User Confirmation/Rejection of the Profile Download.  End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before. | RQ31\_071 |
| 2 | S\_EndUser → LPAd | End User Confirmation is performed within the period as defined in #IUT\_EU\_CONFIRMATION\_TIMEOUT |  |  |
| 3 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 4 | S\_EndUser → LPAd | List Profile operation is initiated | PROFILE\_OPERATIONAL1 is displayed in Disabled state | RQ31\_161 |

##### 5.4.1.2.20 TC\_LPAd\_AddEnableProfile\_Empty\_MatchingID

|  |  |
| --- | --- |
| **General Initial Conditions** | |
| **Entity** | **Description of the general initial condition** |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using empty MatchingID (QR code entry)

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for PROFILE\_OPERATIONAL1 linked to the EID of the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_5 by scanning the QR code | No error | RQ31\_079 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. |  |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 9 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Enabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1:The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

Test Sequence #02 Nominal: Add and Enable a new Operational Profile by using empty MatchingID (manual entry)

|  |  |
| --- | --- |
| **Initial Conditions** |  |
| **Entity** | **Description of the initial condition** |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for PROFILE\_OPERATIONAL1 linked to the EID of the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate combined Add and Enable Profile operations | LPAd requests the Activation Code from the End User | RQ32\_062 RQ32\_066 RQC1\_009 |
| 2 | S\_EndUser→ LPAd | Provide #ACTIVATION\_CODE\_5 by manual entry | No error | RQ31\_079 |
| 3 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 4 | PROC\_ES9+\_INIT\_AUTH | | | |
| 5 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> | | | |
| 6 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | |
| 7 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. |  |
| 8 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 9 | S\_EndUser → LPAd | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed in Enabled state | RQ32\_069 RQ32\_070 RQ32\_071 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

### 5.4.2 Local Profile Management – ListProfiles

#### 5.4.2.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ32\_053, RQ32\_054, RQ32\_058, RQ32\_059

 RQ44\_001

#### 5.4.2.2 Test Cases

##### 5.4.2.2.1 TC\_LPAd\_ListProfiles

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is installed on the eUICC. |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: List the Profiles and their current state

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is Enabled. |
| eUICC | The PROFILE\_OPERATIONAL2 is Disabled. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Request the list of Profiles | Display PROFILE\_OPERATIONAL1 with Enabled state and the PROFILE\_OPERATIONAL2 with Disabled state in human readable format. | RQ32\_053 RQ32\_054 RQ32\_058 RQ32\_059 RQ44\_001 |

### 5.4.3 Local Profile Management - SetNickname

#### 5.4.3.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ32\_0001, RQ32\_002, RQ32\_073, RQ32\_074, RQ32\_076, RQ32\_078

#### 5.4.3.2 Test Cases

##### 5.4.3.2.1 TC\_LPAd\_SetNickname

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Add a Nickname on a Disabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC (see Note). |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is not defined . |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_NOTIF as helper profile | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Select PROFILE\_OPERATIONAL1.  Indicates the intention to change the Profile Nickname of PROFILE\_OPERATIONAL1. | LPA offers to the End User a way to enter the Nickname. | RQ32\_074 |
| 2 | S\_EndUser→ LPAd | Set the Profile Nickname of the PROFILE\_OPERATIONAL1 to #NICKNAME2 | LPAd sets the Profile Nickname (No Error) | RQ32\_001 RQ32\_002 RQ32\_073 RQ32\_076 |
| 3 | Exit the UI menu | | | |
| 4 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName of PROFILE\_OPERATIONAL1. | Profile Nickname of PROFILE\_OPERATIONAL1 equals to #NICKNAME2 | RQ32\_078 |
| 5 | Power off then power on the Device | | | |
| 6 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName of PROFILE\_OPERATIONAL1. | Profile Nickname of PROFILE\_OPERATIONAL1 equals to #NICKNAME2 | RQ32\_078 |

Test Sequence #02 Nominal: Add a Nickname on an Enabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL2 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL2 is not defined. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Select PROFILE\_OPERATIONAL2.  Indicates the intention to change the Profile Nickname of PROFILE\_OPERATIONAL2 | LPA offers to the End User a way to enter the nickname. | RQ32\_074 |
| 2 | S\_EndUser→ LPAd | Set the Profile Nickname of the PROFILE\_OPERATIONAL2 to #NICKNAME3 | LPAd sets the Profile Nickname (No Error) | RQ32\_073 RQ32\_076 |
| 3 | Exit the UI menu | | | |
| 4 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName ofPROFILE\_OPERATIONAL2 | Profile Nickname of PROFILE\_OPERATIONAL2 equals to #NICKNAME3 | RQ32\_078 |
| 5 | Power off then power on the Device | | | |
| 6 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName of PROFILE\_OPERATIONAL2. | Profile Nickname of PROFILE\_OPERATIONAL2 equals to #NICKNAME3 | RQ32\_078 |

##### 5.4.3.2.2 TC\_LPAd\_EditNickname

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Edit the Nickname on a Disabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC (see NOTE). |
| eUICC | The PROFILE\_OPERATIONAL1 is Disabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL1 is equal to #NICKNAME1. |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_NOTIF as helper profile | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Select PROFILE\_OPERATIONAL1  Indicates the intention to change the Profile Nickname of PROFILE\_OPERATIONAL1 | Profile Nickname equals to #NICKNAME1  LPA offers to the End User a way to enter a new Nickname. | RQ32\_075 |
| 2 | S\_EndUser→ LPAd | Set the Profile Nickname of the PROFILE\_OPERATIONAL1 to #NICKNAME2 | LPAd sets the Profile Nickname (No Error) | RQ32\_073 RQ32\_076 |
| 3 | Exit the UI menu | | | |
| 4 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName ofPROFILE\_OPERATIONAL1 | Profile Nickname equals to #NICKNAME2 | RQ32\_078 |
| 5 | Power off then power on the Device | | | |
| 6 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName ofPROFILE\_OPERATIONAL1 | Profile Nickname equals to #NICKNAME2 | RQ32\_078 |

Test Sequence #02 Nominal: Edit the Nickname on an Enabled Operational Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL2 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is Enabled. |
| eUICC | The Nickname of the PROFILE\_OPERATIONAL2 is equal to #NICKNAME3. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Select PROFILE\_OPERATIONAL2  Indicates the intention to change the Profile Nickname of PROFILE\_OPERATIONAL2 | Profile Nickname equals to #NICKNAME3  LPA offers to the End User a way to enter a new Nickname. | RQ32\_075 |
| 2 | S\_EndUser→ LPAd | Set the Profile Nickname of the PROFILE\_OPERATIONAL2 to #NICKNAME4 | LPAd sets the Profile Nickname (No Error) | RQ32\_073 RQ32\_076 |
| 3 | Exit the UI menu | | | |
| 4 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName of PROFILE\_OPERATIONAL2 | Profile Nickname equals to #NICKNAME4 | RQ32\_078 |
| 5 | Power off then power on the Device | | | |
| 6 | S\_EndUser→ LPAd | Perform an LUI dependent action to display the NickName ofPROFILE\_OPERATIONAL2 | Profile Nickname equals to #NICKNAME4 | RQ32\_078 |

### 5.4.4 Local Profile Management - Delete Profile

#### 5.4.4.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ32\_001, RQ32\_002, RQ32\_004, RQ32\_044, RQ32\_047, RQ32\_050

 RQ35\_008

#### 5.4.4.2 Test Cases

##### 5.4.4.2.1 TC\_LPAd\_DeleteProfile\_Disabled\_without\_PPR

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Deleting Disabled Profile, No PPRs

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_TEST\_DP\_ADDRESS1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Disabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Delete Profile procedure is initiated for PROFILE\_OPERATIONAL1 | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. | RQ32\_001 RQ32\_002 RQ32\_044 |
| 2 | LPAd → S\_SM-DP+ | Delete Notification containing #ICCID\_OP\_PROF1 is sent by the LPAd | The delete Notification as defined below is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL1)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA | RQ35\_015 RQ35\_008 |
| 3 | S\_EndUser→ LPAd | Request for List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL1 is not shown. | RQ32\_058 |
| NOTE: The timeout in Step 2 SHALL start after the End User Intent verification. | | | | |

##### 5.4.4.2.2 TC\_LPAd\_DeleteProfile\_Enabled\_without\_PPR

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Deleting Enabled Profile, No PPRs, With PIN

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL5 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_TEST\_DP\_ADDRESS1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL5 is in Enabled state. |
| eUICC | The PROFILE\_OPERATIONAL2 is in Disabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Delete Profile procedure for PROFILE\_OPERATIONAL5 | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. | RQ32\_044 RQ32\_047 |
| 2 | LPAd → S\_SM-DP+ | Send Disable Notification containing #ICCID\_OP\_PROF5 | The disable Notification as defined below is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS5)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  See NOTE | RQ35\_008 RQ35\_015 |
| 3 | LPAd → S\_SM-DP+ | Send Delete Notification containing #ICCID\_OP\_PROF5 | The delete Notification as defined below is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL5)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  See NOTE | RQ35\_008 RQ35\_015 RQ35\_018 |
| 4 | S\_EndUser→ LPAd | Request for List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL5 is not shown. | RQ32\_058 |
| 5 | S\_EndUser→ Device | Power off then power on the Device | During Device boot up no PIN entry is requested from the End User. | RQ32\_051 |
| NOTE: The timeout SHALL start after the End User Intent verification. | | | | |

Test Sequence #02 Nominal: Deleting Enabled Profile, No PPRs, No PIN

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Delete Profile procedure for PROFILE\_OPERATIONAL1 | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. | RQ32\_044 RQ32\_047 |
| 2 | LPAd → S\_SM-DP+ | Send Disable Notification containing #ICCID\_OP\_PROF1 | The disable Notification as defined below is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS1)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  See NOTE | RQ35\_008 RQ35\_015 |
| 3 | LPAd → S\_SM-DP+ | Send Delete Notification containing #ICCID\_OP\_PROF1 | The delete Notification as defined below is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL1)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  See NOTE | RQ35\_008 RQ35\_015 RQ35\_018 |
| 4 | S\_EndUser→ LPAd | Request for List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL1 is not shown. | RQ32\_058 |
| NOTE: The timeout SHALL start after the End User Intent verification. | | | | |

##### 5.4.4.2.3 TC\_LPAd\_DeleteProfile\_Error\_with\_PPR1

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Deleting Enabled Profile, PPR1 set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed for #MCC\_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required). |
| eUICC | The PROFILE\_OPERATIONAL4 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 is in Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Delete Profile procedure is initiated for PROFILE\_OPERATIONAL4 | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified  See NOTE 1 and NOTE 2  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User.  See NOTE 1 and NOTE 2 | RQ32\_044 RQ32\_047 RQ32\_050 |
| 2 | S\_EndUser→ LPAd | Request for List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL4 is shown in Enabled state. | RQ32\_058 |
| NOTE 1: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure.  NOTE 2: The LPAd MAY display an error indicating that the deletion of the Profile is failed. | | | | |

##### 5.4.4.2.4 TC\_LPAd\_DeleteProfile\_Error\_Disabled\_with\_PPR2

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Deleting Disabled Profile, PPR2 set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR2 is allowed for #MCC\_MNC2 with gid1 and gid2 absent (with End User Consent either required or not required). |
| eUICC | The PROFILE\_OPERATIONAL7 is installed on the eUICC (see NOTE). |
| eUICC | The PROFILE\_OPERATIONAL7 is in Disabled state. |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_NOTIF as helper profile | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Delete Profile procedure is initiated for PROFILE\_OPERATIONAL7 | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  See NOTE 1 and NOTE 2  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User.  See NOTE 1 and NOTE 2 | RQ32\_044 RQ32\_047 RQ32\_050 |
| 2 | S\_EndUser→ LPAd | Request for List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL7 is shown in Disabled state. | RQ32\_058 |
| NOTE 1: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure.  NOTE 2: The LPAd MAY display an error indicating that the deletion of the Profile is failed. | | | | |

##### 5.4.4.2.5 TC\_LPAd\_DeleteProfile\_Error\_Enabled\_with\_PPR2

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Deleting Enabled Profile, PPR2 set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR2 is allowed for #MCC\_MNC2 with gid1 and gid2 absent (with End User Consent either required or not required). |
| eUICC | The PROFILE\_OPERATIONAL8 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL8 is in Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate Delete Profile procedure for PROFILE\_OPERATIONAL8 | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  See NOTE 2 and NOTE 3  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User.  See NOTE 2 and NOTE 3 | RQ32\_044 RQ32\_047 RQ32\_050 |
| 2 | LPAd → S\_SM-DP+ | Send Disable Notification containing #ICCID\_OP\_PROF8 | The disable Notification as defined below is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  MTD\_HANDLE\_NOTIF (#PENDING\_NOTIF\_DIS8)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  See NOTE 1 | RQ35\_008 RQ35\_015 |
| 3 | S\_EndUser→ LPAd | Request for List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL8 is shown in Disabled state. | RQ32\_058 |
| 4 | S\_EndUser→ Device | Power off then power on the Device | During Device boot up no PIN entry is requested from the End User. | RQ32\_051 |
| NOTE 1: The timeout SHALL start after the End User Intent verification.  NOTE 2: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure.  NOTE 3: The LPAd MAY display an error indicating that the deletion of the Profile is failed. | | | | |

##### 5.4.4.2.6 TC\_LPAd\_DeleteProfile\_Security\_Errors

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Stop Delete Profile Operation if No Confirmation Provided

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC (see NOTE). |
| eUICC | The PROFILE\_OPERATIONAL1 is in Disabled state. |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_NOTIF as helper profile | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Delete Profile procedure is initiated for PROFILE\_OPERATIONAL1.  The End User SHALL not provide Confirmation. | The LPAd stops the Delete Profile procedure. | RQ32\_001 RQ32\_002 RQ32\_004 |
| 2 | S\_EndUser→ LPAd | Request for List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL1 is shown in Disabled state. | RQ32\_004 |

### 5.4.5 Local Profile Management - Enable Profile

#### 5.4.5.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ32\_001, RQ32\_002, RQ32\_004, RQ32\_006, RQ32\_007, RQ32\_008, RQ32\_011, RQ32\_012, RQ32\_014, RQ32\_019\_1, RQ32\_053

 RQ35\_008, RQ35\_012, RQ35\_014\_1, RQ35\_014\_3, RQ35\_018, RQ35\_019

#### 5.4.5.2 Test Cases

##### 5.4.5.2.1 TC\_LPAd\_EnableProfile

|  |  |
| --- | --- |
| General **Initial Conditions** | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The End User gets presented a list of installed (operational) Profiles with their current state. |

Test Sequence #01 Nominal: Enable a formerly disabled Profile, With PIN

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL5 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL5 is in Disabled state (see NOTE). |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_NOTIF as helper profile, deleting the helper profile before the start of the actual test sequence. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Enable Profile operation for PROFILE\_OPERATIONAL5 | PROFILE\_OPERATIONAL5 is enabled | RQ32\_001 RQ32\_002 RQ32\_006 RQ32\_007 |
| 2 | LPAd → S\_SM-DP+ | Send the Enable Notification containing #ICCID\_OP\_PROF5 | The Enable Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_EN5) is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 |
| 3 | S\_EndUser → Device | Enter #PO1\_PIN1 to authenticate the user | Successful End User authentication for the selected application | RQ32\_19\_1 |
| 4 | S\_EndUser → LPAd | Request List Profiles | PROFILE\_OPERATIONAL5 is shown in Enabled state. | RQ32\_058 |
| NOTE: The timeout SHALL start after the initiation of the Enable Profile operation. | | | | |

Test Sequence #02 Nominal: Enable a formerly disabled Profile, no PIN

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Disabled state. |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_NOTIF as helper profile, deleting the helper profile before the start of the actual test sequence. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Enable Profile operation for PROFILE\_OPERATIONAL1 | PROFILE\_OPERATIONAL1 is enabled | RQ32\_001 RQ32\_002 RQ32\_006 RQ32\_007 |
| 2 | LPAd → S\_SM-DP+ | Send the Enable Notification containing #ICCID\_OP\_PROF1 | The Enable Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_EN1) is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 |
| 3 | S\_EndUser → LPAd | Request List Profiles | PROFILE\_OPERATIONAL1 is shown in Enabled state. | RQ32\_058 |
| NOTE: The timeout SHALL start after the initiation of the Enable Profile operation. | | | | |

##### 5.4.5.2.2 TC\_LPAd\_EnableProfile\_ImplicitDisable

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Enable a Profile with implicit disabling of the formerly enabled Profile, With PIN

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL5 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL6 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL5 is in Enabled state. |
| eUICC | The PROFILE\_OPERATIONAL6 is in Disabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Enable Profile operation for PROFILE\_OPERATIONAL6 |  | RQ32\_006 RQ32\_007 |
| 2 | LPAd → S\_SM-DP+(1) | Disable Notification containing #ICCID\_OP\_PROF5 is sent by the LPAd | The Disable Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS5) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS1) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 |
| 3 | LPAd → S\_SM-DP+(2) | Send the Enable Notification containing #ICCID\_OP\_PROF6 | The Enable Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_EN6) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS2) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 |
| 4 | S\_EndUser → Device | Enter #PO2\_PIN1 to authenticate the user | Successful End User authentication for the selected application | RQ32\_19\_1 |
| 5 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format.  PROFILE\_OPERATIONAL6 is shown in Enabled state. | RQ32\_058 |
| NOTE 1: The Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel.  NOTE 2: The timeout (steps 2 and 3) SHALL start after the initiation of the Enable Profile operation. | | | | |

Test Sequence #02 Nominal: Enable a Profile with implicit disabling of the formerly enabled Profile, no PIN

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL2 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Enabled state. |
| eUICC | The PROFILE\_OPERATIONAL2 is in Disabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Enable Profile operation for PROFILE\_OPERATIONAL2 |  | RQ32\_006 RQ32\_007 |
| 2 | LPAd → S\_SM-DP+(1) | Disable Notification containing #ICCID\_OP\_PROF1 is sent by the LPAd | The Disable Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS1) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS1) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 |
| 3 | LPAd → S\_SM-DP+(2) | Send the Enable Notification containing #ICCID\_OP\_PROF2 | The Enable Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_EN2) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS2) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 |
| 4 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format.  PROFILE\_OPERATIONAL2 is shown in Enabled state. | RQ32\_058 |
| NOTE 1: The Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel.  NOTE 2: The timeout (steps 2 and 3) SHALL start after the initiation of the Enable Profile operation. | | | | |

##### 5.4.5.2.3 TC\_LPAd\_EnableProfile\_Error\_ProfileAlreadyEnabled

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Enable an already enabled Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Enable Profile operation for PROFILE\_OPERATIONAL1 |  | RQ32\_006 RQ32\_007 |
| 2 | LPAd → S\_EndUser | Result of the Profile enabling | Enable Profile procedure terminates indicating an error | RQ32\_012 |

##### 5.4.5.2.4 TC\_LPAd\_EnableProfile\_Error\_PPR1Set

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled |

Test Sequence #01 Error: Enabled Profile when a formerly enabled Profile has set PPR1

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed for #MCC\_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required). |
| eUICC | The PROFILE\_OPERATIONAL4 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 is in Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | S\_EndUser → LPAd | Install PROFILE\_OPERATIONAL1 and ensure that it is disabled (see NOTE 1) | PROFILE\_OPERATIONAL1 is installed and disabled | IC1 |
| 1 | S\_EndUser → LPAd | Initiate the Enable Profile operation for PROFILE\_OPERATIONAL1 | See NOTE 2 and NOTE 3 | RQ32\_006 RQ32\_007 RQ32\_008 RQ32\_014 |
| 2 | S\_EndUser 🡪 LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format.  PROFILE\_OPERATIONAL4 is shown in Enabled state. | RQ32\_058 |
| NOTE 1: If the device supports only O\_D\_ADD\_ENABLE\_COMBINED, any attempt to automatically enable the profile is expected to fail.  NOTE 2:The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure.  NOTE 3:The LPAd MAY display an error indicating that the enabling of the Profile is failed. | | | | |

##### 5.4.5.2.5 Void

### 5.4.6 Local Profile Management- Disable Profile

#### 5.4.6.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ32\_001, RQ32\_002, RQ32\_004, RQ32\_025, RQ32\_026, RQ32\_028, RQ32\_032, RQ32\_034, RQ32\_038, RQ32\_053

 RQ35\_008, RQ35\_018, RQ35\_019

#### 5.4.6.2 Test Cases

##### 5.4.6.2.1 TC\_LPAd\_DisableProfile

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Disable an Enabled Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Enabled state. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Disable Profile operation for PROFILE\_OPERATIONAL1 | PROFILE\_OPERATIONAL1 is disabled | RQ32\_001 RQ32\_002 RQ32\_025 RQ32\_026 |
| 2 | LPAd → S\_SM-DP+ | Send the Disable Notification containing #ICCID\_OP\_PROF1 | The Disable Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DIS1) is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 RQ35\_018 RQ35\_019 |
| 3 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profile(s) with their current states are displayed in a human readable format. PROFILE\_OPERATIONAL1 is shown in Disabled state. | RQ32\_038 RQ32\_053 |
| NOTE: The timeout SHALL start after the initiation of the Disable Profile operation. | | | | |

##### 5.4.6.2.2 TC\_LPAd\_DisableProfile\_Error\_ProfileAlreadyDisabled

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Disable an already disabled Profile

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Disabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser→ LPAd | Initiate the Disable Profile operation for PROFILE\_OPERATIONAL1 |  | RQ32\_025 RQ32\_026 |
| 2 | LPAd → S\_EndUser | Result of the Profile disabling | The Disable Profile procedure terminates indicating a failure | RQ32\_034 |

##### 5.4.6.2.3 TC\_LPAd\_DisableProfile\_Error\_PPR1Set

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Error: Disable an Enabled Profile with PPR1 set

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed for #MCC\_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required). |
| eUICC | The PROFILE\_OPERATIONAL4 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL4 is in Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the Disable Profile operation for PROFILE\_OPERATIONAL4 | See NOTE 1 and NOTE 2 | RQ32\_025 RQ32\_026 RQ32\_028 RQ32\_034 |
| 2 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format  PROFILE\_OPERATIONAL4 is shown in Enabled state | RQ32\_053 |
| NOTE 1: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure.  NOTE 2: The LPAd MAY display an error indicating that the disabling of the Profile is failed. | | | | |

##### 5.4.6.2.4 VOID

### 5.4.7 Local eUICC Management - Retrieve EID Process

#### 5.4.7.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ33\_001, RQ33\_002, RQ33\_003, RQ33\_004

#### 5.4.7.2 Test Cases

##### 5.4.7.2.1 TC\_LPAd\_RetrieveEID

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Retrieve EID

The purpose of this test is to check if the Device is capable to display the stored EID in as QR code or in text string format.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Request to display EID. (See NOTE) | EID is displayed. | RQ33\_001 RQ33\_002 |
| 2 | LPAd → S\_EndUser | Presentation of the EID | The LPA presents the #EID1 to the End User as a text string and/or as a QR code.  If the EID is represented as text string, the text SHALL be identical to #EID1  If the #EID1 is shown as a QR code it SHALL be either #EID1\_QR\_CODE1 or #EID1\_QR\_CODE2 with or without blank spaces. | RQ33\_003 RQ33\_004 RQ33\_005 RQ33\_005\_1 |
| NOTE: LPAd may display the EID by default. | | | | |

### 5.4.8 Local eUICC Management - eUICC Memory Reset Process

#### 5.4.8.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ32\_053

 RQ33\_012, RQ33\_021\_1, RQ33\_021\_2

 RQ35\_008, RQ35\_018, RQ35\_019

#### 5.4.8.2 Test Cases

##### 5.4.8.2.1 TC\_LPAd\_eUICCMemoryReset

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | No proactive session is ongoing.  NOTE: These test cases MAY fail due to the fact that a proactive is ongoing but it is impossible to determine that this is the case. In this instance it is recommended to repeat the test. |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: eUICC Memory Reset, Operational Profile installed, no Operational Profile enabled

The purpose of this test is to check the basic functions of the eUICC Memory Reset. An installed but not enabled Operational Profile SHALL be deleted.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Disabled state (see NOTE). |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL2 with #METADATA\_OP\_PROF2\_NO\_NOTIF as helper profile, deleting the helper profile before the start of the actual test sequence. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the eUICC Memory Reset for operational profiles | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. |  |
| 2 | LPAd → S\_SM-DP+ | Delete Notification containing #ICCID\_OP\_PROF1 is sent by the LPAd | The Delete Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL1) is received by the S\_SM-DP+ within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  The S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 RQ35\_018 RQ35\_019 |
| 3 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format  No Operational Profile is available | RQ32\_053 RQ33\_012 |
| NOTE: The timeout (step 2) SHALL start after the End User Intent verification. | | | | |

Test Sequence #02 Nominal: eUICC Memory Reset, Operational Profile with PPR2 installed, no Operational Profile enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed but not enabled Operational Profile with PPR2 ('Deletion of this Profile is not allowed').

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR2 is allowed for #MCC\_MNC2 with gid1 and gid2 absent (with End User Consent either required, or not required). |
| eUICC | The PROFILE\_OPERATIONAL2 is installed on the eUICC with #METADATA\_OP\_PROF2\_MEMRES1. |
| eUICC | The PROFILE\_OPERATIONAL2 is in Disabled state (see NOTE). |
| NOTE: If neither O\_D\_DISABLE\_SEPARATED nor O\_D\_ADD\_ENABLE\_SEPARATED are supported, follow the rule in section 2.2.4.1 using PROFILE\_OPERATIONAL1 with #METADATA\_OP\_PROF1\_NO\_NOTIF as helper profile, deleting the helper profile before the start of the actual test sequence. | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | | S\_EndUser → LPAd | Initiate the eUICC Memory Reset for operational profiles | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. |  |
| 2 | LPAd → S\_SM-DP+ | Delete Notification containing #ICCID\_OP\_PROF2 is sent by the LPAd | The Delete Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL2) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS2) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  The S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 RQ35\_018 RQ35\_019 |
| 3 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format  No Operational Profile is available | RQ32\_053 RQ33\_012 |
| NOTE: The timeout (step 2) SHALL start after the End User Intent verification. | | | | |

Test Sequence #03 Nominal: eUICC Memory Reset, Operational Profile with PPR2 installed and enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed and enabled Operational Profile with PPR2 ‘'Deletion of this Profile is not allowed’').

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR2 is allowed for #MCC\_MNC2 with gid1 and gid2 absent (with End User Consent either required, or not required). |
| eUICC | The PROFILE\_OPERATIONAL2 is installed on the eUICC with #METADATA\_OP\_PROF2\_MEMRES1. |
| eUICC | The PROFILE\_OPERATIONAL2 is in Enabled state. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| 1 | | S\_EndUser → LPAd | Initiate the eUICC Memory Reset for operational profiles | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. |  | |
| 2 | LPAd → S\_SM-DP+ | Delete Notification containing #ICCID\_OP\_PROF2 is sent by the LPAd | The Delete Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL2) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS2) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  The S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK  See NOTE 2 | RQ35\_008 RQ35\_018 RQ35\_019 | |
| 3 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format  No Operational Profile is available | RQ32\_053 RQ33\_012 | |
| NOTE 1: The timeout (step 2) SHALL start after the End User Intent verification.  NOTE 2: A Disable Notification for PROFILE\_OPERATIONAL2 MAY be sent before the Delete Notification. This notification SHALL NOT be checked. | | | | | |

Test Sequence #04 Nominal: eUICC Memory Reset, Operational Profile with PPR1 installed and enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed and enabled Operational Profile with PPR1 ‘'Disabling of this Profile is not allowed’').

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Test eUICC’s RAT is configured as follows: PPR1 is allowed for #MCC\_MNC4 with gid1 and gid2 absent (with End User Consent either required, or not required). |
| eUICC | The PROFILE\_OPERATIONAL4 is installed on the eUICC with #METADATA\_OP\_PROF4\_MEMRES1. |
| eUICC | The PROFILE\_OPERATIONAL4 is in Enabled state. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| 1 | | S\_EndUser → LPAd | Initiate the eUICC Memory Reset for operational profiles | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. |  | |
| 2 | LPAd → S\_SM-DP+ | Delete Notification containing #ICCID\_OP\_PROF4 is sent by the LPAd | The Delete Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL4) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS4) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  The S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK  See NOTE 2 | RQ35\_008 RQ35\_018 RQ35\_019 | |
| 3 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format  No Operational Profile is available | RQ32\_053 RQ33\_012 | |
| NOTE 1: The timeout (step 2) SHALL start after the End User Intent verification.  NOTE 2: A Disable Notification for PROFILE\_OPERATIONAL4 MAY be sent before the Delete Notification. This notification SHALL NOT be checked. | | | | | |

Test Sequence #05 Nominal: eUICC Memory Reset, multiple Operational Profiles are installed, an Operational Profile is enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes all Operational Profiles installed and send the required Notifications to the appropriate SM-DP+.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL1 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL1 is in Enabled state. |
| eUICC | The PROFILE\_OPERATIONAL2 is installed on the eUICC. |
| eUICC | The PROFILE\_OPERATIONAL2 is in Disabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the eUICC Memory Reset for operational profiles | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. |  |
| 2 | LPAd → S\_SM-DP+(1) | Delete Notifications containing #ICCID\_OP\_PROF1 is sent by the LPAd | The Delete Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL1) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS1) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  The S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK  See NOTE 3 | RQ35\_008 RQ35\_018 RQ35\_019 |
| 3 | LPAd → S\_SM-DP+(2) | Delete Notification containing #ICCID\_OP\_PROF2 is sent by the LPAd | The Delete Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL2) is received by the S\_SM-DP+ (configured with #TEST\_DP\_ADDRESS2) within the timeout #IUT\_LPAd\_NOTIFICATION\_TIMEOUT  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  The S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK | RQ35\_008 RQ35\_018 RQ35\_019 |
| 4 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format  No Operational Profile is available | RQ32\_053 RQ33\_012 |
| NOTE 1: The Delete Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel.  NOTE 2: The timeout (steps 2 and 3) SHALL start after the End User Intent verification.  NOTE 3: A Disable Notification for PROFILE\_OPERATIONAL1 MAY be sent before the Delete Notification. This notification SHALL NOT be checked. | | | | |

##### 5.4.8.2.2 TC\_LPAd\_eUICCMemoryResetWithPINVerification

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | No proactive session is ongoing.  NOTE: these test cases may fail due to the fact that a proactive session is ongoing but it is impossible to determine that this is the case. In this instance it is recommended to repeat the test. |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: eUICC Memory Reset, installed and enabled Operational Profile with PIN verification

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed and enabled Operational Profile with PIN verification enabled.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The PROFILE\_OPERATIONAL5 is installed on the eUICC with #METADATA\_OP\_PROF5. |
| eUICC | The PROFILE\_OPERATIONAL5 is in Enabled state. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the eUICC Memory Reset for operational profiles | For LPAd supporting SGP.22 v2.2.2 or earlier:  Request for Confirmation and successful End User Intent verified.  For LPAd supporting SGP.22 v2.3 or later:  Strong Confirmation is requested by the LPAd and confirmed by the End User. |  |
| 2 | LPAd → S\_SM-DP+ | Delete Notification containing #ICCID\_OP\_PROF5 is sent by the LPAd | The Delete Notification MTD\_HANDLE\_NOTIF(#PENDING\_NOTIF\_DEL5) is received by the S\_SM-DP+ within the timeout (#IUT\_LPAd\_NOTIFICATION\_TIMEOUT, + #IUT\_LPAd\_READY\_AFTER\_REBOOT\_TIMEOUT)  Verify the euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG> using the #PK\_EUICC\_ECDSA  The S\_SM-DP+ SHALL return #R\_HTTP\_204\_OK  See NOTE 3 | RQ35\_008 RQ35\_018 RQ35\_019 |
| 3 | Device | Power off then power on the Device  If the Device does not automatically power off and power on, the S\_EndUser SHALL power off and power on the Device. | During Device boot up no PIN entry is requested from the End User. | RQ33\_011 RQ33\_012 |
| 4 | S\_EndUser → LPAd | Request List Profiles | Installed Operational Profiles with their current states are displayed in a human readable format  No Operational Profile is available | RQ32\_053 RQ33\_012 |
| NOTE 1: The Delete Notification (step 2) can be sent at any step after having successfully initiated the eUICC Memory Reset.  NOTE 2: The timeout (step 2) SHALL start after the End User Intent verification.  NOTE 3: A Disable Notification for PROFILE\_OPERATIONAL5 MAY be sent before the Delete Notification. This notification SHALL NOT be checked. | | | | |

Test Sequence #02 Nominal: VOID

### 5.4.9 Local eUICC Management–- eUICC Test Memory Reset Process

This section is defined as FFS and not applicable for this version of test specification.

### 5.4.10 Local eUICC Management – Set/Edit Default SM-DP+ Address Process

#### 5.4.10.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ33\_021\_2, RQ33\_021\_3, RQ33\_021\_5

#### 5.4.10.2 Test Cases

##### 5.4.10.2.1 TC\_LPAd\_Set/Edit Default SM-DP+ Address

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |

Test Sequence #01 Nominal: Set Default SM-DP+ Address where no Default Address has been set before

The purpose of this test is to set a default SM-DP+ address on a eUICC where no SM-DP+ default address is stored.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | No value is assigned to the Default SM-DP+ field. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the function to retrieve the configured address | The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser  The current Default SM-DP+ Address is empty respectively no Default SM-DP+ Address is shown | RQ33\_021\_2 |
| 2 | S\_EndUser → LPAd | If required, initiate the function to enter #TEST\_DP\_ADDRESS1 as the new Default SM-DP+ address or enter directly #TEST\_DP\_ADDRESS1 as the new Default SM-DP+. | For LPAd supporting SGP.22 v2.2.2 or earlier:  Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before.  (see NOTE 1) | RQ33\_021\_3 |
| 3 | S\_EndUser → LPAd | Initiate the function to retrieve the configured address | The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser  The current Default SM-DP+ Address #TEST\_DP\_ADDRESS1 is shown | RQ33\_021\_5 |
| NOTE 1: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

Test Sequence #02 Nominal: Edit the Default SM-DP+ Address and store it on the eUICC

The purpose of this test is to edit an existing default SM-DP+ address on a eUICC and to ensure that the changes are stored.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Default SM-DP+ field is set to #TEST\_DEFAULT\_DP\_ADDRESS\_1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the function to retrieve the configured address | The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser  The current Default SM-DP+ Address is #TEST\_DEFAULT\_DP\_ADDRESS\_1 | RQ33\_021\_2 |
| 2 | S\_EndUser → LPAd | If required, initiate the function to enter #TEST\_DP\_ADDRESS1 as the new Default SM-DP+ address or enter directly #TEST\_DP\_ADDRESS1 as the new Default SM-DP+. | For LPAd supporting SGP.22 v2.2.2 or earlier:  Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before.  (see NOTE1) | RQ33\_021\_3 |
| 3 | S\_EndUser → LPAd | Initiate the function to retrieve the configured address | The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser  The current Default SM-DP+ Address #TEST\_DP\_ADDRESS1 is shown | RQ33\_021\_5 |
| NOTE 1: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

Test Sequence #03 Nominal: Edit the Default SM-DP+ Address and store a Default Address with an empty value

The purpose of this test is to edit an existing Default SM-DP+ address on a eUICC and to ensure that the changes are stored even if the new Default Address value is empty.

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| eUICC | The Default SM-DP+ field is set to #TEST\_DEFAULT\_DP\_ADDRESS\_1. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_EndUser → LPAd | Initiate the function to retrieve the configured address | The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser  The current Default SM-DP+ Address is #TEST\_DEFAULT\_DP\_ADDRESS\_1 | RQ33\_021\_2 |
| 2 | S\_EndUser → LPAd | If required, initiate the function to enter “” (empty value) as the new Default SM-DP+ address or enter directly “” as the new Default SM-DP+. | For LPAd supporting SGP.22 v2.2.2 or earlier:  Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before.  (see NOTE1) | RQ33\_021\_3 |
| 3 | S\_EndUser → LPAd | Initiate the function to retrieve the configured address | The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser  The current Default SM-DP+ Address is empty respectively no Default SM-DP+ Address is shown | RQ33\_021\_5 |
| NOTE 1: The LPAd supporting SGP.22 v2.3 or later MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure. | | | | |

### 5.4.11 Device Power On – Profile Discovery

#### 5.4.11.1 Conformance Requirements

**References**

GSMA RSP Technical Specification [2]

**Requirements**

 RQ34\_18, RQ34\_020, RQ34\_021, RQ34\_023

#### 5.4.11.2 Test Cases

##### 5.4.11.2.1 TC\_LPAd\_DevicePowerOnProfileDiscovery\_SM-DP+\_address

|  |  |
| --- | --- |
| General Initial Conditions | |
| **Entity** | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The setting of the configuration parameter for Device Power-on Profile discovery is 'Enabled'. |
| Device | The Device is powered off. |

Test Sequence #01 Nominal: Power-on Profile discovery by using the default SM-DP+ Address

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| S\_SM-DP+ | There is a pending Profile download order for PROFILE\_OPERATIONAL1 linked to the EID of the eUICC. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | Power on the Device | | | |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | PROC\_ES9+\_INIT\_AUTH | | | |
| 3 | PROC\_ES9+\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> or missing MatchingID data object | | | |
| 4 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | |
| 5 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified, if not verified before. | RQ34\_023 |
| 6 | PROC\_ES9+\_HANDLE\_NOTIF | | | |
| 7 | LPAd → S\_EndUser | Initiate List Profile operation | PROFILE\_OPERATIONAL1 is displayed | RQ34\_018 RQ34\_020 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | |

##### 5.4.11.2.2 TC\_LPAd\_DevicePowerOnProfileDiscovery\_SM-DS

|  |  |
| --- | --- |
| General Initial Conditions | |
| Entity | Description of the general initial condition |
| Device | The protection of access to the LUI is disabled. |
| Device | The setting of the configuration parameter for Device Power-on Profile discovery is 'Enabled'. |
| Device | The Device is powered off. |

Test Sequence #01 Nominal: Power-on Profile discovery by using the SM-DS

|  |  |
| --- | --- |
| Initial Conditions |  |
| Entity | Description of the initial condition |
| S\_SM-DS | S\_SM-DP+ (#TEST\_DP\_ADDRESS1) performed Profile download Event Registration to the S\_SM-DS (#TEST\_ROOT\_DS\_ADDRESS) with #EVENT\_ID\_1. |
| S\_SM-DP+ | There is a pending Profile download order for #EVENT\_ID\_1 (PROFILE\_OPERATIONAL1). |
| S\_SM-DP+ | The PROFILE\_OPERATIONAL1 on the S\_SM-DP+ is in “Released” state. |
| eUICC | There is no default SM-DP+ address configured. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Expected result | | REQ |
| IC1 | Power-on the Device | | | | |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES11 | | | | |
| 2 | PROC\_ES11\_INIT\_AUTH | | | | |
| 3 | PROC\_ES11\_AUTH\_CLIENT with #MATCHING\_ID\_EMPTY as <MATCHING\_ID> or missing MatchingID data object | | | | |
| 4 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | | |
| 5 | PROC\_ES9+\_INIT\_AUTH | | | | |
| 6 | PROC\_ES9+\_AUTH\_CLIENT with #EVENT\_ID\_1 as <MATCHING\_ID> | | | | |
| 7 | PROC\_ES9+\_GET\_BPP  (see NOTE 1) | | | | |
| 8 | LPAd → S\_EndUser | Request for Confirmation, if not requested before. | End User Intent successfully verified , if not verified before. | | RQ34\_023 |
| 9 | PROC\_ES9+\_HANDLE\_NOTIF | | | | |
| 10 | LPAd → S\_EndUser | Initiate List Profile operation | | PROFILE\_OPERATIONAL1 is displayed | RQ34\_018 RQ34\_021 |
| NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S\_EndUser to postpone or reject the Profile installation. The S\_EndUser SHALL not abort the transaction. | | | | | |

# 6 End-to-End Testing

This section is defined as FFS and not applicable for this version of test specification.

# 7 External Test Specifications

Some test specifications related to the RSP ecosystem have been developed by external organisations (e.g. TCA (formerly SIMalliance)). These organisations defined their own requirements for test benches, test applicability and pass criteria.

This section lists the test specifications that relate to SGP.21 [3] and SGP.22 [2] requirements.

## 7.1 TCA eUICC Profile Package Test Specification

The TCA eUICC Profile Package: Interoperable Format Test Specification [23] SHALL be executed on the eUICC in order to check its compliance with the eUICC Profile Package Specification [4].

Test cases are applicable according to the eUICC Profile Package Specification [4] version and the additional eUICC Profile Package Specification [4] versions (if any) supported by the eUICC, in conjunction with the applicability table of the referred Test Specification [23].

The table below describes the versions of the eUICC Profile Package Specification [4] allowed depending on the SGP.22 version supported by the eUICC:

|  |  |  |
| --- | --- | --- |
| SGP.22 version | eUICC Profile Package Specification [4] versions required for the given SGP.22 version | Allowed values for #IUT\_EUICC\_ADD\_PP\_VERSIONS |
| 2.2.x | 2.1 or 2.2 or 2.3.1 | N/A |
| 2.3 | 2.1 or 2.2 or 2.3.1 | 3.1 or 3.2 or 3.3 |
| 2.4 | 2.1 or 2.2 or 2.3.1 | 3.1 or 3.2 or 3.3 |
| 2.5 | 2.1 or 2.2 or 2.3.1 | 3.1 or 3.2 or 3.3 |

Moreover, eUICC Manufacturers SHALL declare that the following options (as defined in [23]) are supported by the eUICC:

 O\_MILENAGE

 O\_TUAK\_128

 O\_JAVACARD

The successful execution of TCA test cases allows the following RSP requirements to be covered:

 RQ24\_022

 RQ24\_042

 RQ24\_043

Annex A Constants

A.1 Generic Constants

| Name | Content | |
| --- | --- | --- |
| ACTIVATION\_CODE\_1 | 1$#TEST\_DP\_ADDRESS1$#MATCHING\_ID\_1  ACTIVATION\_CODE\_1.png as defined in Annex H | |
| ACTIVATION\_CODE\_2 | 1$#TEST\_DP\_ADDRESS1$#MATCHING\_ID\_2$#S\_SM\_DP+\_OID  ACTIVATION\_CODE\_2.png as defined in Annex H | |
| ACTIVATION\_CODE\_3 | 1$#TEST\_DP\_ADDRESS1$#MATCHING\_ID\_3$$1  ACTIVATION\_CODE\_3.png as defined in Annex H | |
| ACTIVATION\_CODE\_3\_NO\_CC | 1$#TEST\_DP\_ADDRESS1$#MATCHING\_ID\_3  ACTIVATION\_CODE\_3\_NO\_CC.png as defined in Annex H | |
| ACTIVATION\_CODE\_4 | 1$#TEST\_DP\_ADDRESS1$#MATCHING\_ID\_4  ACTIVATION\_CODE\_4.png as defined in Annex H | |
| ACTIVATION\_CODE\_5 | 1$#TEST\_DP\_ADDRESS1$#MATCHING\_ID\_EMPTY  ACTIVATION\_CODE\_5.png as defined in Annex H | |
| ACTIVATION\_CODE\_INVALID\_FORMAT | 1#TEST\_DP\_ADDRESS1$#MATCHING\_ID\_1  ACTIVATION\_CODE\_INVALID\_FORMAT.png as defined in Annex H | |
| ADDITIONAL\_SMDP\_DATA\_EXCEEDED\_MAX | 0x01 02 03…76 77 78  -- additional data objects defined by the S\_SM-DP+ depending on the length of the SM-DP+ OID, to ensure that total length of dpProprietaryData is 129 bytes | |
| ADDITIONAL\_SMDP\_DATA\_MAX\_LENGTH | 0x01 02 03…75 76 77  -- additional data objects defined by the S\_SM-DP+ depending on the length of the SM-DP+ OID, to ensure that total length of dpProprietaryData is 128 bytes | |
| CHANGE\_CIPHER\_SPEC | 1 | |
| CLIENT\_CERT\_TYPE | 64. The Certificate Type requested from the client by the server in the Certificate Request message as ecdsa\_sign(64). | |
| CONFIRMATION\_CODE1 | 0102030405 | |
| CONFIRMATION\_CODE2 | ABCDEFGHIJ | |
| CTX\_PARAMS1  (CtxParams1) | ctxParamsForCommonAuthentication : {  #S\_DEVICE\_INFO  } | |
| CTX\_PARAMS1\_DEVICE\_INFO\_EXT | ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {  matchingId #MATCHING\_ID\_EMPTY,  deviceInfo #S\_DEVICE\_INFO\_EXT } | |
| CTX\_PARAMS1\_EVENT\_ID  (CtxParams1) | ctxParamsForCommonAuthentication : {  matchingId #EVENT\_ID\_1,  #S\_DEVICE\_INFO  } | |
| CTX\_PARAMS1\_EVENT\_ID\_IMEI  (CtxParams1) | ctxParamsForCommonAuthentication : {  matchingId #EVENT\_ID\_1,  #S\_DEVICE\_INFO\_IMEI  } | |
| CTX\_PARAMS1\_IMEI  (CtxParams1) | ctxParamsForCommonAuthentication : {  #S\_DEVICE\_INFO\_IMEI  } | |
| CTX\_PARAMS1\_MATCH\_ID  (CtxParams1) | ctxParamsForCommonAuthentication : {  matchingId #MATCHING\_ID\_1,  #S\_DEVICE\_INFO  } | |
| CTX\_PARAMS1\_MATCH\_ID\_DEV\_INFO  (CtxParams1) | ctxParamsForCommonAuthentication : {  matchingId <MATCHING\_ID>, -- OPTIONAL - see NOTE  #DEVICE\_INFO  }  NOTE: the matchingId field may be present (with value <MATCHING\_ID>) or may be absent. The presence or absence of matchingId may be checked in individual test cases. | |
| DEVICE\_INFO | deviceInfo {  tac ...,  deviceCapabilities {  ...  },  imei ... -- Optional  }--  Check only that the field is present and has a valid TLV asn.1 structure  NOTE: The content of deviceInfo is verified in individual test cases. | |
| DIST\_NAME\_CI'C=IT,O=RSPTEST,OU=TESTCERT,CN=GSMA Test CI' | |
| EF\_UST1 | 0x0A 2E 14 8C E7 32 04 00 00 00 00 00 00  -- NOTE: Service n°17 (GID1) and n°18 (GID2) not available | |
| EF\_UST2 | 0x0A 2E 17 8C E7 32 04 00 00 00 00 00 00  -- NOTE: Service n°17 (GID1) and n°18 (GID2) available | |
| EID1 | 0x89 04 90 32 12 34 51 23 45 12 34 56 78 90 12 35 | |
| EID1\_QR\_CODE1 | QR code which decodes as:  EID:89049032123451234512345678901235 | |
| EID1\_QR\_CODE2 | QR code which decodes as:  EID:89 04 90 32 12 34 51 23 45 12 34 56 78 90 12 35 | |
| EID2 | 0x89 04 90 32 11 23 41 23 40 12 34 56 78 90 13 75 | |
| EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1 | #CI\_PKI\_ID1, #CI\_PKI\_ID2 | |
| EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_2 | #CI\_PKI\_ID3, #CI\_PKI\_ID4 | |
| EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1 | #CI\_PKI\_ID1, #CI\_PKI\_ID2 | |
| EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_2 | #CI\_PKI\_ID3, #CI\_PKI\_ID4 | |
| EUICC\_INFO1\_8\_8\_2\_3\_1 | euiccInfo1\_8\_8\_2\_3\_1 EUICCInfo1 ::= {  svn #RSP\_SVN,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_2  } } | |
| EUICC\_INFO1\_8\_8\_3\_3\_1\_HIGHER | euiccInfo1\_8\_8\_3\_3\_1 EUICCInfo1 ::= {  svn #RSP\_SVN\_HIGHER,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1  } } | |
| EUICC\_INFO1\_8\_8\_3\_3\_1\_LOWER | euiccInfo1\_8\_8\_3\_3\_1 EUICCInfo1 ::= {  svn #RSP\_SVN\_LOWER,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1  } } | |
| EUICC\_INFO1\_8\_8\_4\_3\_7 | euiccInfo1\_8\_8\_4\_3\_7 EUICCInfo1 ::= {  svn #RSP\_SVN,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_2  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1  } } | |
| EUICC\_SIGNED1 | {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present and has a valid TLV asn.1 structure  ctxParams1 #CTX\_PARAMS1  } | |
| EVENT\_ID\_1 | 07399-BGH7E-T8779 | |
| EVENT\_ID\_2 | 07399-BGH7E-T8778 | |
| EXT\_SHA256\_RSA | TLS extension data for "supported\_signature\_algorithms" set as:  o HashAlgorithm sha256 (04) and  o SignatureAlgorithm rsa (01). | |
| FUNCTION\_CALL\_ID\_1 | 0000-0000-0000-0001 | |
| FUNCTION\_CALL\_ID\_2 | 0000-0000-0000-0002 | |
| GID1 | 0x47 53 4D 41 | |
| GID2 | 0x52 53 50 FF | |
| HOST\_ID | 0x47 53 4D 41 20 53 4D 2D 58 58  *--* NOTE: 'GSMA SM-XX' in ASCII | |
| ICCID\_OP\_PROF1 | 0x98 92 09 01 21 43 65 87 09 F5 | |
| ICCID\_OP\_PROF1\_NON\_SWAP | Value of #ICCID\_OP\_PROF1 as an ICCID type as specified in SGP.22 [2] section 5.2.1 – i.e. a String in non-swapped format consisting of 20 characters (i.e. with the padding character): 89…905F. | |
| ICCID\_OP\_PROF2 | 0x98 92 09 01 32 54 76 98 10 F9 | |
| ICCID\_OP\_PROF2\_NON\_SWAP | Value of #ICCID\_OP\_PROF2 as an ICCID type as specified in SGP.22 [2] section 5.2.1 – i.e. a String in non-swapped format consisting of 20 characters (i.e. with the padding character): 89…019F. | |
| ICCID\_OP\_PROF3 | 0x98 92 09 01 43 65 87 09 21 F5 | |
| ICCID\_OP\_PROF4 | 0x98 92 09 01 54 76 98 10 32 F9 | |
| ICCID\_OP\_PROF5 | 0x98 92 09 01 65 87 09 21 43 F5 | |
| ICCID\_OP\_PROF6 | 0x98 92 09 01 76 98 10 32 54 F9 | |
| ICCID\_OP\_PROF7 | 0x98 92 09 01 87 09 21 43 65 F5 | |
| ICCID\_OP\_PROF8 | 0x98 92 09 01 98 10 32 54 76 F9 | |
| ICCID\_OP\_PROF9 | 0x98 92 09 01 21 43 65 87 76 F5 | |
| ICCID\_OP\_PROFX | 0x98 92 09 01 43 65 87 09 FF FF | |
| ICCID\_UNKNOWN | 0x98 92 01 0A 21 43 65 87 09 F8 | |
| ICON\_JPG | ICON\_JPG.jpg as defined in Annex H | |
| ICON\_OP\_PROF1 | profile\_O1.png as defined in Annex H | |
| ICON\_OP\_PROF1\_2\_SEG | profile\_O1\_2\_SEG.png as defined in Annex H | |
| ICON\_OP\_PROF2 | profile\_O2.png as defined in Annex H | |
| ICON\_OP\_PROF3 | profile\_O3.png as defined in Annex H | |
| ICON\_OP\_PROF4 | profile\_O4.png as defined in Annex H | |
| ICON\_OP\_PROF5 | profile\_O5.png as defined in Annex H | |
| ICON\_OP\_PROF6 | profile\_O6.png as defined in Annex H | |
| ICON\_OP\_PROF7 | profile\_O7.png as defined in Annex H | |
| ICON\_OP\_PROF8 | profile\_O8.png as defined in Annex H | |
| IMSI\_OP\_PROF1 | 0x08 29 99 18 11 32 54 76 98 | |
| IMSI\_OP\_PROF2 | 0x08 29 99 28 11 32 54 76 97 | |
| IMSI\_OP\_PROF3 | 0x08 29 99 28 11 32 54 76 96 | |
| IMSI\_OP\_PROF4 | 0x08 29 99 48 43 65 87 09 21 | |
| IMSI\_OP\_PROF5 | 0x08 29 99 18 11 32 54 76 98 | |
| IMSI\_OP\_PROF6 | 0x08 29 99 28 11 32 54 76 97 | |
| IMSI\_OP\_PROF7 | 0x08 29 99 28 43 65 87 09 21 | |
| IMSI\_OP\_PROF8 | 0x08 29 99 28 43 65 87 09 21 | |
| IMSI\_OP\_PROF9 | 0x08 29 99 98 43 65 87 09 21 | |
| INSTALLED\_PROFILES | 0x00 | |
| INVALID\_KEY\_TYPE | 0x80 | |
| INVALID\_REMOTE\_OP\_ID | 8 | |
| ISD\_R\_AID | 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 01 00 | |
| KEY\_LENGTH | 0x10 | |
| KEY\_TYPE | 0x88 | |
| MATCHING\_ID\_1 | 04386-AGYFT-A74Y8-3F815 | |
| MATCHING\_ID\_2 | 04386-AGYFT-A74Y8-3F816 | |
| MATCHING\_ID\_3 | 04386-AGYFT-A74Y8-3F817 | |
| MATCHING\_ID\_4 | 04386-AGYFT-A74Y8-3F818 | |
| MCC\_MNC\_WILDCARD | 0x92 F9 EE | |
| MCC\_MNC1 | 0x92 F9 18 | |
| MCC\_MNC2 | 0x92 F9 28 | |
| MCC\_MNC4 | 0x92 F9 48 | |
| MCC\_MNC9 | 0x92 F9 98 | |
| MIN\_TLS\_CIPHER\_SUITES | The minimum TLS cipher suites proposed by the Client:  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | |
| MNO\_SCP80\_AUTH\_KEY | 0x11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 10 | |
| MNO\_SCP80\_DATA\_ENC\_KEY | 0x99 AA BB CC DD EE FF 10 11 22 33 44 55 66 77 88 | |
| MNO\_SCP80\_ENC\_KEY | 0x66 77 88 99 AA BB CC DD 11 22 33 44 55 EE FF 10 | |
| NAME\_OP\_PROF\_LONG | Operational Profile Name with long name of sixty four characters  NOTE: the exact text above SHOULD be used, as it is exactly 64 characters long. | |
| NAME\_OP\_PROF1 | Operational Profile Name 1 | |
| NAME\_OP\_PROF1\_NON\_ASCII | Operational Profile Name UTF-8 encoding: 0x4F 70 65 72 61 74 69 6F 6E 61 6C 20 50 72 6F 66 69 6C 65 20 4E 61 6D 65 20 E4 BD A0 E5 A5 BD | |
| NAME\_OP\_PROF2 | Operational Profile Name 2 | |
| NAME\_OP\_PROF3 | Operational Profile Name 3 | |
| NAME\_OP\_PROF4 | Operational Profile Name 4 | |
| NAME\_OP\_PROF5 | Operational Profile Name 5 | |
| NAME\_OP\_PROF6 | Operational Profile Name 6 | |
| NAME\_OP\_PROF7 | Operational Profile Name 7 | |
| NAME\_OP\_PROF8 | Operational Profile Name 8 | |
| NAME\_OP\_PROF9 | Operational Profile Name 9 | |
| NICKNAME1 | Nickname 1 | |
| NICKNAME2 | Nickname 2 | |
| NICKNAME3 | Nickname 3 | |
| NICKNAME4 | Nickname 4 | |
| OWNER\_OP\_PROF1 | { mccMnc #MCC\_MNC1 } | |
| OWNER\_OP\_PROF2 | { mccMnc #MCC\_MNC2 } | |
| PATH\_AUTH\_CLIENT | /gsma/rsp2/es9plus/authenticateClient | |
| PATH\_CANCEL\_ORDER | /gsma/rsp2/es2plus/cancelOrder | |
| PATH\_CANCEL\_SESSION | /gsma/rsp2/es9plus/cancelSession | |
| PATH\_CONFIRM\_ORDER | /gsma/rsp2/es2plus/confirmOrder | |
| PATH\_DELETE\_EVENT | /gsma/rsp2/es12/deleteEvent | |
| PATH\_DOWNLOAD\_ORDER | /gsma/rsp2/es2plus/downloadOrder | |
| PATH\_GET\_BPP | /gsma/rsp2/es9plus/getBoundProfilePackage | |
| PATH\_HANDLE\_NOTIF | /gsma/rsp2/es9plus/handleNotification | |
| PATH\_INITIATE\_AUTH | /gsma/rsp2/es9plus/initiateAuthentication | |
| PATH\_REGISTER\_EVENT | /gsma/rsp2/es12/registerEvent | |
| PO1\_PIN1 | 0x32 34 36 38 FF FF FF FF | |
| PO2\_PIN1 | 0x33 35 37 39 FF FF FF FF | |
| PPK\_ENC\_INV\_SIZE | 0x01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 0D 0E 0F 10 0D 0E 0F 10 | |
| PPK\_INIT\_MAC\_INV\_SIZE | 0x05 0A 04 0B 03 0C 02 0D 01 0E 00 0F 09 01 08 02 09 01 08 02 09 01 08 02 | |
| PPK\_MAC\_INV\_SIZE | 0x01 0E 00 0F 09 01 08 02 05 0A 04 0B 03 0C 02 0D 03 0C 02 0D 03 0C 02 0D | |
| PROFILE\_STATUS\_AVAILABLE | Available | |
| PROP\_TLS\_CIPHER\_SUITES | The TLS cipher suites proposed by the Client:  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256  o TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA  o TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256  o TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256 | |
| REMOTE\_OP\_ID\_INSTALL | 1 | |
| RSP\_SVN | This field is set to #IUT\_RSP\_VERSION (e.g. 2.1.0) | |
| RSP\_SVN\_H | This field is set to #IUT\_RSP\_VERSION encoded as the value part of an ASN.1 VersionType (e.g. 0x02 01 00) | |
| RSP\_SVN\_HIGHER | 100.0.0 | |
| RSP\_SVN\_LOWER | 0.0.0 | |
| RSP\_SVN\_V2\_2\_1 | 2.2.1 | |
| RSP\_SVN\_V2\_2\_2 | 2.2.2 | |
| RSP\_SVN\_V2\_3 | 2.3.0 | |
| S\_DEVICE\_CAP\_EXT | deviceCapExt DeviceCapExt ::= {  unknownServiceSupport2 }  using the following definition of DeviceCapExt:  DeviceCapExt ::= INTEGER {  unknownServiceSupport1 (0),  unknownServiceSupport2 (1) } | |
| S\_DEVICE\_INFO | deviceInfo {  tac #S\_TAC,  deviceCapabilities {  gsmSupportedRelease '050000'H,  utranSupportedRelease '080000'H,  cdma2000onexSupportedRelease '010000'H,  cdma2000hrpdSupportedRelease '010000'H,  cdma2000ehrpdSupportedRelease '020000'H,  eutranSupportedRelease '020000'H,  contactlessSupportedRelease '090000'H,  rspCrlSupportedVersion #RSP\_SVN\_H  }  } | |
| S\_DEVICE\_INFO\_EXT | deviceInfo DeviceInfo {  tac #S\_TAC,  deviceCapabilities {  gsmSupportedRelease '050000'H,  utranSupportedRelease '080000'H,  cdma2000onexSupportedRelease '010000'H,  cdma2000hrpdSupportedRelease '010000'H,  cdma2000ehrpdSupportedRelease '020000'H,  eutranEpcSupportedRelease '020000'H, contactlessSupportedRelease '090000'H,  rspCrlSupportedVersion, #RSP\_SVN\_H  nrEpcSupportedRelease '0F0000'H,  nr5gcSupportedRelease '0F0000'H,  eutran5gcSupportedRelease '0F0000'H,  -- No lpaSvn field  -- No catSupportedClasses field  -- No euiccFormFactorType field  deviceAdditionalFeatureSupport {  naiSupport '0F0000'H  },  unknownServiceSupport #S\_DEVICE\_CAP\_EXT  }  }  Note: the definition of DeviceInfo used above is equivalent to the definition in SGP.22 v2.5 (specific version of [2]) with the addition of a further field called “unknownServiceSupport” of type DeviceCapExt (see #S\_DEVICE\_CAP\_EXT) after the “deviceAdditionalFeatureSupport” field. | |
| S\_DEVICE\_INFO\_IMEI | deviceInfo {  tac #S\_TAC,  deviceCapabilities {  gsmSupportedRelease '050000'H,  utranSupportedRelease '080000'H,  cdma2000onexSupportedRelease '01000'H,  eutranSupportedRelease '020000'H  },  imei #S\_IMEI  } | |
| S\_EUICC\_CHALLENGE | 0x01 02 03 04 05 06 07 08 01 02 03 04 05 06 07 08 | |
| S\_EUICC\_CHALLENGE\_2 | 0x21 22 23 24 25 26 27 28 21 22 23 24 25 26 27 28 | |
| S\_EUICC\_INFO1 | euiccInfo1 EUICCInfo1 ::= {  svn #RSP\_SVN,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1  } } | |
| S\_EUICC\_INFO1\_V2\_2\_1 | euiccInfo1 EUICCInfo1 ::= {  svn #RSP\_SVN\_V2\_2\_1,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1  } } | |
| S\_EUICC\_INFO1\_V2\_2\_2 | euiccInfo1 EUICCInfo1 ::= {  svn #RSP\_SVN\_V2\_2\_2,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1  } } | |
| S\_EUICC\_INFO1\_V2\_3 | euiccInfo1 EUICCInfo1 ::= {  svn #RSP\_SVN\_V2\_3,  euiccCiPKIdListForVerification {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1  },  euiccCiPKIdListForSigning {  #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1  } } | |
| S\_EUICC\_INFO2 | euiccInfo2 EUICCInfo2 ::= {  profileVersion #PROFILE\_VERSION,  svn #RSP\_SVN\_H,  euiccFirmwareVer   #EUICC\_FIRMWARE\_VER,  extCardResource   #S\_EXT\_CARD\_RESOURCE,  uiccCapability #UICC\_CAPABILITY,  rspCapability #RSP\_CAPABILITY,  euiccCiPKIdListForVerification   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1},  euiccCiPKIdListForSigning   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1},  ppVersion #PP\_VERSION,  sasAcreditationNumber   #SAS\_ACREDITATION\_NUMBER } | |
| S\_EUICC\_INFO2\_EXT | euiccInfo2 EUICCInfo2 ::= {  profileVersion #PROFILE\_VERSION,  svn #RSP\_SVN\_H,  euiccFirmwareVer   #EUICC\_FIRMWARE\_VER,  extCardResource   #S\_EXT\_CARD\_RESOURCE,  uiccCapability #UICC\_CAPABILITY,  rspCapability #RSP\_CAPABILITY,  euiccCiPKIdListForVerification   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1},  euiccCiPKIdListForSigning   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1},  ppVersion #PP\_VERSION,  sasAcreditationNumber   #SAS\_ACREDITATION\_NUMBER,  unknownEuiccInfo2Ext  #S\_EUICC\_INFO2\_UNKNOWN\_EXT } | |
| S\_EUICC\_INFO2\_UICC\_EXT | euiccInfo2 EUICCInfo2 ::= {  profileVersion #PROFILE\_VERSION,  svn #RSP\_SVN\_H,  euiccFirmwareVer   #EUICC\_FIRMWARE\_VER,  extCardResource   #S\_EXT\_CARD\_RESOURCE,  uiccCapability #UICC\_CAPABILITY\_EXT,  rspCapability #RSP\_CAPABILITY,  euiccCiPKIdListForVerification   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1},  euiccCiPKIdListForSigning   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1},  ppVersion #PP\_VERSION,  sasAcreditationNumber   #SAS\_ACREDITATION\_NUMBER } | |
| S\_EUICC\_INFO2\_UNKNOWN\_EXT | euiccInfo2Ext EuiccInfo2Ext ::= {  unknownServiceSupport2 }  using the following definition of EuiccInfo2Ext:  EuiccInfo2Ext ::= [120] INTEGER { -- Tag '9F78'  unknownServiceSupport1 (0),  unknownServiceSupport2 (1) } | |
| S\_EUICC\_INFO2\_DEV\_EXT | euiccInfo2 EUICCInfo2 ::= {  profileVersion #PROFILE\_VERSION,  svn #RSP\_SVN\_H,  euiccFirmwareVer   #EUICC\_FIRMWARE\_VER,  extCardResource   #S\_EXT\_CARD\_RESOURCE,  uiccCapability #UICC\_CAPABILITY,  rspCapability #RSP\_CAPABILITY\_EXT,  euiccCiPKIdListForVerification   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1},  euiccCiPKIdListForSigning   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1},  ppVersion #PP\_VERSION,  sasAcreditationNumber   #SAS\_ACREDITATION\_NUMBER } | |
| S\_EXT\_SHA256\_ECDSA | TLS extension data for "supported\_signature\_algorithms" set as:  o HashAlgorithm sha256 (04) and  o SignatureAlgorithm ecdsa (03). | |
| S\_IMEI | 0x00 00 00 00 11 11 11 11 | |
| S\_MNO\_F\_REQ\_ID | “S\_MNO” | |
| S\_SAH\_SHA256\_ECDSA | Signature And Hash Algorithm extension sent in the CertificateRequest message set as:  o HashAlgorithm sha256 (04) and  o SignatureAlgorithm ecsda(3). | |
| S\_SESSION\_ID\_EMPTY | Empty TLS session ID to identify a new session, with the Length set as ‘zero’. | |
| S\_SM\_DP+\_F\_REQ\_ID | “S\_SM\_DP\_PLUS” | |
| S\_SM\_DP+\_OID | 2.999.10 | |
| S\_SM\_DP+\_OID2 | 2.999.12 | |
| S\_SM\_DP+\_OID4 | 2.999.14 | |
| S\_SM\_DP+\_OID8 | 2.999.18 | |
| S\_SM\_DS\_F\_REQ\_ID | “S\_SM\_DS” | |
| S\_SM\_DS\_OID | 2.999.15 | |
| S\_TAC | 0x00 00 00 00 | |
| S\_TLS\_CIPHER\_SUITE | TLS cipher suite selected as follows:  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256  if present in <TLS\_CIPHER\_SUITES>, otherwise  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 | |
| SERVER\_ADDRESS | FQDN of the SERVER Under Test which can be one of the following depending on the entity under test:   #IUT\_SM\_DP\_ADDRESS   #IUT\_SM\_DS\_ADDRESS\_ES11 | |
| SIMA\_RESULT\_OK | simaresp EUICCResponse ::= {  peStatus {  {status ok}  }  } | |
| SP\_NAME\_LONG | SP Name as thirty two characters  NOTE: the exact text above SHOULD be used, as it is exactly 32 characters long. | |
| SP\_NAME\_NON\_ASCII | SP Name UTF-8 encoding: 0x53 50 20 4E 61 6D 65 20 E3 83 AB | |
| SP\_NAME1 | SP Name 1 | |
| SP\_NAME2 | SP Name 2 | |
| SP\_NAME3 | SP Name 3 | |
| SP\_NAME4 | SP Name 4 | |
| SP\_NAME8 | SP Name 8 | |
| SP\_NAME9 | SP Name 9 | |
| SSD\_AID | 0xA0 00 00 05 59 10 10 01 02 73 64 56 61 6C 75 65 | |
| TEST\_ALT\_DS\_ADDRESS | testaltsmds.example.com | |
| TEST\_DEFAULT\_DP\_ADDRESS\_1 | testdefaultsmdpplus1.example.com | |
| TEST\_DP\_ADDRESS1 | testsmdpplus1.example.com | |
| TEST\_DP\_ADDRESS2 | testsmdpplus2.example.com | |
| TEST\_DP\_ADDRESS3 | testsmdpplus3.example.com | |
| TEST\_DP\_ADDRESS4 | testsmdpplus4.example.com | |
| TEST\_DP\_ADDRESS8 | testsmdpplus8.example.com | |
| TEST\_DS\_ADDRESS1 | testsmds1.example.com | |
| TEST\_ROOT\_DS\_ADDRESS | testrootsmds.example.com | |
| TLS\_VERSION\_1\_1 | 1.1 | |
| TLS\_VERSION\_1\_2 | 1.2  The minimum TLS Version supported by the Server. | |
| UNKNOWN\_BPP\_SEGMENT | 0xC9 05 01 02 03 04 05 | |
| UNKNOWN\_SERVER\_ADDRESS | unknownserver.example.com | |
| UNSUP\_TLS\_CIPHER\_SUITES | The TLS cipher suites proposed by the Client:  o TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA  o TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256 | |
| UPP\_OP\_PROF1 | The Unprotected Profile Package related to the PROFILE\_OPERATIONAL1 (see Annex E). | |
| UPP\_OP\_PROF2 | The Unprotected Profile Package related to the PROFILE\_OPERATIONAL2 (see Annex E). | |
| UPP\_OP\_PROF3 | The Unprotected Profile Package related to the PROFILE\_OPERATIONAL3 (see Annex E). | |
| UPP\_OP\_PROF4 | The Unprotected Profile Package related to the PROFILE\_OPERATIONAL4 (see Annex E). | |
| UPP\_OP\_PROF9 | The Unprotected Profile Package related to the PROFILE\_OPERATIONAL9 (see Annex E). | |
| USIM\_AID | 0xA0 00 00 00 87 10 02 FF 33 FF 01 89 00 00 01 00 | |
| VENDOR\_SPECIFIC\_EXTENSION1 | VendorSpecificExtension : {  {  vendorOid 2.999.16,  vendorSpecificData ‘C1020304’  }  } | |
| VENDOR\_SPECIFIC\_EXTENSION2 | VendorSpecificExtension : {  {  vendorOid 2.999.17,  vendorSpecificData ‘02020202’  }  } | |

A.2 Test Certificates and Test Keys

All ECC certificates and keys described below are based on either:

 NIST P-256 curve, defined in Digital Signature Standard [11]

 brainpoolP256r1 curve, defined in RFC 5639 [8]

 FRP256V1 curve, defined in ANSSI ECC [9]

NOTE: SGP.26 [25] contains test keys, valid test certificates and instructions for how to generate invalid certificates. Unless specified differently, the test keys and test certificates used in the present document are bundled with SGP.26 [25].

|  |  |
| --- | --- |
| Name | Description |
| CERT\_CI\_ECDSA | Certificate of the CI for its Public ECDSA Key |
| CERT\_CLIENT\_TLS | CERT.CLIENT.TLS certificate of the Client under test, based on NIST or Brainpool for this version of the specification, where the Certificate MAY be one of the following depending on the type of Server and whether it is a Client under test or a Client Simulator:   #CERT\_SM\_DP\_TLS   #CERT\_SM\_DS\_TLS   #CERT\_S\_SM\_DP\_TLS   #CERT\_S\_SM\_DS\_TLS  **•** #CERT\_S\_OPERATOR\_TLS |
| CERT\_EUICC\_ECDSA | Certificate of the eUICC for its Public ECDSA key  CERT.EUICC.ECDSA in the X.509 format signed by the EUM with SK.EUM.ECDSA |
| CERT\_EUICC\_ECDSA\_EID2 | Certificate of the eUICC for its Public ECDSA key (CERT.EUICC.ECDSA) in the X509 format signed by the EUM with SK.EUM.ECDSA with the subject field value serialNumber set as #EID2.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUICC\_ECDSA\_EXPIRED | RSP Certificate of the eUICC (CERT.EUICC.ECDSA) set as a fixed test CERT with 13th January 2016 set in the validity field.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUICC\_ECDSA\_INVALID\_EX\_CP | RSP Certificate of the eUICC (CERT.EUICC.ECDSA) set as a fixed test CERT with an invalid Certificate Policies extension field OID extnValue set as “id-rspRole-ci”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUICC\_ECDSA\_INVALID\_EX\_KU | RSP Certificate of the eUICC (CERT.EUICC.ECDSA) set as a fixed test CERT with an invalid Key Usage extension field extnValue set as “dataEncipherment”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUICC\_ECDSA\_INVALID\_SIG | RSP Certificate of the eUICC (CERT.EUICC.ECDSA) set as a fixed test CERT with an invalid signature in the signatureValue field.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUICC\_ECDSA\_INVALID\_SUB\_ORG | RSP Certificate of the eUICC (CERT.EUICC.ECDSA) set as a fixed test CERT with an invalid 'organization' attribute value in the subject field set as “ERRORNAME”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUICC\_ECDSA\_INVALID\_SUB\_SN | RSP Certificate of the eUICC (CERT.EUICC.ECDSA) set as a fixed test CERT with an invalid 'serialNumber' attribute value (starting with incorrect IIN) in the subject field set as “89299000112341234012345678901353”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUM\_ECDSA | Certificate of the EUM for its Public ECDSA key  CERT.EUM.ECDSA in the X.509 format signed by the requested CI with SK.CI.ECDSA. |
| CERT\_EUM\_ECDSA\_EXPIRED | RSP Certificate of the eUICC (CERT.EUM.ECDSA) set as a fixed test CERT with 13th January 2016 set in the validity field.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUM\_ECDSA\_INVALID\_EX\_BC\_cA | RSP Certificate of the EUM (CERT.EUM.ECDSA) set as a fixed test CERT with an invalid Basic Constraints extension field set as “cA = false”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUM\_ECDSA\_INVALID\_EX\_BC\_PLC | RSP Certificate of the EUM (CERT.EUM.ECDSA) set as a fixed test CERT with an invalid Basic Constraints extension field set as “pathLenConstraint = 1”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUM\_ECDSA\_INVALID\_EX\_CP | RSP Certificate of the EUM (CERT.EUM.ECDSA) set as a fixed test CERT with an invalid Certificate Policies extension field OID extnValue set as “id-rspRole-ci”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUM\_ECDSA\_INVALID\_EX\_KU | RSP Certificate of the EUM (CERT.EUM.ECDSA) set as a fixed test CERT with an invalid Key Usage extension field extnValue set as “dataEncipherment”.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUM\_ECDSA\_INVALID\_SIG | RSP Certificate of the EUM (CERT.EUM.ECDSA) set as a fixed test CERT with an invalid signature in the signatureValue field.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_EUM\_ECDSA\_UNKNOWN | RSP Certificate of the EUM (CERT.EUM.ECDSA) set as a fixed test CERT with the Authority Key Identity not trusted by the SM-DP+ as it is not found in #EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1 or #EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1.  Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_S\_CLIENT\_TLS | CERT.CLIENT.TLS certificate of the S\_CLIENT, based on NIST or Brainpool for this version of the specification, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS   #CERT\_S\_SM\_DS\_TLS |
| CERT\_S\_CLIENT\_TLS\_EXPIRED | CERT.CLIENT.TLS certificate of the S\_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_EXPIRED   #CERT\_S\_SM\_DS\_TLS\_EXPIRED |
| CERT\_S\_CLIENT\_TLS\_INV\_CERT\_POL | CERT.CLIENT.TLS certificate of the S\_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_CERT\_POL   #CERT\_S\_SM\_DS\_TLS\_INV\_CERT\_POL |
| CERT\_S\_CLIENT\_TLS\_INV\_CRITICAL\_EXT | CERT.CLIENT.TLS certificate of the S\_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_CRITICAL\_EXT   #CERT\_S\_SM\_DS\_TLS\_INV\_CRITICAL\_EXT |
| CERT\_S\_CLIENT\_TLS\_INV\_EXT\_KEY\_USAGE | CERT.CLIENT.TLS certificate of the S\_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_EXT\_KEY\_USAGE   #CERT\_S\_SM\_DS\_TLS\_INV\_EXT\_KEY\_USAGE |
| CERT\_S\_CLIENT\_TLS\_INV\_KEY\_USAGE | CERT.CLIENT.TLS certificate of the S\_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_KEY\_USAGE   #CERT\_S\_SM\_DS\_TLS\_INV\_KEY\_USAGE |
| CERT\_S\_CLIENT\_TLS\_INV\_OID | CERT.CLIENT.TLS certificate of the S\_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_OID   #CERT\_S\_SM\_DS\_TLS\_INV\_OID |
| CERT\_S\_CLIENT\_TLS\_INV\_SIG | CERT.CLIENT.TLS certificate of the S\_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_SIG   #CERT\_S\_SM\_DS\_TLS\_INV\_SIG |
| CERT\_S\_OPERATOR\_TLS | Certificate of the S\_MNO in X.509 format and based on NIST or Brainpool for this version of the specification  The CERT\_S\_OPERATOR\_TLS Test certificate is not defined in SGP.26. The content of CERT\_S\_OPERATOR\_TLS is expected to be provided either by the test tool or by the SM-DP+ vendor. |
| CERT\_S\_SERVER\_TLS | CERT.SERVER.TLS certificate of the S\_SERVER, based on NIST or Brainpool for this version of the specification, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS on ES9+   #CERT\_S\_SM\_DS\_TLS on ES11 or ES12 |
| CERT\_S\_SERVER\_TLS\_EXPIRED | CERT.SERVER.TLS certificate of the S\_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_EXPIRED   #CERT\_S\_SM\_DS\_TLS\_EXPIRED |
| CERT\_S\_SERVER\_TLS\_INV\_CERT\_POL | CERT.SERVER.TLS certificate of the S\_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_CERT\_POL   #CERT\_S\_SM\_DS\_TLS\_INV\_CERT\_POL |
| CERT\_S\_SERVER\_TLS\_INV\_CRITICAL\_EXT | CERT.SERVER.TLS certificate of the S\_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_CRITICAL\_EXT   #CERT\_S\_SM\_DS\_TLS\_INV\_CRITICAL\_EXT |
| CERT\_S\_SERVER\_TLS\_INV\_EXT\_KEY\_USAGE | CERT.SERVER.TLS certificate of the S\_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_EXT\_KEY\_USAGE   #CERT\_S\_SM\_DS\_TLS\_INV\_EXT\_KEY\_USAGE |
| CERT\_S\_SERVER\_TLS\_INV\_KEY\_USAGE | CERT.SERVER.TLS certificate of the S\_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_KEY\_USAGE   #CERT\_S\_SM\_DS\_TLS\_INV\_KEY\_USAGE |
| CERT\_S\_SERVER\_TLS\_INV\_SIG | CERT.SERVER.TLS certificate of the S\_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:   #CERT\_S\_SM\_DP\_TLS\_INV\_SIG   #CERT\_S\_SM\_DS\_TLS\_INV\_SIG |
| CERT\_S\_SM\_DP\_TLS | CERT.DP.TLS certificate of the S\_SM-DP+, based on the same CI as defined in #IUT\_LPAd\_CI based on NIST for this version of the specification |
| CERT\_S\_SM\_DP2\_TLS | CERT.DP.TLS certificate of the S\_SM-DP+, based on the same CI as defined in #IUT\_LPAd\_CI based on NIST for this version of the specification. Contains different SM-DP+ hostname (FQDN) as #CERT\_S\_SM\_DP2\_TLS. |
| CERT\_S\_SM\_DP4\_TLS | CERT.DP.TLS certificate of the S\_SM-DP+, based on the same CI as defined in #IUT\_LPAd\_CI based on NIST for this version of the specification. Contains the SM-DP+ hostname (FQDN) #TEST\_DP\_ADDRESS4 and OID value #S\_SM\_DP+\_OID4. |
| CERT\_S\_SM\_DP8\_TLS | CERT.DP.TLS certificate of the S\_SM-DP+, based on the same CI as defined in #IUT\_LPAd\_CI based on NIST for this version of the specification. Contains the SM-DP+ hostname (FQDN) #TEST\_DP\_ADDRESS8 and OID value #S\_SM\_DP+\_OID8. |
| CERT\_S\_SM\_DP\_TLS\_EXPIRED | Expired CERT.DP.TLS certificate of the S\_SM-DP+ with a valid signature, correctly formatted as X.509 certificate. |
| CERT\_S\_SM\_DP\_TLS\_INV\_CERT\_POL | CERT.DP.TLS certificate of the S\_SM-DP+ with invalid 'Certificate Policies' extension (OID not set to ‘id-rspRole-dp-tls' or 'id-rspRole-ds-tls'), formatted as X.509 certificate. |
| CERT\_S\_SM\_DP\_TLS\_INV\_CRITICAL\_EXT | CERT.DP.TLS certificate of the S\_SM-DP+ with one of the critical extensions not present, formatted as X.509 certificate. |
| CERT\_S\_SM\_DP\_TLS\_INV\_CURVE | CERT.DP.TLS certificate of the S\_SM-DP+, based on the different CI as defined in #IUT\_LPAd\_CI, not based on   NIST P-256 curve, defined in Digital Signature Standard [11]   brainpoolP256r1 curve, defined in RFC 5639 [8]   FRP256V1 curve, defined in ANSSI ECC [9] |
| CERT\_S\_SM\_DP\_TLS\_INV\_EXT\_KEY\_USAGE | CERT.DP.TLS certificate of the S\_SM-DP+ with invalid 'extended key usage' extension (not set to any combination of 'id-kp-serverAuth' or 'id-kp-clientAuth'), formatted as X.509 certificate. |
| CERT\_S\_SM\_DP\_TLS\_INV\_KEY\_USAGE | CERT.DP.TLS certificate of the S\_SM-DP+ with invalid 'key usage' extension (not set to 'digitalSignature'), formatted as X.509 certificate. |
| CERT\_S\_SM\_DP\_TLS\_INV\_OID | CERT.DP.TLS certificate of the S\_SM-DP+ containing an invalid SM-DP+OID, different to #S\_SM\_DP+\_OID, correctly formatted as X.509 certificate. |
| CERT\_S\_SM\_DP\_TLS\_INV\_SIG | Invalid CERT.DP.TLS certificate of the S\_SM-DP+ with an invalid signature with the same tag and length as a valid signature, correctly formatted as X.509 certificate. |
| CERT\_S\_SM\_DPauth\_ECDSA | Certificate of the S\_SM-DP+ for its Public ECDSA key used for SM‑DP+ authentication. This certificate contains the OID #S\_SM\_DP+\_OID. |
| CERT\_S\_SM\_DP2auth\_ECDSA | Certificate of the S\_SM-DP+ for its Public ECDSA key used for SM‑DP+ authentication. This certificate contains the OID #S\_SM\_DP+\_OID2. |
| CERT\_S\_SM\_DPauth\_INV\_SIGN | Invalid certificate of the S\_SM-DP+ for its Public ECDSA key used for authentication. This certificate contains the OID #S\_SM\_DP+\_OID and contains an invalid signature (i.e. not generated with the #SK\_CI\_ECDSA but with the same tag and length as a valid signature) |
| CERT\_S\_SM\_DPauth\_INV\_CURVE | Certificate of the S\_SM-DP+ for its Public ECDSA key used for Authentication. This certificate contains the OID #S\_SM\_DP+\_OID and a public key based on a curve different from the following ones:   NIST P-256 curve, defined in Digital Signature Standard [11]   brainpoolP256r1 curve, defined in RFC 5639 [8]   FRP256V1 curve, defined in ANSSI ECC [9] |
| CERT\_S\_SM\_DSauth\_INV\_CURVE | Certificate of the S\_SM-DS for its Public ECDSA key used for Authentication. This certificate contains the OID #S\_SM\_DS\_OID and a public key based on a curve different from the following ones:   NIST P-256 curve, defined in Digital Signature Standard [11]   brainpoolP256r1 curve, defined in RFC 5639 [8]   FRP256V1 curve, defined in ANSSI ECC [9] |
| CERT\_S\_SM\_DPpb\_ECDSA | Certificate of the S\_SM-DP+ for its Public ECDSA key used for Profile Package Binding. This certificate contains the OID #S\_SM\_DP+\_OID. |
| CERT\_S\_SM\_DPpb\_INV\_SIGN | Invalid certificate of the S\_SM-DP+ for its Public ECDSA key used for Profile Package Binding. This certificate contains the OID #S\_SM\_DP+\_OID and contains an invalid signature (i.e. not generated with the #SK\_CI\_ECDSA but with the same tag and length as a valid signature) |
| CERT\_S\_SM\_DPpb\_INV\_CURVE | Certificate of the S\_SM-DP+ for its Public ECDSA key used for Profile Package Binding. This certificate contains the OID #S\_SM\_DP+\_OID and a public key based on a curve different from the following ones:   NIST P-256 curve, defined in Digital Signature Standard [11]   brainpoolP256r1 curve, defined in RFC 5639 [8]   FRP256V1 curve, defined in ANSSI ECC [9] |
| CERT\_S\_SM\_DP2pb\_ECDSA | Certificate of the S\_SM-DP+ for its Public ECDSA key used for Profile Package Binding. This certificate contains the OID #S\_SM\_DP+\_OID2. |
| CERT\_S\_SM\_DS\_TLS | CERT.DS.TLS certificate of the S\_SM-DS based on the same CI as defined in #IUT\_LPAd\_CI based on NIST or Brainpool for this version of the specification |
| CERT\_S\_SM\_DS2\_TLS | CERT.DS.TLS certificate of the S\_SM-DS based on the same CI as defined in #IUT\_LPAd\_CI based on NIST or Brainpool for this version of the specification. Contains different SM-DS hostname (FQDN) as #CERT\_S\_SM\_DS2\_TLS. |
| CERT\_S\_SM\_DS\_TLS\_EXPIRED | Expired CERT.DS.TLS certificate of the S\_SM-DS with a valid signature, correctly formatted as X.509 certificate. |
| CERT\_S\_SM\_DS\_TLS\_INV\_CERT\_POL | CERT.DS.TLS certificate of the S\_SM-DS with invalid ‘Certificate Policies’ extension (OID not set to 'id-rspRole-ds-tls'), formatted as X.509 certificate. |
| CERT\_S\_SM\_DS\_TLS\_INV\_CRITICAL\_EXT | CERT.DS.TLS certificate of the S\_SM-DS with one of the critical extensions not present, formatted as X.509 certificate. |
| CERT\_S\_SM\_DS\_TLS\_INV\_CURVE | CERT.DP.TLS certificate of the S\_SM-DP+, based on the different CI as defined in #IUT\_LPAd\_CI, not based on   NIST P-256 curve, defined in Digital Signature Standard [11]   brainpoolP256r1 curve, defined in RFC 5639 [8]   FRP256V1 curve, defined in ANSSI ECC [9] |
| CERT\_S\_SM\_DS\_TLS\_INV\_EXT\_KEY\_USAGE | CERT.DS.TLS certificate of the S\_SM-DS with invalid 'extended key usage' extension (not set to any combination of 'id-kp-serverAuth' or 'id-kp-clientAuth'), formatted as X.509 certificate. |
| CERT\_S\_SM\_DS\_TLS\_INV\_KEY\_USAGE | CERT.DP.TLS certificate of the S\_SM-DS with invalid 'key usage' extension (not set to 'digitalSignature'), formatted as X.509 certificate. |
| CERT\_S\_SM\_DS\_TLS\_INV\_OID | CERT.DS.TLS certificate of the S\_SM-DS containing an invalid SM-DS OID, different to #S\_SM\_DS\_OID, correctly formatted as X.509 certificate. |
| CERT\_S\_SM\_DS\_TLS\_INV\_SIG | Invalid CERT.DS.TLS certificate of the S\_SM\_DS with an invalid signature with the same tag and length as a valid signature, correctly formatted as X.509 certificate. |
| CERT\_S\_SM\_DSauth\_ECDSA | Certificate of the S\_SM-DS for its Public ECDSA key used for SM‑DS authentication. This certificate contains the OID #S\_SM\_DS\_OID. |
| CERT\_S\_SM\_DSauth\_INV\_SIGN | Invalid certificate of the S\_SM-DS for its Public ECDSA key used for SM‑DS authentication. This certificate contains an invalid signature, (i.e. not generated with the #SK\_CI\_ECDSA but with the same tag and length as a valid signature) |
| CERT\_SERVER\_TLS | CERT.SERVER.TLS certificate of the Server under test, based on NIST or Brainpool for this version of the specification, where the Certificate MAY be one of the following depending on the type of Server and whether it is a Server under test or a Server simulator:   #CERT\_SM\_DP\_TLS   #CERT\_SM\_DS\_TLS   #CERT\_S\_SM\_DP\_TLS   #CERT\_S\_SM\_DS\_TLS |
| CERT\_SM\_DP\_TLS | Certificate of the SM-DP+ for securing TLS, based on NIST or Brainpool for this version of the specification.  CERT.DP.TLS in X.509 format. |
| CERT\_SM\_DPauth\_ECDSA | Certificate of the SM-DP+ for its Public ECDSA key used for SM‑DP+ authentication (CERT.DPauth.ECDSA) set as a fixed test CERT.  Depending on the SM-DP+ configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.   The Authority Key Identifier is set as #CI\_PKI\_ID1 |
| CERT\_SM\_DPpb\_ECDSA | Certificate of the SM-DP+ for its Public ECDSA key used for Profile Package Binding (CERT.DPpb.ECDSA) set as a fixed test CERT.  Depending on the SM-DP+ configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| CERT\_SM\_DS\_TLS | Certificate of the SM-DS for securing TLS, based on NIST or Brainpool for this version of the specification.  CERT.DS.TLS in X.509 format. |
| CERT\_SM\_DSauth\_ECDSA | Certificate of the SM-DS for its Public ECDSA key used for SM‑DS authentication (CERT.DSauth.ECDSA) set as a fixed test CERT.  Depending on the SM-DS configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.   The Authority Key Identifier is set as #CI\_PKI\_ID1 |
| CERT\_SM\_XXauth\_ECDSA | CERT\_SM\_XXauth\_ECDSA of the server under test, where XX = DP or XX = DS depending on the entity under test:   #CERT\_SM\_DPauth\_ECDSA   #CERT\_SM\_DSauth\_ECDSA |
| CI\_PKI\_ID1 | The CI Subject Key Identifier as defined in SGP.26 [25]. |
| CI\_PKI\_ID2 | 0x21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 |
| CI\_PKI\_ID3 | 0x31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 |
| CI\_PKI\_ID4 | 0x41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 |
| CI\_PK\_ID\_INV | 0x00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 |
| PK\_CI\_ECDSA | Public Key of the CI, contained within #CERT\_CI\_ECDSA |
| PK\_EUICC\_ECDSA | Public Key of the eUICC, contained within #CERT\_EUICC\_ECDSA |
| PK\_S\_CLIENT\_TLS | Public key of CERT\_S\_CLIENT\_TLS of the S\_CLIENT, where the key MAY be one of the following depending on the role of the simulator:   #PK\_S\_SM\_DP\_TLS   #PK\_S\_SM\_DS\_TLS |
| PK\_S\_SERVER\_TLS | Public key of CERT\_S\_SERVER\_TLS of the S\_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:   #PK\_S\_SM\_DP\_TLS on ES9+   #PK\_S\_SM\_DS\_TLS on ES11 |
| PK\_S\_SM\_DP\_TLS | Public key of CERT.DP.TLS of the S\_SM-DP+. |
| PK\_S\_SM\_DPauth\_ECDSA | Public Key of the S\_SM-DP+, contained within #CERT\_S\_SM\_DPauth\_ECDSA |
| PK\_S\_SM\_DPpb\_ECDSA | Public Key of the S\_SM-DP+, contained within #CERT\_S\_SM\_DPpb\_ECDSA |
| PK\_S\_SM\_DS\_TLS | Public key of CERT\_S\_DS\_TLS of the S\_SM-DS. |
| PK\_SM\_DPauth\_ECDSA | Public Key of the SM-DP+, contained within #CERT\_SM\_DPauth\_ECDSA |
| PK\_SM\_DPpb\_ECDSA | Public Key of the SM-DP+, contained within #CERT\_SM\_DPpb\_ECDSA |
| PK\_SM\_DSauth\_ECDSA | Public Key of the SM-DS, contained within #CERT\_SM\_DSauth\_ECDSA |
| PK\_SM\_XXauth\_ECDSA | PK\_SM\_XXauth\_ECDSA of the server under test, where XX = DP or XX = DS depending on the entity under test:   #PK\_SM\_DPauth\_ECDSA   #PK\_SM\_DSauth\_ECDSA |
| SK\_CI\_ECDSA | Private Key of the CI |
| SK\_EUICC\_ECDSA | Private key of the eUICC for creating signatures |
| SK\_S\_SM\_DPauth\_ECDSA | Private Key of the of S\_SM-DP+ for creating signatures for SM-DP+ authentication |
| SK\_S\_SM\_DSauth\_ECDSA | Private Key of the of S\_SM-DS for creating signatures for SM-DS authentication |
| SK\_S\_SM\_DPpb\_ECDSA | Private key of the S\_SM-DP+ used to provide signatures for Profile binding |

Annex B Dynamic Content

| **Variable** | **Description** |
| --- | --- |
| ANY\_ADD\_PP\_VERSIONS | Any value of the content of the EUICCInfo2.additionalEuiccProfilePackageVersions field |
| ANY\_PROFILE\_VERSION | Any value of type VersionType |
| ANY\_SVN | Any value of type VersionType |
| ANY\_SW\_IN\_ERROR | Any Status Word in error (different from 0x9000) |
| BPP | Content of a Bound Profile Package to download within the eUICC. |
| BPP\_OTPK\_EUICC\_ECKA | One-time Public Key of the eUICC for ECKA used for the BPP |
| BPP\_SEG\_A0 | Bound Profile Package TLV segment containing the tag and length fields of the firstSequenceOf87 TLV plus the first 0x87 TLV containing the ConfigureISDP command |
| BPP\_SEG\_A1 | Bound Profile Package following TLV segment array, as defined in SGP.22 [2] – section 2.5.5:   * array first element containing the Tag and length fields of the sequenceOf88 TLV * array following elements containing each of the ‘88’ TLVs containing the StoreMetadata command |
| BPP\_SEG\_A2 | Bound Profile Package TLV segment containing the Tag and length fields of the secondSequenceOf87 TLV plus the first '87' TLV, containing the ReplaceSessionKeys command |
| BPP\_SEG\_A3 | Bound Profile Package following TLV segment array, as defined in SGP.22 [2] – section 2.5.5:   * array first element containing the tag and length fields of the sequenceOf86 TLV * array following elements containing each of the '86' TLVs containing the Protected Profile Package (PPP) |
| BPP\_SEG\_INIT | Bound Profile Package TLV segment containing the tag and length fields of the BoundProfilePackage TLV plus the initialiseSecureChannelRequest command |
| C\_APDUS\_SCRIPT | List of Command APDUs formatted as an expanded structure with definite length coding as defined in ETSI TS 102 226 [14]. |
| CC | SCP80 cryptographic checksum as defined in ETSI TS 102 225 [13] (8 bytes long). |
| CHANNEL\_NUMBER | The logical channel number newly opened in the eUICC. If no logical channel is opened, the value is set to 0x00 (i.e. Basic Channel). |
| CLIENT\_TLS\_EPHEM\_KEY | Client's ephemeral key and associated information. |
| CONF\_ISDP\_PROF1\_ENC | An element of firstSequenceOf87, consisting of #CONF\_ISDP\_PROF1\_SMDP protected with <S\_ENC> and <S\_MAC> and encapsulated in a TLV with tag 0x87, length <L> to a maximum size of 1020 bytes including the tag and length fields. |
| EUICC\_CANCEL\_SESSION\_SIGNATURE | euiccCancelSessionSignature is created using the SK.EUICC.ECDSA signed over euiccCancelSessionSigned coded as ASN.1 OCTET STRING. |
| EUICC\_CANCEL\_SESSION\_SIGNATURE\_INVALID | eUICC signature randomly generated and coded as an ASN.1 OCTET STRING not equal to <EUICC\_CANCEL\_SESSION\_SIGNATURE> but with the same length as a valid signature |
| EUICC\_CHALLENGE | Random eUICC challenge, coded as asn.1 OCTET STRING, 16 bytes. |
| EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING | List of CI Public Key Identifiers supported on the eUICC for signature creation, coded as ASN.1 sequence of SubjectKeyIdentifier. The CI Public Key Identifier is from the list of possible CI Public Key Identifier. This possible CI Public Key Identifiers as supported by the eUICC will be defined later on. |
| EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION | List of CI Public Key Identifiers supported on the eUICC for signature verification, coded as ASN.1 sequence of SubjectKeyIdentifier. The CI Public Key Identifier is from the list of possible CI Public Key Identifier. This possible CI Public Key Identifiers as supported by the eUICC will be defined later on. |
| EUICC\_CI\_PK\_ID\_TO\_BE\_USED | CI Public Key Identifier to be used by the eUICC for signature, coded as ASN.1 sequence of SubjectKeyIdentifier. |
| EUICC\_CS\_SIGNATURE | The eUICC cancel session signature computed using the #SK\_EUICC\_ECDSA across the EuiccCancelSessionSigned present in the CancelSessionResponse structure |
| EUICC\_RSP\_CAPABILITY | RspCapability of the eUICC, coded as ASN.1 BIT STRING |
| EUICC\_SIGN\_PIR | The eUICC signature of the Profile Installation Result (PIR). The input data used to generate the <EUICC\_SIGN\_PIR> is the profileInstallationResultData TLV. |
| EUICC\_SIGNATURE1 | The eUICC signature 1 (euiccSignature1) computed using #SK\_EUICC\_ECDSA across the euiccSigned1 present in the AuthenticateServerResponse structure, coded as ASN.1 OCTET STRING. |
| EUICC\_SIGNATURE1\_INVALID | eUICC signature randomly generated and coded as an ASN.1 OCTET STRING not equal to <EUICC\_SIGNATURE1> |
| EUICC\_SIGNATURE2 | The eUICC signature 2 (euiccSignature2) computed using the #SK\_EUICC\_ECDSA across the following data objects:   * euiccSigned2 * smdpSignature2 present in the PrepareDownloadRequest structure |
| EUICC\_SIGNATURE2\_INVALID | eUICC signature randomly generated and coded as an ASN.1 OCTET STRING not equal to <EUICC\_SIGNATURE2> |
| EVENT\_ID | An EventID value in String format, generated by the SM-DP+ during Event Record registration. |
| EVENT\_ID\_D | An EventID value in String format, generated by the Alternative SM-DS during cascaded Event Record deletion on ES15 |
| EVENT\_ID\_R | The EventID value in String format generated by the Alternative SM-DS during cascaded Event Record registration on ES15. |
| EXT\_CARD\_RESOURCE | Extended Card Resource Information according to ETSI TS 102 226 [14], coded as ASN.1 OCTET STRING. 'Number of installed application' value field is '00'. |
| EXT\_SHA256\_ECDSA | TLS extension data for "supported\_signature\_algorithms" set as a minimum of HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03). |
| FORWARDING\_INDICATOR\_ANY | Any boolean value (TRUE/FALSE) |
| FREE\_MEM\_OP\_PROF\_INSTALLED | Non-volatile memory (tag 0x82) available in the eUICC when two or more PROFILE\_OPERATIONAL are installed |
| FREE\_MEM\_OP\_PROF1\_DELETED | Non-volatile memory (tag 0x82) available in the eUICC after PROFILE\_OPERATIONAL1 deletion |
| FREE\_MEM\_OP\_PROF1\_INSTALLED | Non-volatile memory (tag 0x82) available in the eUICC when only PROFILE\_OPERATIONAL1 is installed |
| FREE\_MEMORY\_NO\_PROFILE | Non-volatile memory (tag 0x82) available in the eUICC when there is no Profile installed |
| FUNCTION\_CALL\_ID | The function call ID generated by the entity that calls the function |
| FUNCTION\_REQ\_ID | The function requester ID |
| INVALID\_SM\_DP\_OID | SM-DP+ OID (as defined in section 1.3) not equal to #IUT\_SM\_DP\_OID |
| INVALID\_TRANSACTION\_ID | A Transaction Identifier generated by the S\_SM-DP+ or the S\_SM‑DS that SHALL be different from <S\_TRANSACTION\_ID> if exists. Otherwise, a random value is generated. |
| ISD\_P\_AID | The ISD-P AID newly created in the eUICC. This AID value is in the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1]. |
| ISD\_P\_AID1 | The ISD-P AID created in the eUICC for the PROFILE\_OPERATIONAL1. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1]. |
| ISD\_P\_AID2 | The ISD-P AID created in the eUICC for the PROFILE\_OPERATIONAL2. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1]. |
| ISD\_P\_AID3 | The ISD-P AID created in the eUICC for the PROFILE\_OPERATIONAL3. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1]. |
| ISD\_P\_AID4 | The ISD-P AID created in the eUICC for the PROFILE\_OPERATIONAL4. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1]. |
| ISD\_P\_AIDX | An invalid ISD-P AID not present on the eUICC. This AID value is in the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. |
| L | Exact length of the corresponding tag or of the remaining data. |
| MATCHING\_ID | Unique identifier as defined in [2]. The content can be either empty, or the value of the EventID, or the value of the Activation Code token. |
| MATCHING\_ID\_EVENT | A Unique identifier of an Event for a specific EID generated by the SM-DP+ / SM-DS. |
| METADATA\_OP\_PROF1\_SEG | The #METADATA\_OP\_PROF1 is mac-ed with <S\_MAC> and split as necessary into segments of a maximum size of 1020 bytes (including the tag, length field, and MAC), |
| MNO\_SCP80\_COUNTER | SCP80 counter of the MNO-SD related to the KVN 0x01 (5 bytes long). Initial value is set to 0x00 00 00 00 01 and is incremented by one each time a secured packet is sent. |
| NB\_EXECUTED\_C\_APDUS | Number of executed Command TLV objects as defined in ETSI TS 102 226 [14]. |
| NOTIF\_SEQ\_NO\_DE1 | The Sequence Number of the Delete Notification related to the PROFILE\_OPERATIONAL1. |
| NOTIF\_SEQ\_NO\_DI1 | The Sequence Number of the Disable Notification related to the PROFILE\_OPERATIONAL1. |
| NOTIF\_SEQ\_NO\_EN1 | The Sequence Number of the Enable Notification related to the PROFILE\_OPERATIONAL1. |
| NOTIF\_SEQ\_NO\_EN2 | The Sequence Number of the Enable Notification related to the PROFILE\_OPERATIONAL2. |
| NOTIF\_SEQ\_NO\_IN1 | The Sequence Number of the Install Notification related to the PROFILE\_OPERATIONAL1. |
| NOTIF\_SEQ\_NO\_IN1\_PIR | The Sequence Number of the Install Notification (PIR) related to the PROFILE\_OPERATIONAL1. |
| NOTIF\_SEQ\_NO\_IN2 | The Sequence Number of the Install Notification related to the PROFILE\_OPERATIONAL2. |
| NOTIF\_SEQ\_NO\_IN2\_PIR | The Sequence Number of the Install Notification (PIR) related to the PROFILE\_OPERATIONAL2. |
| NOTIF\_SEQ\_NO2\_DE1 | The Sequence Number of the second Delete Notification related to the PROFILE\_OPERATIONAL1. |
| NOTIF\_SEQ\_NO2\_DI1 | The Sequence Number of the second Disable Notification related to the PROFILE\_OPERATIONAL1. |
| NOTIF\_SEQ\_NO2\_EN1 | The Sequence Number of the second Enable Notification related to the PROFILE\_OPERATIONAL1. |
| OT\_SK\_S\_SM\_DP+\_ECKA | One-time Private Key generated by the S\_SM-DP+ for ECKA. Depending on the eUICC configuration, this key is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| OTPK\_EUICC\_ECKA | One-time Public Key generated by the eUICC for ECKA. Depending on the eUICC configuration, this key is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| OTPK\_EUICC\_ECKA\_NEW | One-time Public Key of the eUICC for ECKA used for the BPP which is a new generated value different from <OTPK\_EUICC\_ECKA> |
| OTPK\_S\_SM\_DP+\_ECKA | One-time Public Key generated by the S\_SM-DP+ for ECKA. Depending on the eUICC configuration, this key is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| OTPK\_SM\_DP+\_ECKA | One-time Public Key generated by the SM-DP+ for ECKA. Depending on the eUICC configuration, this key is based on NIST P-256, brainpoolP256r1 or FRP256V1. |
| PPK\_ENC | Random PPK-ENC value (16 bytes key length). This value is different from <S\_ENC> value. |
| PPK\_INIT\_MAC | Random initial MAC chaining value (16 bytes). This value is different from the <S\_MAC\_CHAIN> value. |
| PPK\_MAC | Random PPK-MAC value (16 bytes key length). This value is different from <S\_MAC> value. |
| PPP\_OP\_PROF1\_SEG\_PPK | An element of sequenceOf86, consisting of a <UPP\_OP\_PROF1\_SEG> protected with <PPK\_ENC> and <PPK\_MAC> and encapsulated in a TLV with tag 0x86 length <L>, up to a maximum size of 1020 bytes including the tag and length field. |
| PPP\_OP\_PROF1\_SEG\_SK | An element of sequenceOf86, consisting of a <UPP\_OP\_PROF1\_SEG> segment protected with <S\_ENC> and <S\_MAC> and encapsulated in a TLV with tag 0x86, length <L>, up to a maximum size of 1020 bytes including the tag and length field. |
| PPP\_OP\_PROF1\_SEG\_SK\_INV | <PPP\_OP\_PROF1\_SEG\_SK> modified (wrong encryption) |
| PPR\_IDS | Forbidden Profile Policy Rules. This PPR list MAY be empty or MAY contain either PPR1 or PPR2 or both. |
| PROPRIETARY\_DATA | Proprietary Data returned by the eUICC as part of FCI template |
| R\_APDU\_PARTx | Sub-part of a R-APDU (see Annex D.4.2) |
| RANDOM\_SM\_DP+\_SIGN | Random SM-DP+ signature (i.e. content of the tag 0x5F37) with a size corresponding to a valid one. |
| RANDOM\_SM\_DS\_SIGN | Random SM-DS signature (i.e. content of the tag 0x5F37) with a size corresponding to a valid one. |
| REPLACE\_S\_KEYS\_REQ\_ENC | An element of secondSequenceOf87, consisting of #REPLACE\_S\_KEYS\_REQ protected with <S\_ENC> and <S\_MAC> and encapsulated in a TLV with tag 0x87, up to a maximum size of 1020 bytes including the tag and length field. |
| REASON\_CODE\_ANY | Any Reason Code, as defined in SGP.22 [2] – section 5.2.6.2 |
| RSP\_SERVER\_ADDRESS | RSP Server address in FQDN format where the operation corresponding to the Event can be processed. |
| S\_ENC | SCP03T Encryption Session key (128 bits length) resulting from the key agreement with eUICC. |
| S\_HASHED\_CC | Hashed Confirmation Code generated by the LPA. When generated by the S\_LPAd, the S\_LPAd SHALL use #CONFIRMATION\_CODE1 in the calculation unless otherwise specified. |
| S\_HASHED\_CC\_ERROR | A random generated hash value of the Confirmation Code not equal to S\_HASHED\_CC. |
| S\_INIT\_MAC | SCP03T Initial MAC chaining value (128 bits length) resulting from the key agreement with eUICC. |
| S\_MAC | SCP03T MACing Session key (128 bits length) resulting from the key agreement with eUICC. |
| S\_MAC\_CHAIN | Current MAC chaining value used for SCP03t BPP protection. |
| S\_SEL\_TLS\_CIPHER\_SUITE | TLS cipher suite selected by the Server set as follows:  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256  if present in <TLS\_CIPHER\_SUITES>, otherwise  o TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256. |
| S\_SESSION\_ID\_SERVER | Random value of the TLS session\_id in ServerHello which is different from <SESSION\_ID\_CLIENT>. This value is non-empty. |
| S\_SM\_DP+\_SIGN | The S\_SM-DP+ signature (smdpSign), computed using the #SK\_S\_SM\_DPpb\_ECDSA across the following data objects:   * remoteOpId * transactionId * controlRefTemplate * smdpOtpk * euiccOtpk, as provided earlier in the prepareDownloadResponse data object |
| S\_SM\_DP+\_SIGNATURE2 | The ASN.1 OCTET STRING encoded SM-DP+ signature 2 (field smdpSignature2) computed using the private key related to the server certificate (field smdpCertificate) present in the PrepareDownloadRequest structure. This signature SHALL be generated across the following data objects:   * smdpSignature2 * euiccSignature1 present in the AuthenticateServerResponse structure |
| S\_SMDP\_CHALLENGE | The SM-DP+ Challenge (serverChallenge) randomly chosen by the simulated SM-DP+ to be signed later by the eUICC for the eUICC authentication, coded as ASN.1 OCTET STRING of 16 bytes. |
| S\_SMDP\_SIGNATURE\_INV | <S\_SMDP\_SIGNATURE1> NOT computed with the #SK\_S\_SM\_DPauth\_ECDSA but with the same length as a valid signature |
| S\_SMDP\_SIGNATURE1 | The ASN.1 OCTET STRING encoded SM-DP+ signature (field serverSignature1) computed using the private key related to the server certificate (field serverCertificate) present in the AuthenticateServerRequest structure. |
| S\_SMDP\_SIGNED\_INV\_ADDR | <S\_SMDP\_SIGNED1> with a different SM-DP+ address (#TEST\_DP\_ADDRESS2 instead of #TEST\_DP\_ADDRESS1) |
| S\_SMDP\_SIGNED1  (ServerSigned1) | {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  } |
| S\_SMDS\_CHALLENGE | The SM-DS Challenge (serverChallenge) randomly chosen by the simulated SM-DS to be signed later by the eUICC for the eUICC authentication, coded as ASN.1 OCTET STRING of 16 bytes. |
| S\_SMDS\_SIGNATURE\_INV | <S\_SMDS\_SIGNATURE1> NOT computed with the #SK\_S\_SM\_DSauth\_ECDSA but with the same length as a valid signature |
| S\_SMDS\_SIGNATURE1 | The SM-DS signature 1 (serverSignature1) computed using #SK\_S\_SM\_DSauth\_ECDSA across the serverSigned1 present in the AuthenticateServerRequest structure, coded as ASN.1 OCTET STRING |
| S\_SMDS\_SIGNED\_ADDR1  (ServerSigned1) | {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DS\_ADDRESS1,  serverChallenge <S\_SMDS\_CHALLENGE>  } |
| S\_SMDS\_SIGNED\_INV\_ADDR | <S\_SMDS\_SIGNED1> with a different SM-DS address (#TEST\_DP\_ADDRESS1 instead of #TEST\_ROOT\_DS\_ADDRESS) |
| S\_SMDS\_SIGNED1  (ServerSigned1) | {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>  } |
| S\_TRANSACTION\_ID | The TransactionID (Unique Transaction Identifier) generated by the (S\_)SM-DP+, or (S\_)SM-DS which is used to uniquely identify the RSP session and to correlate the multiple ESXX request messages that belong to the same RSP session. This value (binary value) can start from 0x01 and can be increased by 1 each time a Profile is downloaded in the eUICC. 1-16 bytes (ASN.1 OCTET STRING). |
| SAH\_SHA256\_ECDSA | Signature And Hash Algorithm extension sent in the CertificateRequest message set as a minimum of:   * HashAlgorithm sha256 (04) and * SignatureAlgorithm ecdsa (03). |
| SEL\_TLS\_CIPHER\_SUITE | TLS cipher suite selected by the Server |
| SEQ\_NUMBER | Sequence Number related to a Notification Metadata generated by the eUICC. |
| SERVER\_CHALLENGE | Random value generated by the SM-XX server under test coded as ASN.1 OCTET STRING of 16 bytes which can be one of the following depending on the entity under test:   * <SMDP\_CHALLENGE> * <SMDS\_CHALLENGE> |
| SERVER\_CHALLENGE\_2 | Random value generated by the SM-XX server under test coded as ASN.1 OCTET STRING of 16 bytes which can be one of the following depending on the entity under test:   * <SMDP\_CHALLENGE\_2> * <SMDS\_CHALLENGE\_2> |
| SERVER\_FINISHED | The first protected message with the negotiated algorithms, keys, and secrets. It is the Hash of the concatenation of all the data from all messages in this handshake up to, but not including, this message i.e. all handshake messages starting at ClientHello up to, but not including, this Finished message itself.  NOTE: ChangeCipherSpec messages, alerts, and any other record type are not handshake messages and are not included in the hash computations. Also, HelloRequest messages are omitted from handshake hashes. |
| SERVER\_SIGNATURE1 | Server signature (serverSignature1) which can be one of the following depending on the entity under test:   * SM-DP+ signature (serverSignature1) generated over #SERVER\_SIGNED1 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING * SM-DS signature (serverSignature1) generated over #SERVER\_SIGNED1 using SK.DSauth.ECDSA, coded as ASN.1 OCTET STRING |
| SERVER\_SIGNATURE1\_2 | SERVER signature (serverSignature1) which can be one of the following depending on the entity under test:   * SM-DP signature (serverSignature1) generated over #SERVER\_SIGNED1\_2 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING * SM-DS signature (serverSignature1) generated over #SERVER\_SIGNED1\_2 using SK.DSauth.ECDSA, coded as ASN.1 OCTET STRING |
| SERVER\_TLS\_EPHEM\_KEY | Server's ephemeral key and associated information. |
| SESSION\_ID\_CLIENT | Random or empty value of the TLS session\_id in ClientHello. |
| SESSION\_ID\_RANDOM | Random value of the TLS session. |
| SHS | Shared Secret resulting from the key agreement with eUICC. |
| SM\_DP+\_SIGN | The SM-DP+ signature in ES8+/InitialiseSecureChannelRequest/smdpSign. |
| SMDP\_CHALLENGE | Random value generated by the SM-DP+ coded as ASN.1 OCTET STRING of 16 bytes. |
| SMDP\_CHALLENGE\_2 | Random value generated by the SM-DP+ coded as ASN.1 OCTET STRING of 16 bytes. |
| SMDP\_CHALLENGE\_INVALID | SM-DP+ Challenge randomly generated by the simulated SM-DP+ coded as ASN.1 OCTET STRING of 16 bytes not equal to <SMDP\_CHALLENGE>. |
| SMDP\_METADATA\_SEG\_MAC | An element of sequenceOf88, consisting of a segment of maximum size 1008 bytes protected with <S\_MAC> and encapsulated in a TLV with tag 0x88, length <L>, up to a maximum size of 1020 bytes including the tag and length field. |
| SMDP\_SIGNATURE2 | SM-DP+ signature (smdpSignature2) generated over smdpSigned2 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING |
| SMDS\_CHALLENGE | Random value generated by the SM-DS coded as ASN.1 OCTET STRING of 16 bytes. |
| SMDS\_CHALLENGE\_2 | Random value generated by the SM-DS coded as ASN.1 OCTET STRING of 16 bytes. |
| SMDS\_CHALLENGE\_INVALID | SM-DS Challenge randomly generated by the simulated SM-DS coded as ASN.1 OCTET STRING of 16 bytes not equal to <SMDS\_CHALLENGE>. |
| STORE\_DATA\_BLOCK\_NUM | The STORE DATA block number coded sequentially from 0x00 to 0xFF. If the value 0xFF has been reached and more STORE DATA commands are needed to complete the transfer, the numbering restarts and the next STORE DATA block number is set to 0x00. |
| SUBJECT\_CODE\_ANY | Any Subject Code, as defined in SGP.22 [2] – section 5.2.6.1 |
| TBS\_EUICC\_NOTIF\_SIG | The eUICC signature generated over tbsOtherNotification. NotificationMetadata, coded as ASN.1 OCTET STRING. |
| TLS\_CIPHER\_SUITES | TLS cipher suite list supported by LPAd or the Client (SM-DP+ or SM-DS) under test. |
| TRANSACTION\_ID\_2 | A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session as OCTET STRING of up to 16 bytes. |
| TRANSACTION\_ID\_AC | A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session used by the AuthenticateClient function as OCTET STRING of up to 16 bytes. |
| TRANSACTION\_ID\_GBPP | A unique Transaction ID generated by an SM-DP+ within the scope and lifetime of each SM-DP+ to uniquely identify the ongoing RSP session used by the GetBoundProfilePackage function as OCTET STRING of up to 16 bytes. |
| TRANSACTION\_ID\_IA | A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or an SM-DS to uniquely identify the ongoing RSP session used by the InitiateAuthentication function as OCTET STRING of up to 16 bytes. |
| TRANSACTION\_ID\_ISC | A unique Transaction ID generated by an SM-DP+ within the scope and lifetime of each SM-DP+ to uniquely identify the ongoing RSP session used by the InitialiseSecureChannelRequest function as OCTET STRING of up to 16 bytes. |
| TRANSACTION\_ID\_SIGNED | A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session as OCTET STRING of up to 16 bytes signed as part of #SERVER\_SIGNED1 |
| TRANSACTION\_ID\_SIGNED\_2 | A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session as OCTET STRING of up to 16 bytes signed as part of #SERVER\_SIGNED1 |
| TRANSACTION\_ID\_SIGNED\_AC | A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session used by the AuthenticateClient function as OCTET STRING of up to 16 bytes. |
| TRANSACTION\_ID\_SIGNED\_IA | A unique Transaction ID generated by an SM-DP+ or an SM-SD within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session used by the InitiateAuthentication function as OCTET STRING of up to 16 bytes. |
| TRE\_PROPERTIES | The value of the treProperties field in EUICCInfo2. |
| TRE\_REFERENCE | The value of the treProductReference field in EUICCInfo2. |
| UPP\_OP\_PROF1\_SEG | A segment of the #UPP\_OP\_PROF1, with a maximum size of 1007 bytes. |
| UPP\_OP\_PROF2\_SEG | A segment of the #UPP\_OP\_PROF2, with a maximum size of 1007 bytes. |

Annex C Methods And Procedures

This section describes methods and procedures used in the interfaces compliance test cases. They are part of test cases and SHALL not be executed in standalone mode.

C.1 Methods

If the method is used in the “expected result” column, all parameters SHALL be verified by the simulated entity (test tool). If the method is used in the “Sequence / Description” column, the command SHALL be generated by the simulated entity.

|  |  |
| --- | --- |
| Method | MTD\_AUTHENTICATE\_CLIENT |
| Description | Generates or verifies the JSON formatted AuthenticateClient request |
| Parameter(s) | * paramTransactionId: random 16 byte identifier encoded as String Hexadecimal. * paramAuthenticateServerResponse: server authentication response structured as ASN.1 encoded as base 64. |
| Details | JSON body  {  "transactionId" : paramTransactionId,   "authenticateServerResponse" : paramAuthenticateServerResponse } |

|  |  |
| --- | --- |
| Method | MTD\_CANCEL\_ORDER |
| Description | Sends and checks the JSON formatted CancelOrder request |
| Parameter(s) |  paramFunctionRequesterId   paramFunctionCallId   paramIccid: identification of the targeted profile (mandatory)   paramEID: EID of the targeted eUICC (conditional)   paramMatchingId: matching ID generated by the Operator (conditional)   paramProfileStatus: final Profile status indicator (mandatory) |
| Details | JSON requestHeader  {  "header" : {  "functionRequesterIdentifier" : "paramFunctionRequesterId",  "functionCallIdentifier" : "paramFunctionCallId"  }  JSON body  {  "iccid" : paramIccid  "eid" : paramEID,  "matchingId" : paramMatchingId  "profileStatus" : paramProfileStatus  }  }  Note: if some of the value of the parameters above are not provided, those parameters are not included as part of JSON body |

|  |  |
| --- | --- |
| Method | MTD\_CANCEL\_SESSION |
| Description | Sends or verifies the JSON formatted CancelSession request |
| Parameter(s) |  paramTransactionId: random 16 byte identifier.   paramCancelSessionResponse: eUICC information structured as ASN.1 encoded as base 64. |
| Details | JSON body  {  "transactionId" : paramTransactionId,  "cancelSessionResponse" : paramCancelSessionResponse  } |

|  |  |
| --- | --- |
| Method | MTD\_CHECK\_SMS\_POR |
| Description | Check the content of the SMS POR containing the response of the ES6.UpdateMetadata request |
| Parameter(s) | paramExpectedSW: the expected Status Word of the last STORE DATA command |
| Details | Parse and retrieve the SCP80 response packet from the SMS.  SCP80 response status code SHALL be equal to 0x00 – POR OK.  The additional data from the response packet SHALL be formatted as an expanded structure with definite length as defined in ETSI TS 102 226 [14] and contains the following TLV:  AB <L>  80 <L> <NB\_EXECUTED\_C\_APDUS> *-- Number of executed C-APDUs*  23 <L> 00 90 00 *–- R-APDU of the INSTALL FOR PERSONALIZATION command*  23 <L> paramExpectedSW *–- SW of the last STORE DATA command executed*    <NB\_EXECUTED\_C\_APDUS> SHALL be equal to the number of executed C-APDUs (i.e. one INSTALL FOR PERSONALIZATION + n STORE DATA command(s)) |

|  |  |
| --- | --- |
| Method | MTD\_CONFIRM\_ORDER |
| Description | Sends and checks the JSON formatted ConfirmOrder request |
| Parameter(s) |  paramFunctionRequesterId   paramFunctionCallId   paramIccid: identification of the targeted profile (mandatory)   paramEID: EID of the targeted eUICC (conditional)   paramMatchingId: matching ID generated by the Operator (optional)   paramConfirmationCode: confirmation code provided by the Operator (optional)   paramSmdsAddress: SM-DS to be used for event registration (conditional)   paramReleaseFlag: boolean indicating if the profile shall be released (mandatory) |
| Details | JSON requestHeader  {  "header" : {  "functionRequesterIdentifier" : "paramFunctionRequesterId",  "functionCallIdentifier" : "paramFunctionCallId"  }  JSON body  {  "iccid" : paramIccid  "eid" : paramEID,  "matchingId" : paramMatchingId  "confirmationCode" : paramConfirmationCode  "smdsAddress" : paramSmdsAddress "releaseFlag" : paramReleaseFlag  }  }  Note: if some of the value of the parameters above are not provided, those parameters are not included as part of JSON body |

|  |  |
| --- | --- |
| Method | MTD\_DELETE\_EVENT |
| Description | Sends and checks the JSON formatted DeleteEvent request |
| Parameter(s) |  paramFunctionRequesterId: identification of the function requester.   paramFunctionCallId: identification of the function call.   paramEID: EID of the targeted eUICC   paramEventId: unique Identification of the Event to be registered |
| Details | JSON requestHeader  {  "header" : {  "functionRequesterIdentifier" : "paramFunctionRequesterId",  "functionCallIdentifier" : "paramFunctionCallId"  }  JSON body  {  "eid" : paramEID,  "eventId" : paramEventId  }  } |

|  |  |
| --- | --- |
| Method | MTD\_DISABLE\_PROFILE |
| Description | Generate the ASN.1 DisableProfileRequest structure according to the input parameters. |
| Parameter(s) |  paramIccidValue: The ICCID of the Profile to Disable (optional)   paramIsdpAidValue: The ISD-P AID of the Profile to Disable (optional)   paramRefreshFlag: Boolean, TRUE if refreshFlagSHALL be set, FALSE otherwise  Either paramIccidValue or paramIsdpAidValue is passed as a parameter. |
| Details | IF paramIccidValue is provided Then  req DisableProfileRequest::= {  profileIdentifier iccid : paramIccidValue,  refreshFlag paramRefreshFlag  }  Else  req DisableProfileRequest::= {  profileIdentifier isdpAid : paramIsdpAidValue,  refreshFlag paramRefreshFlag  }  End if |

|  |  |
| --- | --- |
| Method | MTD\_DOWNLOAD\_ORDER |
| Description | Sends and checks the JSON formatted DownloadOrder request |
| Parameter(s) |  paramFunctionRequesterId   paramFunctionCallId   paramEID: EID of the targeted eUICC (optional)   paramIccid: identification of the targeted profile (conditional)   paramProfileType: identification of the targeted profile type (conditional) |
| Details | JSON requestHeader  {  "header" : {  "functionRequesterIdentifier" :"paramFunctionRequesterId",  "functionCallIdentifier" :"paramFunctionCallId"  }  JSON body  {  "eid" : paramEID,  "iccid" : paramIccid  "profileType" : paramProfileType  }  }  Note: if some of the value of the parameters above are not provided, those parameters are not included as part of JSON body |

|  |  |
| --- | --- |
| Method | MTD\_ENABLE\_PROFILE |
| Description | Generate the ASN.1 EnableProfileRequest structure according to the input parameters. |
| Parameter(s) |  paramIccidValue: The ICCID of the Profile to Disable (optional)   paramIsdpAidValue: The ISD-P AID of the Profile to Disable (optional)   paramRefreshFlag: Boolean, TRUE if refreshFlagSHALL be set, FALSE otherwise  Either paramIccidValue or paramIsdpAidValue is passed as a parameter. |
| Details | IF paramIccidValue is provided Then  req EnableProfileRequest ::= {  profileIdentifier iccid : paramIccidValue,  refreshFlag paramRefreshFlag  }  Else  req EnableProfileRequest ::= {  profileIdentifier isdpAid : paramIsdpAidValue,  refreshFlag paramRefreshFlag  }  End if |

|  |  |
| --- | --- |
| Method | MTD\_DELETE\_PROFILE |
| Description | Generate the ASN.1 DeleteProfileRequest structure according to the input parameters. |
| Parameter(s) |  paramIccidValue: The ICCID of the Profile to Delete (optional)   paramIsdpAidValue: The ISD-P AID of the Profile to Delete (optional)  Either paramIccidValue or paramIsdpAidValue is passed as a parameter. |
| Details | IF paramIccidValue is provided Then  req DeleteProfileRequest ::= iccid : paramIccidValue  Else  req DeleteProfileRequest ::= isdpAid : paramIsdpAidValue  End if |

|  |  |
| --- | --- |
| Method | MTD\_GET\_PROFILE\_INFO |
| Description | Generate the ASN.1 ProfileInfoListRequest according to the input parameters. |
| Parameter(s) |  paramIccidValue: The ICCID of the Profile   paramIsdpAidValue: The ISD-P AID of the Profile  Either paramIccidValue or paramIsdpAidValue is passed as a parameter. |
| Details | IF paramIccidValue is provided Then  req ProfileInfoListRequest::= {  searchCriteria iccid: paramIccidValue  }  Else  req ProfileInfoListRequest::= {  searchCriteria isdpAid: paramIsdpAidValue  }  End If |

|  |  |
| --- | --- |
| Method | MTD\_GENERATE\_BPP |
| Description | Generate a BPP according to the input parameters. |
| Parameter(s) |  paramInitSC: The InitialiseSecureChannel request   paramConfISDP: The ConfigureISDP request (plain)   paramStoreMetadata: The StoreMetadata request (plain)   paramReplaceSessionKeys: The ReplaceSessionKeys request (plain) – Optional parameter   paramUPP: The Unprotected Profile Package to download |
| Details | Split the paramStoreMetadata in several segments of maximum 1008 bytes. Each Metadata segment is named <METADATA\_SEG> here after.  Split the paramUPP in several segments of maximum 1007 bytes. Each UPP segment named <UPP\_SEG> here after.  Create the following structure of data:  req BoundProfilePackage ::= {  paramInitSC,  firstSequenceOf87 {  0x87 <L> paramConfISDP  },  sequenceOf88 {  0x88 <L> <METADATA\_SEG>,  …  0x88 <L> <METADATA\_SEG>  },  -- secondSequenceOf87 SHALL be set only if paramReplaceSessionKeys is  -- provided  secondSequenceOf87 {  0x87 <L> paramReplaceSessionKeys  },  sequenceOf86 {  0x86 <L> <UPP\_SEG>,  …  0x86 <L> <UPP\_SEG>  }  }  Use <OT\_SK\_S\_SM\_DP+\_ECKA> and <OTPK\_EUICC\_ECKA> in order to generate the <SHS>.  Concatenate #KEY\_TYPE, #KEY\_LENGTH, <L> #HOST\_ID and <L> #EID1 as SharedInfo.  Retrieve <S\_ENC>, <S\_MAC> and <S\_INIT\_MAC> across SHA-256 calculated from <SHS> and SharedInfo.  Encrypt paramConfISDP with <S\_ENC>.  Calculate and add a MAC to the tag 0x87 of firstSequenceOf87 by using <S\_MAC>.  Calculate and add a MAC to all tags 0x88 of sequenceOf88 by using <S\_MAC>.  If paramReplaceSessionKeys is provided Then  Encrypt paramReplaceSessionKeys with <S\_ENC>  Calculate and add a MAC to the tag 0x87 of secondSequenceOf87 by using <S\_MAC>.  End If  Encrypt all <UPP\_SEG> with <S\_ENC>, or <PPK\_ENC> if paramReplaceSessionKeys is provided.  Calculate and add a MAC to all tags 0x86 of sequenceOf86 by using <S\_MAC>, or <PPK\_MAC> (and <PPK\_INIT\_MAC> for the first tag) if paramReplaceSessionKeys is provided. |

|  |  |
| --- | --- |
| Method | MTD\_GENERATE\_HASHED\_CC |
| Description | Generate an Hashed Confirmation Code based on the Confirmation Code and the Transaction ID given in parameter. |
| Parameter(s) |  paramConfirmationCode: The Confirmation Code (plain)   paramTransactionId: The Transaction ID (plain) |
| Details | Generate a SHA-256 of the paramConfirmationCode.  Concatenate the obtained hash value with the paramTransactionId.  Generate and return a SHA-256 of these two concatenated elements. |

|  |  |
| --- | --- |
| Method | MTD\_GET\_BPP |
| Description | Generates or verifies the JSON formatted GetBoundProfilePackage request |
| Parameter(s) |  paramTransactionId: random 16 byte identifier.   paramPrepareDownloadResponse structured as ASN.1 encoded as base 64. |
| Details | JSON body  {  "transactionId" : paramTransactionId,   "prepareDownloadResponse" : paramPrepareDownloadResponse  } |

|  |  |
| --- | --- |
| Method | MTD\_HANDLE\_NOTIF |
| Description | Generates or verifies the JSON formatted HandleNotification request |
| Parameter(s) | paramPendingNotification: PendingNotification data object |
| Details | JSON body  {  "pendingNotification" : paramPendingNotification  } |

|  |  |
| --- | --- |
| Method | MTD\_HTTP\_REQ |
| Description | Sends or verifies a secured HTTP request message delivering a JSON object payload using a network to an off-card entity. |
| Parameter(s) | * paramServerAddress: Target Server address * paramFunctionPath: Function path * paramRequestMessage: JSON Request message |
| Details | HTTP POST paramFunctionPath HTTP/1.1  Host: paramServerAddress  User-Agent: See NOTE 1  X-Admin-Protocol:gsma/rsp/v#RSP\_SVN Content-Type:application/json OR application/json;charset=UTF-8 (see NOTE 2) Content-Length: <L>  paramRequestMessage  NOTE 1: If the request is sent by the LPAd, the User-Agent SHALL be gsma-rsp-lpad. The "User-Agent" field may contain additional information after a semicolon. Otherwise the value of User-Agent is not specified by the current document. The additional information shall not be checked.  NOTE 2: the Content-Type checking is relaxed in this specification, in order to allow for common internet usage of “charset=UTF-8” and for compatibility with SGP.22 v3.0. If the request is sent by the entity under test, both values are acceptable (where linear white space as specified in RFC 2616 is allowed after the semi-colon). Further, all parts of these allowed Content-Type value SHALL be checked in a case-insensitive manner, as per RFC 2616. If the request is sent by a simulator, application/json shall be used.The HTTP POST request may contain additional header fields. These shall not be checked. |

|  |  |
| --- | --- |
| Method | MTD\_HTTP\_RESP |
| Description | Sends or verifies a secured HTTP response message delivering a JSON object payload using a network to an off-card entity. |
| Parameter(s) |  paramResponseMessage: JSON Response message |
| Details | HTTP/1.1 200 (OK)  X-Admin-Protocol: gsma/rsp/v#RSP\_SVN  Content-Type: application/json OR application/json;charset=UTF-8 (see NOTE)  Content-Length: <L>  paramResponseMessage  NOTE: the Content-Type checking is relaxed in this specification, in order to allow for common internet usage of “charset=UTF-8” and for compatibility with SGP.22 v3.0  If the response is sent by the entity under test, both values are acceptable (where linear white space as specified in RFC 2616 is allowed after the semi-colon). Further, all parts of these allowed Content-Type value SHALL be checked in a case-insensitive manner, as per RFC 2616. If the response is sent by a simulator, application/json shall be used.  The HTTP response may contain additional header fields. These shall not be checked. |

|  |  |
| --- | --- |
| Method | MTD\_INITIATE\_AUTHENTICATION |
| Description | Generates or verifies the JSON formatted Initiate Authentication request on ES9+ or ES11 as applicable. |
| Parameter(s) | * paramEUICCChallenge: random 16 byte challenge coded as base 64 * paramEUICCInfo1: eUICC information structured coded as base 64 * paramServerAddress: FQDN of the Server. |
| Details | JSON body  {  "euiccChallenge" : paramEUICCChallenge,   "euiccInfo1" : paramEUICCInfo1,  "smdpAddress" : paramServerAddress  } |

|  |  |
| --- | --- |
| Method | MTD\_REGISTER\_EVENT |
| Description | Send or checks the JSON formatted RegisterEvent request |
| Parameter(s) |  paramFunctionRequesterId: identification of the function requester.   paramFunctionCallId: identification of the function call.   paramEID: EID of the targeted eUICC   paramRspServerAddress: Address of the Server sending the RegisterEvent formatted as FQDN   paramEventId: unique Identification of the Event to be registered   paramForwardingIndicator: TRUE if registration has to be made to the Root SM-DS; FALSE if this is not to be made to the Root SM-DS |
| Details | JSON requestHeader  {  "header" : {  "functionRequesterIdentifier" : "paramFunctionRequesterId",  "functionCallIdentifier" : "paramFunctionCallId"  }  JSON body  {  "eid" : paramEID,  "rspServerAddress" : paramRspServerAddress,  "eventId" : paramEventId,  "forwardingIndicator" : paramForwardingIndicator  }  } |

|  |  |
| --- | --- |
| Method | MTD\_REMOVE\_NOTIF |
| Description | Constructs the command data for RemoveNotificationFromList |
| Parameter(s) |  paramSeqNumber: the sequence number to be removed |
| Details | request NotificationSentRequest ::= {  seqNumber paramSeqNumber  } |

|  |  |
| --- | --- |
| Method | MTD\_RETRIEVE\_NOTIF\_SEQ\_NUM |
| Description | Constructs the command data for RetrieveNotificationsList filtered by sequence number |
| Parameter(s) |  paramSeqNumber: the sequence number to be retrieved |
| Details | request RetrieveNotificationsListRequest ::= {  searchCriteria seqNumber paramSeqNumber } |

|  |  |
| --- | --- |
| Method | MTD\_SELECT |
| Description | Generates the SELECT command as defined in GlobalPlatform Card Specification [6]. |
| Parameter(s) |  paramAID: the AID to select |
| Details | - CLA = 0x or 4x (x = <CHANNEL\_NUMBER>)  - INS = A4  - P1 = 04  - P2 = 00  - LC = <L>  - paramAID  - LE = 00 |

|  |  |
| --- | --- |
| Method | MTD\_SEND\_SMS\_PP |
| Description | Generate and send an envelope SMS-PP download to the MNO-SD |
| Parameter(s) |  paramApdusList: the list of APDUs (plain) to send |
| Details | Generate and send the following envelope:  80 C2 00 00 <L>  D1 <L>  02 02 83 81 *-- Device identity Tag*  06 07 91 33 86 09 40 00 F0 *-- Address Tag (TON/NPI/..)*  0B <L> -- SMS TPDU  44 -- SMS-DELIVER  05 85 02 13 F2 -- TP-Originating-Address  7F -- TP-Protocol-Identifier  F6 -- TP-Data-Coding-Scheme  71 30 12 41 55 74 40 -- TP-Service-Centre-Time-Stamp  <L> -- TP-User-Data-Length  02 -- User-Data-Header-Length  70 -- IEIa  00 -- IEIDLa  <L> -- Command Packet Length (2 bytes)  <L> -- Command Header Length (1 byte)  12 21 -- SPI  00 -- KIC  15 –- KID (SCP80 Keyset version 0x01 in Triple DES)  B2 01 00 –- MNO-SD TAR  <MNO\_SCP80\_COUNTER>  00 -- Padding Counter  <CC> -- Cryptographic checksum  <C\_APDUS\_SCRIPT> -- Command APDUs script  <C\_APDUS\_SCRIPT> SHALL contain the paramApdusList (i.e. each APDU is named <APDU1>; <APDU2>; …; <APDUn> here after) formatted as an expanded structure with definite length as defined in ETSI TS 102 226 [14]:  AA <L>  22 <L> <APDU1>  22 <L> <APDU2>  …  22 <L> <APDUn>  The Cryptographic checksum <CC> SHALL be generated in Triple DES (outer-CBC mode using two different keys) with the #MNO\_SCP80\_AUTH\_KEY as defined in ETSI TS 102 225 [13].  If the command packet length is higher than 140 bytes, it SHALL be sent over several envelopes: SMS concatenation as defined in 3GPP TS 23.040 [22] SHALL be used. |

|  |  |
| --- | --- |
| Method | MTD\_STORE\_DATA |
| Description | Generates the STORE DATA command (Case 4) as defined in GlobalPlatform Card Specification [6]. |
| Parameter(s) |  paramCommandData: the command data |
| Details | - CLA = 8x or Cx (x = <CHANNEL\_NUMBER>)  - INS = E2  - P1 = 91  - P2 = 00  - LC = <L>  - paramCommandData  - LE = 00 |

|  |  |
| --- | --- |
| Method | MTD\_STORE\_DATA\_Case3 |
| Description | Generates the STORE DATA command (Case3) as defined in GlobalPlatform Card Specification [6]. |
| Parameter(s) |  paramCommandData: the command data |
| Details | - CLA = 8x or Cx (x = <CHANNEL\_NUMBER>)  - INS = E2  - P1 = 90  - P2 = 00  - LC = <L>  - paramCommandData |

|  |  |
| --- | --- |
| Method | MTD\_STORE\_DATA\_SCRIPT |
| Description | Generate (multiple) STORE DATA command(s) by breaking the data into smaller components (if needed) for transmission. |
| Parameter(s) |  paramTLVDataToTransmit: TLVs array or single TLV to transfer to the eUICC   paramCase4Command (optional parameter, default value = TRUE): TRUE if the APDU is a Case 4 command, FALSE if the APDU is a Case 3 command |
| Details | For each element of paramTLVDataToTransmit  If the size of the element is greater than 255 bytes, split the element in several blocks of  255 bytes. The last block MAY be shorter. Each block is named <DATA\_SUB\_PART> here  after.  If the element is up to 255 bytes, <DATA\_SUB\_PART> contains the value of the element.  The bit b1 of P1 in the STORE DATA commands is named <B1\_P1> here after and is  defined as below:  If paramCase4Command = TRUE Then  <B1\_P1> = 1  Else  <B1\_P1> = 0  End If  Set <STORE\_DATA\_BLOCK\_NUM> to 0  For each <DATA\_SUB\_PART>  If <DATA\_SUB\_PART> is an intermediate part, generate the following STORE DATA:  - CLA = 8x or Cx (x = <CHANNEL\_NUMBER>)  - INS = E2  - P1 = 1x (x = <B1\_P1>)  - P2 = <STORE\_DATA\_BLOCK\_NUM>  - LC = <L>  - <DATA\_SUB\_PART>  - LE = 00 –- present only if paramCase4Command = TRUE  If <DATA\_SUB\_PART> is the last part, generate the following STORE DATA:  - CLA = 8x or Cx (x = <CHANNEL\_NUMBER>)  - INS = E2  - P1 = 9x (x = <B1\_P1>)  - P2 = <STORE\_DATA\_BLOCK\_NUM>  - LC = <L>  - <DATA\_SUB\_PART>  - LE = 00 –- present only if paramCase4Command = TRUE  Increase the <STORE\_DATA\_BLOCK\_NUM> by 1  End  End |

|  |  |
| --- | --- |
| Method | MTD\_TEST\_ES8+\_GET\_BPP\_PPK |
| Description | Tests the received boundProfilePackage element according to #R\_GET\_BPP\_RESP\_OP1\_PPK |
| Parameter(s) |  paramResponse the response to GetBoundProfilePackage   paramS\_MAC the 128 bit SCP03t MACing Session key   paramS\_ENC the 128 bit SCP03t Encryption Session key   paramPPK\_MAC the 128 bit Profile Protection MACing Key   paramPPK\_ENC the 128 bit Profile Protection Encryption Key   * paramMetaData the ASN.1 StoreMetadataRequest element associated to a RSP profile |
| Details | Parse paramResponse into #R\_GET\_BPP\_RESP\_OP1\_PPK and perform the following tests:   Verify that each element in firstSequenceOf87, sequenceOf88, secondSequenceOf87 and sequenceOf86 has a total length (including tag and length fields) of 1020 or less   Verify the integrity of each element in firstSequenceOf87, sequenceOf88 and secondSequenceOf87 using paramS\_MAC   Verify that <TRANSACTION\_ID\_ISC> in #INIT\_SC\_PROF1 matches <S\_TRANSACTION\_ID>   Verify the validity of smdpSign <SM\_DP+\_SIGN> in #INIT\_SC\_PROF1 using #PK\_SM\_DPpb\_ECDSA   Retrieve #CONF\_ISDP\_PROF1\_SMDP from <CONF\_ISDP\_PROF1\_ENC> using paramS\_ENC and validate the content of #CONF\_ISDP\_PROF1\_SMDP   Construct the complete metadata element from the <SMDP\_METADATA\_SEG\_MAC> segment(s) and verify that it matches paramMetaData   Retrieve #REPLACE\_S\_KEYS\_REQ from <REPLACE\_S\_KEYS\_REQ\_ENC> using paramS\_ENC and validate the content of #REPLACE\_S\_KEYS\_REQ   Verify that the lengths of paramPPK\_ENC and paramPPK\_MAC in #REPLACE\_S\_KEYS\_REQ are each 16 bytes   Verify the integrity of each <PPP\_OP\_PROF1\_SEG\_PPK> element using paramPPK\_MAC   Retrieve the <UPP\_OP\_PROF1\_SEG> segment(s) from the <PPP\_OP\_PROF1\_SEG\_PPK> segment(s) using paramPPK\_ENC, construct the complete Profile from the <UPP\_OP\_PROF1\_SEG> segment(s), then verify that the complete Profile matches #UPP\_OP\_PROF1 |

|  |  |
| --- | --- |
| Method | MTD\_TEST\_ES8+\_GET\_BPP\_SK |
| Description | Tests the received boundProfilePackage element according to #R\_GET\_BPP\_RESP\_OP1\_SK |
| Parameter(s) |  paramResponse the response to GetBoundProfilePackage   paramS\_MAC the 128 bit SCP03t MACing Session key   paramS\_ENC the 128 bit SCP03t Encryption Session key   * paramMetaData the ASN.1 StoreMetadataRequest element associated to a RSP profile |
| Details | Parse paramResponse into #R\_GET\_BPP\_RESP\_OP1\_SK and perform the following tests:   Verify that each element in firstSequenceOf87, sequenceOf88 and sequenceOf86 has a total length (including tag and length fields) of 1020 or less   Verify the integrity of each element in firstSequenceOf87, sequenceOf88 and sequenceOf86 using paramSMAC   Verify that <TRANSACTION\_ID\_ISC> in #INIT\_SC\_PROF1 matches <S\_TRANSACTION\_ID>   Verify the validity of smdpSign <SM\_DP+\_SIGN> in #INIT\_SC\_PROF1 using #PK\_SM\_DPpb\_ECDSA   Retrieve #CONF\_ISDP\_PROF1\_SMDP from <CONF\_ISDP\_PROF1\_ENC> using paramS\_ENC and validate the content of #CONF\_ISDP\_PROF1\_SMDP   Construct the complete metadata element from the <SMDP\_METADATA\_SEG\_MAC> segment(s) and verify that it matches paramMetaData   Retrieve the <UPP\_OP\_PROF1\_SEG> segment(s) from the <PPP\_OP\_PROF1\_SEG\_SK> segment(s) using paramS\_ENC, then construct the complete Profile from the <UPP\_OP\_PROF1\_SEG> segment(s), then verify that the complete Profile matches #UPP\_OP\_PROF1 |

|  |  |
| --- | --- |
| Method | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC |
| Description | Finalizes the Transport Layer Security (TLS) handshake in Server authentication mode on ES9+, or ES11 (Client side). |
| Parameter(s) |  paramClientKeyExchange: ClientKeyExchange message |
| Details | Sends the session key information in TLS ClientKeyExchange message, ChangeCipherSpec and Finished message. |

|  |  |
| --- | --- |
| Method | MTD\_TLS\_CLIENT\_HELLO |
| Description | Sends or checks the Client Hello message used to initiate the Transport Layer Security (TLS) handshake in Server authentication or Mutual authentication mode on ES9+, ES11, ES12 or ES15. |
| Parameter(s) |  paramTLSversion: TLS protocol version   paramAlgs: cipher suite types supported   paramSessionID: Session ID   paramExts: Extensions data for “supported\_signature\_algorithms”, “trusted\_ca\_keys” or other (optional) |
| Details | Sends or receives a TLS ClientHello message according to the parameters defined above.  In addition the following parameters will be set:   The list of compression algorithms supported by the client is not explicitly defined, but by default it will be set to NULL.   The random of 4 bytes representing time since epoch on client host and 28 random bytes is not explicitly defined but it SHALL be generated by the test tool TLS implementation  NOTE: The Supported Elliptic Curves Extension and the Supported Point Formats Extension extensions MAY be sent by the Client. |

|  |  |
| --- | --- |
| Method | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH |
| Description | Sends or checks the messages to finalize the Transport Layer Security (TLS) handshake in Mutual authentication mode on ES12 or ES15 (Client side). |
| Parameter(s) |  paramClientCertificate: TLS Client certificate for authentication used in the Client Certificate Message   paramClientKeyExchange: The Client TLS Ephemeral Key used in the ClientKeyExchange message |
| Details | Sends the TLS Client Certificate, ClientKeyExchange, Certificate Verify, ChangeCipherSpec and Finished message in this order according to the parameters defined above.  NOTE 1: The CertificateVerify Message is not explicitly defined in this method but the CLIENT or test tool implementation SHALL be responsible for generating this message. It is the signature of the concatenation of all the data from all messages in this handshake up to, but not including, this message i.e. all handshake messages starting at ClientHello up to, but not including, this message itself using the specified Signature and Hash Algorithm.  NOTE 2: ChangeCipherSpec messages, alerts, and any other record type are not handshake messages and are not included in the signature computations. |

|  |  |
| --- | --- |
| Method | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC |
| Description | Sends or checks the replies to the Client Hello in the Transport Layer Security (TLS) handshake in Mutual authentication mode on ES12 or ES15. |
| Parameter(s) | * paramTLSVersion: TLS protocol version used in the Server Hello Message * paramAlgs: cipher suite selected used in the Server Hello Message * paramSessionID: Session ID used in the Server Hello Message * paramServerCertificate: TLS Server certificate for authentication used in the Server Certificate Message * paramServerTLSEphemeralKey: TLS Server ephemeral key used in the Server Key Exchange Message * paramClientCertificateType: type of certificate requested used in the Client Certificate Request Message * paramSignatureAndHashAlgorithm: Signature and Hash Algorithm to be verified used in the Client Certificate Request Message * paramDistinguishedName: DN of the CI that signed and issued the certificate used in the Client Certificate Request Message |
| Details | Sends or receives a TLS ServerHello, Server Certificate, ServerKeyExchange, Client Certificate Request and ServerHelloDone message in this order according to the parameters defined above. In addition the following parameter will be received:   * ServerHello   o The random of 4 bytes representing time since epoch on client host and 28 random bytes is not explicitly defined but it SHALL be generated by the Server under test.   * ServerKeyExchange   o The ECParameters are not explicitly defined in the ServerKeyExchange message but it SHALL be generated by the Server under test or the test tool implementation.  NOTE: The Supported Elliptic Curves Extension and the Supported Point Formats Extension extensions MAY be sent by the CLIENT therefore this method SHALL respond appropriately when used by the SERVER or the S\_SERVER. |

|  |  |
| --- | --- |
| Method | MTD\_TLS\_SERVER\_END |
| Description | Send or checks the finalization of the Transport Layer Security (TLS) handshake in Server or Mutual authentication mode on ES9+,ES11, ES12 or ES15 (Server side). |
| Parameter(s) |  paramChangeCipherSpec: ChangeCipherSpec message   paramFinish: Finished message |
| Details | Sends a ChangeCipherSpec and Finished message in this order according to the parameters defined above. |

|  |  |
| --- | --- |
| Method | MTD\_TLS\_SERVER\_HELLO\_ETC |
| Description | Send or Receives to the Client Hello in the Transport Layer Security (TLS) handshake in Server authentication mode on ES9+, or ES11. |
| Parameter(s) |  paramTLSversion: TLS protocol version   paramAlgs: cipher suite selected   paramSessionID: Session ID   paramCertificate: TLS server certificate for authentication   paramServerTLSEphemeralKey: TLS Server ephemeral key. |
| Details | Sends or Receives a TLS ServerHello, Server Certificate, ServerKeyExchange and ServerHelloDone message in this order according to the parameters defined above.  NOTE 1: The random of 4 bytes representing time since epoch on client host and 28 random bytes is not explicitly defined in the Server Hello message but it SHALL be generated by the Server under test.  NOTE 2: If no parameter mentioned paramServerTLSEphemeralKey, the value SHALL be set as defined in [24] for ServerKeyExchange. No verification required. |

C.2 Procedures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES11\_AUTH\_CLIENT | | |
|  | Description | Authenticate Server procedure and Event Retrieval from SM-SD. | | |
| For LPAd testing, execute the following steps: | | | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DS | Send ES11.AuthenticateClient method | MTD\_HTTP\_REQ(#TEST\_ROOT\_DS\_ADDRESS, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>, #R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_DS\_OK) | No error |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID | | |
|  | Description | Performs Common Mutual Authentication on ES11 from S\_LPAd to SM-DS under test supplying <EVENT\_ID\_R> and verifies that the pending Event #EVENT\_ENTRY\_1 is retrieved. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SM-DS | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | |
| 2 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EVENT\_ID\_R)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_EVENT\_ID\_ERROR | | |
|  | Description | Performs Common Mutual Authentication on ES11 from S\_LPAd to SM-DS under test supplying <EVENT\_ID\_R> and verifies that the pending Event #EVENT\_ENTRY\_1 is not available. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SM-DS | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | |
| 2 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_EVENT\_ID\_R)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_9\_5\_3\_9) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID | | |
|  | Description | Performs Common Mutual Authentication on ES11 from S\_LPAd to SM-DS under test supplying no MatchingId and verifies that the pending Event #EVENT\_ENTRY\_1 is retrieved. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SM-DS | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | |
| 2 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_OMITTED)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES11\_VERIFY\_EVENT\_RETRIEVAL\_NO\_EVENT\_ID\_ERROR | | |
|  | Description | Performs Common Mutual Authentication on ES11 from S\_LPAd to SM-DS under test supplying no MatchingId and verifies that no events are available. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SM-DS | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | |
| 2 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DS\_ADDRESS\_ES11)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DS | MTD\_HTTP\_REQ(  #IUT\_SM\_DS\_ADDRESS\_ES11,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_MATCHING\_ID\_OMITTED)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_EMPTY\_OK) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Procedure | PROC\_ES11\_INIT\_AUTH | | |
|  | | Description | Initiate Authentication procedure with SM-DS. | | |
| For LPAd testing, execute the following steps: | | | | | |
| Step | | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DS | Send ES11.InitiateAuthentication method | MTD\_HTTP\_REQ( #TEST\_ROOT\_DS\_ADDRESS, #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  <EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_ROOT\_DS\_ADDRESS)) |  |
| 2 | S\_SM-DS → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_DS\_OK) | No error |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_EUICC\_INITIALIZATION\_SEQUENCE | | |
|  | Description | Initialize communication between the S\_Device and the eUICC. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_Device → eUICC | RESET | ATR present |  |
| 2 | S\_Device → eUICC | [SELECT\_MF] | FCP Template present  SW=0x9000 |  |
| 3 | S\_Device → eUICC | [TERMINAL\_CAPABILITY\_LPAd] | SW=0x9000 |  |
| 4 | S\_Device → eUICC | [TERMINAL\_PROFILE] | Toolkit initialization THEN SW=0x9000 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Procedure | PROC\_EUICC\_INITIALIZATION\_SEQUENCE\_eUICCProfileStateChanged | | | |
|  | Description | Initialize communication between the S\_Device and the eUICC. | | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_Device → eUICC | RESET | ATR returned by eUICC |  |
| 2 | S\_Device → eUICC | [SELECT\_MF] | FCP Template present  SW=0x9000 |  |
| 3 | S\_Device → eUICC | [TERMINAL\_CAPABILITY\_LPAd] | SW=0x9000 |  |
| 4 | S\_Device → eUICC | [TERMINAL\_PROFILE\_eUICCProfileStateChanged] | Toolkit initialization THEN SW=0x9000 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_OPEN\_LOGICAL\_CHANNEL\_AND\_SELECT\_ISDR | | |
|  | Description | The LPAd opens a logical channel and selects the ISD-R. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | [MANAGE\_CHANNEL\_OPEN] | Extract the <CHANNEL\_NUMBER> from response data  SW=0x9000 |  |
| 2 | S\_LPAd → eUICC | MTD\_SELECT(#ISD\_R\_AID) | SW=0x9000 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_AUTH\_CLIENT | | |
|  | **Description** | Authenticate Server procedure without Confirmation Code.  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO and #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK are used with the correct MatchingID defined by the Add Profile initiation procedure (Activation Code content or Empty MatchingID). | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| For LPAd testing, execute the following steps: | | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_OK) | No error |  |
| For SM-DP+ testing, execute the following steps: | | | | |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_AUTH\_CLIENT\_CC | | |
|  | **Description** | Authenticate Server procedure (via Activation Code) with Confirmation Code. #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO and #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK are used with the correct MatchingID defined by the Add Profile initiation procedure (Activation Code content or Empty MatchingID). | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| For LPAd testing, execute the following steps: | | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.AuthenticateClient method | MTD\_HTTP\_REQ( #TEST\_DP\_ADDRESS1, #PATH\_AUTH\_CLIENT, MTD\_AUTHENTICATE\_CLIENT(<S\_TRANSACTION\_ID>,  #R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP (#AUTH\_CLIENT\_OK\_CC) | No error |  |
| For SM-DP+ testing, execute the following steps: | | | | |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP(#R\_AUTH\_CLIENT\_OK\_CC) |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Procedure | PROC\_ES9+\_GET\_BPP | | | |
|  | | Description | Get BPP procedure without Confirmation Code. | | | |
| Step | | Direction | Sequence / Description | Expected result | | REQ |
| For LPAd testing, execute the following steps: | | | | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_NO\_CC)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_OK) | | No error |  |
| For SM-DP+ testing, execute the following steps: | | | | | |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP)) | | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_SK) |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | Procedure | PROC\_ES9+\_GET\_BPP\_CC | | | |
|  | | Description | Get BPP procedure with Confirmation Code. | | | |
| Step | | Direction | Sequence / Description | Expected result | | REQ |
| For LPAd testing, execute the following steps: | | | | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.GetBoundProfilePackage method | | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_GET\_BPP, MTD\_GET\_BPP(<S\_TRANSACTION\_ID>,  #R\_PREP\_DOWNLOAD\_WITH\_CC)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP(#GET\_BPP\_OK) | | No error |  |
| For SM-DP+ testing, execute the following steps: | | | | | |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_CC)) | | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_SK) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Procedure | PROC\_ES9+\_HANDLE\_NOTIF | | |
|  | | Description | Handle Notification procedure. | | |
| Step | | Direction | Sequence / Description | Expected result | REQ |
| For LPAd testing, execute the following steps: | | | | | |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.HandleNotification method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_HANDLE\_NOTIF, MTD\_HANDLE\_NOTIF(#R\_PIR\_OK)) See NOTE 2 |  |
| 2 | S\_SM-DP+ → LPAd | #R\_HTTP\_204\_OK | No error |  |
| NOTE 1: Other Notifications MAY be sent within the same HTTPS session.  NOTE 2: The values of notificationAddress, iccid and smdpOid used in #R\_PIR\_OK MAY vary depending on the context (ICCID of the downloaded profile, used SM-DP+ address and certificate). | | | | |
| For SM-DP+ testing: Not Used (FFS). | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_AUTH\_CLIENT\_FAIL\_DEF\_DP\_USE\_CASE\_INVALID\_MATCHING\_ID | | |
|  | **Description** | AuthenticateClient fails due to an Invalid Matching ID. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_6\_3\_8) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_SK | | |
|  | **Description** | End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the Bound Profile Package. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_SK) |  |
| 5 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS)  Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CANCEL\_SESSION\_PPK | | |
|  | **Description** | End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys, profile protection keys and the generation of the Bound Profile Package. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK) |  |
| 5 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS)  Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_PPK | | |
|  | **Description** | End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys, profile protection keys and the generation of the Bound Profile Package when a Confirmation Code is required. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_CC) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK) |  |
| 5 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS)  Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC\_EN | | |
|  | **Description** | Performs Common Mutual Authentication and then delivers the Bound Profile Package to the LPAd for enable metadata notifications. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_EN) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROF\_DOWNLOAD\_ACT\_CODE\_USE\_CASE\_CANCEL\_SESSION | | |
|  | **Description** | End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the Bound Profile Package. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK) |  |
| 5 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS)  Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROF\_DOWNLOAD\_SM\_DS\_USE\_CASE\_CANCEL\_SESSION | | |
|  | **Description** | End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the bound profile package. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_ES9+\_TLS\_INITIALIZATION\_SERVER\_AUTH | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(    #IUT\_SM\_DP\_ADDRESS,    #PATH\_INITIATE\_AUTH,    MTD\_INITIATE\_AUTHENTICATION(       #S\_EUICC\_CHALLENGE,        #S\_EUICC\_INFO1,       #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(    #IUT\_SM\_DP\_ADDRESS,    #PATH\_AUTH\_CLIENT,    MTD\_AUTHENTICATE\_CLIENT(       <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_SMDS\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(   #IUT\_SM\_DP\_ADDRESS,     #PATH\_GET\_BPP,   MTD\_GET\_BPP(     <S\_TRANSACTION\_ID>,      #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK) |  |
| 5 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(    #IUT\_SM\_DP\_ADDRESS,    #PATH\_CANCEL\_SESSION,    MTD\_CANCEL\_SESSION(       <S\_TRANSACTION\_ID>,        #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS)  Cancel Session request accepted by SM-DP+ and ongoing RSP session shall enter retry mode. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |
|  | **Description** | Performs Common Mutual Authentication for the Profile Download Default SM\_DP+ use case without a confirmation code. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |
|  | **Description** | Performs Common Mutual Authentication for the Profile Download Default SM\_DP+ use case with a confirmation code. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_CC) |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Procedure | PROC\_ES9+\_INIT\_AUTH | | |
|  | | **Description** | Initiate Authentication procedure. | | |
| For LPAd testing, execute the following steps: | | | | | |
| Step | | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SM-DP+ | Send ES9+.InitiateAuthentication method | MTD\_HTTP\_REQ(  #TEST\_DP\_ADDRESS1,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  <EUICC\_CHALLENGE>,   #R\_EUICC\_INFO1, #TEST\_DP\_ADDRESS1)) |  |
| 2 | S\_SM-DP+ → LPAd | MTD\_HTTP\_RESP( #INITIATE\_AUTH\_OK) | No error |  |
| For SM-DP+ testing, execute the following steps: | | | | |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION( #S\_EUICC\_CHALLENGE,  #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_VERIFY\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_NO\_CC\_FAIL | | |
|  | **Description** | Verifies that Common Mutual Authentication for the Profile Download Default SM\_DP+ use case without a confirmation code fails due to the profile being in the ‘Installed‘ or ‘Error’ state. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(    #IUT\_SM\_DP\_ADDRESS,    #PATH\_INITIATE\_AUTH,    MTD\_INITIATE\_AUTHENTICATION(       #S\_EUICC\_CHALLENGE,        #S\_EUICC\_INFO1,       #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP(#R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(    #IUT\_SM\_DP\_ADDRESS,    #PATH\_AUTH\_CLIENT,    MTD\_AUTHENTICATE\_CLIENT(       <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_1\_1\_3\_8) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_VERIFY\_SESSION\_IS\_CANCELLED | | |
|  | **Description** | Verify that the RSP session identified by the TransactionID <S\_TRANSACTION\_ID> has been cancelled by the eUICC (i.e. Common Mutual Authentication and Profile Download procedures SHALL be rejected as long as no GetEUICCChallenge has been requested). | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #PREP\_DOWNLOAD\_NO\_CC) | #R\_PREP\_DOWN\_NO\_SESSION  SW=0x9000  The transactionId returned in the response SHALL not be checked (any value SHALL be accepted) |  |
| 2 | S\_LPAd → eUICC | MTD\_STORE\_DATA\_SCRIPT(  #AUTHENTICATE\_SMDP) | #R\_AUTH\_SERVER\_NO\_SESSION  SW = 0x9000  The transactionId returned in the response SHALL not be checked (any value SHALL be accepted) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROF\_DOWNLOAD\_DEF\_DP\_USE\_CASE\_CC\_CANCEL\_SESSION\_SK | | |
|  | **Description** | End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the Bound Profile Package when a Confirmation Code is required. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_CC) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_SK) |  |
| 5 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_CANCEL\_SESSION,  MTD\_CANCEL\_SESSION(  <S\_TRANSACTION\_ID>,   #CS\_RESP\_OK\_POSTPONED)) | MTD\_HTTP\_RESP( #R\_SUCCESS)  Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC\_RETRY | | |
|  | **Description** | Performs Common Mutual Authentication for the Profile Download Default SM\_DP+ use case without a confirmation code. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_RETRY\_OK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_INVALID\_CC | | |
|  | **Description** | Performs Common Mutual Authentication for the Profile Download Default SM\_DP+ use case with an invalid confirmation code provided in the GetBoundProfilePackage. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| IC1 | PROC\_ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_CC | | |  |
| 1 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>, #PREP\_DOWNLOAD\_RESP\_8\_2\_7\_3\_8)) | MTD\_HTTP\_RESP(  #R\_ERROR\_8\_2\_7\_3\_8) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC**\_**ES9+\_CMA\_PD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC\_RETRY | | |
|  | **Description** | Performs Common Mutual Authentication for the Profile Download Default SM\_DP+ use case without a confirmation code. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_RETRY\_OK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH | | |
| **Description** | Establishes the Transport Layer Security (TLS) v1.2 connection between the Client (S\_)LPAd and (S\_)SERVER using Server authentication mode on ES9+ or ES11. | | |
| For LPAd testing, execute the following steps: | | | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | LPAd → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(  #IUT\_TLS\_VERSION, <TLS\_CIPHER\_SUITES>, <SESSION\_ID\_CLIENT>, <EXT\_SHA256\_ECDSA>) |  |
| 2 | S\_SERVER → LPAd | MTD\_TLS\_SERVER\_HELLO\_ETC(#TLS\_VERSION\_1\_2, #S\_TLS\_CIPHER\_SUITE, <S\_SESSION\_ID\_SERVER>, #CERT\_S\_SERVER\_TLS) | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC(<CLIENT\_TLS\_EPHEM\_KEY>) |  |
| 3 | S\_SERVER → LPAd | Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages) | HTTPS connection established |  |
| For Server (SM-DP+ or SM-DS) testing, execute the following steps: | | | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2, <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,  #CERT\_SERVER\_TLS) |  |
| 2 | S\_LPAd → SERVER | MTD\_TLS\_CLIENT\_KEY\_EXCH\_ETC( <CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,  <SERVER\_FINISHED>) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC\_NO\_CC | | |
|  | **Description** | Performs Common Mutual Authentication and then delivers the Bound Profile Package to the LPAd. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_VERIFY\_PROFILE\_DOWNLOAD\_DEF\_SMDP\_ADDRESS\_UC | | |
|  | **Description** | Verifies that Common Mutual Authentication occurs successfully and that the Bound Profile Package is generated and successfully delivered to the LPAd. | | |
| Step | Direction | Sequence / Description | Expected result | REQ |
| 1 | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | | | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH, MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK)  OR  MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_RETRY\_OK) |  |
| 4 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,   #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,   #PREP\_DOWNLOAD\_RESP)) | MTD\_HTTP\_RESP( #R\_GET\_BPP\_RESP\_OP1\_PPK) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_VERIFY\_PROFILE\_NOT\_RELEASED\_EMPTY\_MID | | |
|  | **Description** | Performs Common Mutual Authentication on ES9+ from S\_LPAd to SM-DP+ under test supplying an empty MatchingId and verifies that there is no pending profile in Released state. | | |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| 1 | S\_LPAd → SM-DP+ | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | |  | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>, #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_ERROR\_8\_2\_1\_2) |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_EMPTY\_MID\_WITH\_CC | | |
|  | **Description** | Performs Common Mutual Authentication on ES9+ from S\_LPAd to SM-DP+ under test supplying an empty MatchingId and verifies that there is at least one pending profile in Released state. | | |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| 1 | S\_LPAd → SM-DP+ | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | |  | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK)) | MTD\_HTTP\_RESP( # R\_AUTH\_CLIENT\_OK\_CC) |  | |
| 4 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK) |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH | | |
|  | **Description** | Establishes the Transport Layer Security (TLS) v1.2 connection between the Client and Server using Mutual authentication mode on ES2+, ES12 or ES15.  For Client and Server testing the Server MAY be the SM-DS or the SM-DP+. | | |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| For Client testing, execute the following steps: | | | | | |
| 1 | CLIENT → S\_SERVER | Send TLS Client Hello | MTD\_TLS\_CLIENT\_HELLO(    #IUT\_CLIENT\_TLS\_VER,   <TLS\_CIPHER\_SUITES>,   <SESSION\_ID\_CLIENT>,   <EXT\_SHA256\_ECDSA>) |  | |
| 2 | S\_SERVER → CLIENT | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(     #TLS\_VERSION\_1\_2, <S\_SEL\_TLS\_CIPHER\_SUITE>,   <SESSION\_ID\_RANDOM>,   #CERT\_S\_SERVER\_TLS,   <SERVER\_TLS\_EPHEM\_KEY>,   #CLIENT\_CERT\_TYPE,   #S\_SAH\_SHA256\_ECDSA,   #DIST\_NAME\_CI) | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(   #CERT\_CLIENT\_TLS, <CLIENT\_TLS\_EPHEM\_KEY>) |  | |
| 3 | S\_SERVER → CLIENT | MTD\_TLS\_SERVER\_END(    #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) | HTTPS connection established |  | |
| For Server testing, execute the following steps: | | | | | |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI) |  | |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_CLIENT\_TLS,  <CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_ES9+\_VERIFY\_PROFILE\_RELEASED\_WITH\_MID\_WITH\_CC | | |
|  | **Description** | Performs Common Mutual Authentication on ES9+ from S\_LPAd to SM-DP+ under test supplying a MatchingId set to #MATCHING\_ID\_1 and verifies that there is a pending profile in Released state. | | |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| 1 | S\_LPAd → SM-DP+ | PROC\_TLS\_INITIALIZATION\_SERVER\_AUTH on ES9+ | |  | |
| 2 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_INITIATE\_AUTH,  MTD\_INITIATE\_AUTHENTICATION(  #S\_EUICC\_CHALLENGE,   #S\_EUICC\_INFO1,  #IUT\_SM\_DP\_ADDRESS)) | MTD\_HTTP\_RESP( #R\_INITIATE\_AUTH\_OK) |  | |
| 3 | S\_LPAd → SM-DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_AUTH\_CLIENT,  MTD\_AUTHENTICATE\_CLIENT(  <S\_TRANSACTION\_ID>,  #AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK)) | MTD\_HTTP\_RESP( #R\_AUTH\_CLIENT\_OK\_CC) |  | |
| 4 | S\_LPAd → SM‑DP+ | MTD\_HTTP\_REQ(  #IUT\_SM\_DP\_ADDRESS,  #PATH\_GET\_BPP,  MTD\_GET\_BPP(  <S\_TRANSACTION\_ID>,  #PREP\_DOWNLOAD\_RESP\_CC)) | MTD\_HTTP\_RESP(#R\_GET\_BPP\_RESP\_OP1\_PPK) |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Procedure | PROC\_TLS\_INITIALIZATION\_MUTUAL\_AUTH\_INV\_OID | | |
|  | **Description** | Establishes the Transport Layer Security (TLS) v1.2 connection between the Client and Server using Mutual authentication mode on ES12 or ES15 with a Client Certificate that has an invalid OID.  For Client and Server testing the Server MAY be the SM-DS or the SM-DP+. | | |
| Step | Direction | Sequence / Description | Expected result | REQ | |
| 1 | S\_CLIENT → SERVER | MTD\_TLS\_CLIENT\_HELLO(  #TLS\_VERSION\_1\_2,  #MIN\_TLS\_CIPHER\_SUITES,  #S\_SESSION\_ID\_EMPTY,  #S\_EXT\_SHA256\_ECDSA) | MTD\_TLS\_MUTUAL\_AUTH\_SERVER\_HELLO\_ETC(   #TLS\_VERSION\_1\_2,  <SEL\_TLS\_CIPHER\_SUITE>,  <SESSION\_ID\_RANDOM>,   #CERT\_SERVER\_TLS,  #CLIENT\_CERT\_TYPE,  <SAH\_SHA256\_ECDSA>,  #DIST\_NAME\_CI) |  | |
| 2 | S\_CLIENT → SERVER | MTD\_TLS\_MUTUAL\_AUTH\_CLIENT\_EXCH(  #CERT\_S\_CLIENT\_TLS\_INV\_OID,  <CLIENT\_TLS\_EPHEM\_KEY>) | MTD\_TLS\_SERVER\_END(  #CHANGE\_CIPHER\_SPEC,   <SERVER\_FINISHED>) |  | |

Annex D Commands And Responses

D.1 ES8+ Requests And Responses

D.1.1 ES8+ Requests

|  |  |
| --- | --- |
| Name | Content |
| CONF\_ISDP\_EMPTY | req ConfigureISDPRequest ::={} |
| CONF\_ISDP\_MAX\_LENGTH | req ConfigureISDPRequest ::={  dpProprietaryData { *-- size=128 bytes*  dpOid #S\_SM\_DP+\_OID,  additionalSmdpData #ADDITIONAL\_SMDP\_DATA\_MAX\_LENGTH  }  }  -- NOTE: Instead of  DpProprietaryData ::= SEQUENCE {  dpOid OBJECT IDENTIFIER  -- additional data objects defined by the  -- SM-DP+ MAY follow  }  -- the following structure is used to test the  -- DpProprietaryData size:  DpProprietaryData ::= SEQUENCE {  dpOid OBJECT IDENTIFIER,  additionalSmdpData OCTET STRING OPTIONAL  } |
| CONF\_ISDP\_PROF1 | req ConfigureISDPRequest ::={  dpProprietaryData {  dpOid #S\_SM\_DP+\_OID  }  } |
| CONF\_ISDP\_PROF1\_SMDP | req ConfigureISDPRequest ::={  dpProprietaryData {  dpOid #IUT\_SM\_DP\_OID  -- additional data objects defined by the SM-DP+ MAY follow  }-- optional } |
| CONF\_ISDP\_SIZE\_EXCEEDED | req ConfigureISDPRequest ::={  dpProprietaryData { *-- size=129 bytes*  dpOid #S\_SM\_DP+\_OID,  additionalSmdpData   #ADDITIONAL\_SMDP\_DATA\_EXCEEDED\_MAX  }  }  -- NOTE: Instead of  DpProprietaryData ::= SEQUENCE {  dpOid OBJECT IDENTIFIER  -- additional data objects defined by the  -- SM-DP+ MAY follow  }  -- the following structure is used to test the  -- DpProprietaryData size:  DpProprietaryData ::= SEQUENCE {  dpOid OBJECT IDENTIFIER,  additionalSmdpData OCTET STRING OPTIONAL  } |
| FULL\_METADATA | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1}  } |
| INIT\_SC\_INVALID\_CRT | req InitialiseSecureChannelRequest ::={  remoteOpId #REMOTE\_OP\_ID\_INSTALL,  transactionId <S\_TRANSACTION\_ID>,  controlRefTemplate {  keyType #INVALID\_KEY\_TYPE,  keyLen #KEY\_LENGTH,  hostId #HOST\_ID  },  smdpOtpk <OTPK\_S\_SM\_DP+\_ECKA>,  smdpSign <S\_SM\_DP+\_SIGN>  } |
| INIT\_SC\_INVALID\_OP\_ID | req InitialiseSecureChannelRequest ::={  remoteOpId #INVALID\_REMOTE\_OP\_ID,  transactionId <S\_TRANSACTION\_ID>,  controlRefTemplate {  keyType #KEY\_TYPE,  keyLen #KEY\_LENGTH,  hostId #HOST\_ID  },  smdpOtpk <OTPK\_S\_SM\_DP+\_ECKA>,  smdpSign <S\_SM\_DP+\_SIGN>  } |
| INIT\_SC\_INVALID\_SIGN | req InitialiseSecureChannelRequest ::={  remoteOpId #REMOTE\_OP\_ID\_INSTALL,  transactionId <S\_TRANSACTION\_ID>,  controlRefTemplate {  keyType #KEY\_TYPE,  keyLen #KEY\_LENGTH,  hostId #HOST\_ID  },  smdpOtpk <OTPK\_S\_SM\_DP+\_ECKA>,  smdpSign <S\_SM\_DP+\_SIGN>  }  *The <S\_SM\_DP+\_SIGN> SHALL NOT be computed using the #SK\_S\_SM\_DPpb\_ECDSA but SHALL have the same length as for a valid signature* |
| INIT\_SC\_INVALID\_TRANS\_ID | req InitialiseSecureChannelRequest ::={  remoteOpId #REMOTE\_OP\_ID\_INSTALL,  transactionId <INVALID\_TRANSACTION\_ID>,  controlRefTemplate {  keyType #KEY\_TYPE,  keyLen #KEY\_LENGTH,  hostId #HOST\_ID  },  smdpOtpk <OTPK\_S\_SM\_DP+\_ECKA>,  smdpSign <S\_SM\_DP+\_SIGN>  } |
| INIT\_SC\_PROF1 | req InitialiseSecureChannelRequest ::={  remoteOpId #REMOTE\_OP\_ID\_INSTALL,  transactionId <TRANSACTION\_ID\_ISC>,  controlRefTemplate {  keyType #KEY\_TYPE,  keyLen #KEY\_LENGTH,  hostId #IUT\_SM\_DP\_HOST\_ID  },  smdpOtpk <OTPK\_SM\_DP+\_ECKA>,  smdpSign <SM\_DP+\_SIGN>  } |
| METADATA\_ICCID\_MISMATCH | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF2,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1  } |
| METADATA\_MCCMNC\_MISMATCH | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileOwner {  mccMnc #MCC\_MNC2  },  profilePolicyRules {ppr2}  } |
| METADATA\_NO\_CLASS | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  }  } |
| METADATA\_OP\_PROF1 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  }  } |
| METADATA\_OP\_PROF1\_EN | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationEnable  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  }  } |
| METADATA\_OP\_PROF1\_INST\_DIFF | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS2  }  },  profileOwner {  mccMnc #MCC\_MNC1  }  } |
| METADATA\_OP\_PROF1\_NO\_NOTIF | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  profileOwner {  mccMnc #MCC\_MNC1  }  } |
| METADATA\_OP\_PROF2\_MEMRES1 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF2,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF2,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS2  }  },  profileOwner {  mccMnc #MCC\_MNC2  },  profilePolicyRules { ppr2 }  } |
| METADATA\_OP\_PROF4\_MEMRES1 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF4,  serviceProviderName #SP\_NAME4,  profileName #NAME\_OP\_PROF4,  iconType png,  icon #ICON\_OP\_PROF4,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS4  }  },  profileOwner {  mccMnc #MCC\_MNC4  },  profilePolicyRules {  ppr1  }  } |
| METADATA\_OP\_PROF2 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF2,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF2,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS2  }  },  profileOwner {  mccMnc #MCC\_MNC2  }  } |
| METADATA\_OP\_PROF2\_NO\_INSTALL | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF2,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF2,  profileClass operational,  notificationConfigurationInfo {  {  profileManagementOperation {  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS2  }  },  profileOwner {  mccMnc #MCC\_MNC2  }  } |
| METADATA\_OP\_PROF2\_NO\_NOTIF | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF2,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF2,  profileClass operational,  profileOwner {  mccMnc #MCC\_MNC2  }  } |
| METADATA\_OP\_PROF1\_NO\_INSTALL | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  {  profileManagementOperation {  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  }  } |
| METADATA\_OP\_PROF2\_TEST\_DP\_ADDRESS1 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF2,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF2,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC2  }  } |
| METADATA\_OP\_PROF3 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF3,  serviceProviderName #SP\_NAME3,  profileName #NAME\_OP\_PROF3,  iconType png,  icon #ICON\_OP\_PROF3,  profileClass operational,  profileOwner {  mccMnc #MCC\_MNC2  },  profilePolicyRules { ppr2 }  } |
| METADATA\_OP\_PROF4 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF4,  serviceProviderName #SP\_NAME4,  profileName #NAME\_OP\_PROF4,  iconType png,  icon #ICON\_OP\_PROF4,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS4  }  },  profileOwner {  mccMnc #MCC\_MNC4  },  profilePolicyRules {  ppr1  }  } |
| METADATA\_OP\_PROF5 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF5,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF5,  iconType png,  icon #ICON\_OP\_PROF5,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  }  } |
| METADATA\_OP\_PROF6 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF6,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF6,  iconType png,  icon #ICON\_OP\_PROF6,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS2  }  },  profileOwner {  mccMnc #MCC\_MNC2  }  } |
| METADATA\_OP\_PROF7 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF7,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF7,  iconType png,  icon #ICON\_OP\_PROF7,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS8  }  },  profileOwner {  mccMnc #MCC\_MNC2  },  profilePolicyRules {  ppr2  }  } |
| METADATA\_OP\_PROF8 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF8,  serviceProviderName #SP\_NAME8,  profileName #NAME\_OP\_PROF8,  iconType png,  icon #ICON\_OP\_PROF8,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS8  }  },  profileOwner {  mccMnc #MCC\_MNC2  },  profilePolicyRules {  ppr2  }  } |
| METADATA\_OP\_PROF9 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF9,  serviceProviderName #SP\_NAME9,  profileName #NAME\_OP\_PROF9,  profileOwner {  mccMnc #MCC\_MNC9,  gid1 #GID1,  gid2 #GID2  },  profilePolicyRules {  ppr2  }  } |
| METADATA\_OP1\_GID1GID2\_PRESENT | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileOwner {  mccMnc #MCC\_MNC1,  gid1 #GID1,  gid2 #GID2  },  profilePolicyRules {ppr2}  } |
| METADATA\_OP9\_GID1GID2\_MISSING | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF9,  serviceProviderName #SP\_NAME9,  profileName #NAME\_OP\_PROF9,  profileOwner {  mccMnc #MCC\_MNC9  }  } |
| METADATA\_PPR\_NO\_OWNER | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profilePolicyRules {ppr2}  } |
| METADATA\_SERVICE\_SPECIFIC\_STORED | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1},  serviceSpecificDataStoredInEuicc #VENDOR\_SPECIFIC\_EXTENSION1  } |
| METADATA\_SERVICE\_SPECIFIC\_NOT\_STORED | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1},  serviceSpecificDataNotStoredInEuicc #VENDOR\_SPECIFIC\_EXTENSION2  } |
| METADATA\_SERVICE\_SPECIFIC\_STORED\_AND\_NOT\_STORED | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1},  serviceSpecificDataStoredInEuicc #VENDOR\_SPECIFIC\_EXTENSION1,  serviceSpecificDataNotStoredInEuicc #VENDOR\_SPECIFIC\_EXTENSION2  } |
| METADATA\_WILDCARD | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileOwner {  mccMnc #MCC\_MNC\_WILDCARD  },  profilePolicyRules {ppr2}  } |
| METADATA\_WITH\_JPG | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType jpg,  icon #ICON\_JPG  } |
| METADATA\_WITH\_NOTIFS | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS3  },  { profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS2  },  { profileManagementOperation {  notificationEnable  },  notificationAddress #TEST\_DP\_ADDRESS2  },  { profileManagementOperation {  notificationEnable  },  notificationAddress #TEST\_DP\_ADDRESS3  },  { profileManagementOperation {  notificationDisable  },  notificationAddress #TEST\_DP\_ADDRESS3  },  { profileManagementOperation {  notificationDisable  },  notificationAddress #TEST\_DP\_ADDRESS4  },  { profileManagementOperation {  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  },  { profileManagementOperation {  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS3  }  }  } |
| METADATA\_WITH\_PPR1\_PPR2 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1,ppr2}  } |
| METADATA\_WITH\_PPR2 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr2}  } |
| METADATA\_WITH\_PPRS\_AND\_ICON | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1,ppr2}  } |
| METADATA\_WITHOUT\_ICON | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType jpg  } |
| REPLACE\_S\_KEYS\_REQ | req ReplaceSessionKeysRequest ::={  initialMacChainingValue <PPK\_INIT\_MAC>,  ppkEnc <PPK\_ENC>,  ppkCmac <PPK\_MAC>  } |
| REPLACE\_S\_KEYS\_REQ\_INV\_SIZE | req ReplaceSessionKeysRequest ::={  initialMacChainingValue #PPK\_INIT\_MAC\_INV\_SIZE,  ppkEnc #PPK\_ENC\_INV\_SIZE,  ppkCmac #PPK\_MAC\_INV\_SIZE  } |
| S\_INIT\_SC\_PROF1 | req InitialiseSecureChannelRequest ::={  remoteOpId #REMOTE\_OP\_ID\_INSTALL,  transactionId <S\_TRANSACTION\_ID>,  controlRefTemplate {  keyType #KEY\_TYPE,  keyLen #KEY\_LENGTH,  hostId #HOST\_ID  },  smdpOtpk <OTPK\_S\_SM\_DP+\_ECKA>,  smdpSign <S\_SM\_DP+\_SIGN>  } |
| SMDP\_METADATA\_ABS | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1 } |
| SMDP\_METADATA\_ALL | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  {   profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #IUT\_SM\_DP\_ADDRESS  }  },  profileOwner {   mccMnc #MCC\_MNC1  },  profilePolicyRules { ppr1, ppr2 } } |
| SMDP\_METADATA\_NON\_ASCII | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME\_NON\_ASCII,  profileName #NAME\_OP\_PROF1\_NON\_ASCII } |
| SMDP\_METADATA\_NOTIF\_MULTI | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  notificationConfigurationInfo {  {   profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #IUT\_SM\_DP\_ADDRESS  },  {   profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1   }  } } |
| SMDP\_METADATA\_OP\_PROF1\_EN | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationEnable  },  notificationAddress #IUT\_SM\_DP\_ADDRESS  }  }  } |
| SMDP\_METADATA\_OP\_PROF1\_PPR2 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileClass operational,  profileOwner {   mccMnc #MCC\_MNC1  },  profilePolicyRules { ppr2 } } |
| SMDP\_METADATA\_PN\_LONG | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF\_LONG } |
| SMDP\_METADATA\_SPN\_LONG | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME\_LONG,  profileName #NAME\_OP\_PROF1 } |

D.2 ES9+ Requests And Responses

D.2.1 ES9+ Requests

|  |  |  |
| --- | --- | --- |
| Name | Content | |
| AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1  #CTX\_PARAMS1\_ACT\_CODE  },  euiccSignature1  <EUICC\_SIGNATURE1>,  euiccCertificate  #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_ACT\_CODE\_UC\_OK\_EID2 | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1  #CTX\_PARAMS1\_ACT\_CODE  },  euiccSignature1  <EUICC\_SIGNATURE1>,  euiccCertificate  #CERT\_EUICC\_ECDSA\_EID2,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_ACT\_CODE\_2\_UC\_OK | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1  #CTX\_PARAMS1\_ACT\_CODE\_2  },  euiccSignature1  <EUICC\_SIGNATURE1>,  euiccCertificate  #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_OK\_eUICC\_EXT | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2  #S\_EUICC\_INFO2\_EXT  ctxParams1  #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_OK\_UICC\_EXT | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2\_UICC\_EXT  ctxParams1  #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_OK\_DEVICE\_EXT | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2\_DEV\_EXT  ctxParams1  #CTX\_PARAMS1\_DEVICE\_INFO\_EXT  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_1\_3\_8 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_EID2,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_4\_8 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2   #S\_EUICC\_INFO2\_INSUF\_MEM\_ERROR,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_BC\_cA | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_BC\_cA } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_BC\_PLC | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_BC\_PLC } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_CP | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_CP } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_EX\_KU | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_KU } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_1\_SIG | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_SIG } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_2\_6\_3 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_EXPIRED } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_EX\_CP | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_EX\_CP,  eumCertificate   #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_EX\_KU | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_EX\_KU,  eumCertificate   #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_SIG | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_SIG,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_SUB\_ORG | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_SUB\_ORG,  eumCertificate   #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_1\_SUB\_SN | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_SUB\_SN,  eumCertificate   #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_3\_6\_3 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA\_EXPIRED,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_6\_1\_CHA | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE\_INVALID>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_1\_6\_1\_SIG | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1\_INVALID>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_2\_5\_4\_3 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2\_PPR2,   ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_10\_1\_3\_9 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <INVALID\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_8\_11\_1\_3\_9 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_UNKNOWN } | |
| AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2  ctxParams1  #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_SMDP\_MATCHING\_ID\_EMPTY | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_SMDP\_MATCHING\_ID\_OMITTED | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress  #IUT\_SM\_DP\_ADDRESS,  serverChallenge  <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2  ctxParams1  #CTX\_PARAMS1\_MATCHING\_ID\_OMITTED  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| AUTH\_SERVER\_RESP\_SMDS\_UC\_OK | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1 #CTX\_PARAMS1\_SMDS   },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } | |
| AUTH\_SERVER\_RESP\_SMDS\_UC\_OK\_EID2 | resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId  <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DP\_ADDRESS,  serverChallenge   <SMDP\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1 #CTX\_PARAMS1\_SMDS   },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA\_EID2,  eumCertificate #CERT\_EUM\_ECDSA } | |
| CS\_RESP\_ERROR\_8\_1\_6\_1 | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason postponed   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE\_INVALID> } | |
| CS\_RESP\_ERROR\_8\_8\_3\_10 | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid <INVALID\_SM\_DP\_OID>,  reason postponed   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE> } | |
| CS\_RESP\_ERROR\_8\_10\_1\_3\_9 | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <INVALID\_TRANSACTION\_ID>,   smdpOid #IUT\_SM\_DP\_OID,  reason postponed   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE> } | |
| CS\_RESP\_OK\_EU\_REJ | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason endUserRejection   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } | |
| CS\_RESP\_OK\_L\_BPP\_EXE\_ERROR | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason loadBppExecutionError   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } | |
| CS\_RESP\_OK\_M\_DATA\_MISMATCH | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason metadataMismatch   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } | |
| CS\_RESP\_OK\_POSTPONED | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason postponed  },  euiccCancelSessionSignature  <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } | |
| CS\_RESP\_OK\_PPR\_NOT\_ALLOWED | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason pprNotAllowed   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } | |
| CS\_RESP\_OK\_TIMEOUT | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason timeout  },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } | |
| CS\_RESP\_OK\_UNDEFINED | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #IUT\_SM\_DP\_OID,  reason undefinedReason   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } | |
| CTX\_PARAMS1\_ACT\_CODE | ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {  matchingId #MATCHING\_ID\_1,  deviceInfo #S\_DEVICE\_INFO  } | |
| CTX\_PARAMS1\_ACT\_CODE\_2 | ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {  matchingId #MATCHING\_ID\_2,  deviceInfo #S\_DEVICE\_INFO  } | |
| CTX\_PARAMS1\_MATCHING\_ID\_EMPTY | ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {  matchingId #MATCHING\_ID\_EMPTY,  deviceInfo #S\_DEVICE\_INFO } | |
| CTX\_PARAMS1\_SMDS | ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {  matchingId <MATCHING\_ID\_EVENT>,  deviceInfo #S\_DEVICE\_INFO } | |
| EUICC\_FIRMWARE\_VER | 0x01 00 00 | |
| EXT\_CARD\_RESOURCE\_LIMITED\_SPACE | The Extended Card Resource Information according to ETSI TS 102 226 and set as:  0x81 <L> #INSTALLED\_PROFILES  0x82 <L> #NON\_VOLATILE\_MEM\_LIMITED\_SPACE  0x83 <L> #S\_VOLATILE\_MEM | |
| INITIATE\_AUTH\_DS\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDS\_SIGNED1>,  "serverSignature1" :  <S\_SMDS\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" :  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" :  #CERT\_S\_SM\_DSauth\_ECDSA  }  -- NOTE: select the CI as defined in the note in the chapter 2.1.4 of SGP.23 | |
| INITIATE\_AUTH\_DS\_OK\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDS\_SIGNED\_ADDR1>,  "serverSignature1" :  <S\_SMDS\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" :  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" :  #CERT\_S\_SM\_DSauth\_ECDSA  }  -- NOTE: select the CI as defined in the note in the chapter 2.1.4 of SGP.23 | |
| INITIATE\_AUTH\_INV\_CERT\_DS | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDS\_SIGNED1>,  "serverSignature1" : <S\_SMDS\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" : <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  -- NOTE: select the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> "serverCertificate" : #CERT\_S\_SM\_DSauth\_INV\_SIGN } | |
| INITIATE\_AUTH\_INV\_CI\_DS | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDS\_SIGNED1>,  "serverSignature1" : <S\_SMDS\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" : #CI\_PK\_ID\_INV,   "serverCertificate" : #CERT\_S\_SM\_DSauth\_ECDSA  -- NOTE: select and choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> } | |
| INITIATE\_AUTH\_INV\_SIGN\_DS | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDS\_SIGNED1>,  "serverSignature1" : <S\_SMDS\_SIGNATURE\_INV>,  "euiccCiPKIdTobeUsed" : <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" : #CERT\_S\_SM\_DSauth\_ECDSA }  -- NOTE: select the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | |
| INITIATE\_AUTH\_INV\_SMDS\_ADDRESS | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDS\_SIGNED\_INV\_ADDR>,  "serverSignature1" : <S\_SMDS\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" : <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" : #CERT\_S\_SM\_DSauth\_ECDSA }  -- NOTE: select the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and choose the #CERT\_S\_SM\_DSauth\_ECDSA leading to the same Root CI certificate | |
| INITIATE\_AUTH\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDP\_SIGNED1>,  "serverSignature1" :  <S\_SMDP\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" :  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" :  #CERT\_S\_SM\_DPauth\_ECDSA  }  -- NOTE: select the CI as defined in the note in the chapter 2.1.4 of SGP.23 | |
| INITIATE\_AUTH\_INV\_CERT | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDP\_SIGNED1>,  "serverSignature1" :  <S\_SMDP\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" :  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,-- NOTE:  select the CI Key ID in highest  priority from the  <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>  "serverCertificate" :  #CERT\_S\_SM\_DPauth\_INV\_SIGN  } | |
| INITIATE\_AUTH\_INV\_CI | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDP\_SIGNED1>,  "serverSignature1" :  <S\_SMDP\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" : #CI\_PKI\_ID2,  "serverCertificate" :  #CERT\_S\_SM\_DPauth\_ECDSA -- NOTE:  select and choose the  #CERT\_S\_SM\_DPauth\_ECDSA leading to  the CI Key ID in highest priority from  the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>  } | |
| INITIATE\_AUTH\_INV\_OID | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDP\_SIGNED1>,  "serverSignature1" :  <S\_SMDP\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" :  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" :  #CERT\_S\_SM\_DP2auth\_ECDSA  }  -- NOTE: select the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>  -- NOTE: serverSignature1 SHALL be calculated correctly, using the secret key related to CERT\_S\_SM\_DP2auth\_ECDSA. | |
| INITIATE\_AUTH\_INV\_SIGN | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" : <S\_SMDP\_SIGNED1>,  "serverSignature1" :  <S\_SMDP\_SIGNATURE\_INV>,  "euiccCiPKIdTobeUsed" :  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" :  #CERT\_S\_SM\_DPauth\_ECDSA  }  -- NOTE: select the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate | |
| INITIATE\_AUTH\_INV\_SMDP+\_ADDRESS | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "serverSigned1" :  <S\_SMDP\_SIGNED\_INV\_ADDR>,  "serverSignature1" :  <S\_SMDP\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" :  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  "serverCertificate" :  #CERT\_S\_SM\_DPauth\_ECDSA  }  -- NOTE: select the CI Key ID in highest priority from the <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING> and choose the #CERT\_S\_SM\_DPauth\_ECDSA leading to the same Root CI certificate  -- NOTE: serverSignature1 SHALL be calculated correctly, using <S\_SMDP\_SIGNED\_INV\_ADDR>. | |
| MATCHING\_ID\_EMPTY |  | |
| NON\_VOLATILE\_MEM\_LIMITED\_SPACE | '0x00 01' | |
| PENDING\_NOTIF\_DEL1 | response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDelete  },  notificationAddress  #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_DEL2 | response PendingNotification ::= otherSignedNotification :  {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation  {  notificationDelete  },  notificationAddress  #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF2  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_DEL4 | response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDelete  },  notificationAddress  #TEST\_DP\_ADDRESS4,  iccid #ICCID\_OP\_PROF4  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_DEL5 | response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDelete  },  notificationAddress  #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF5  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_DEL6 | response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDelete  },  notificationAddress  #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF6  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_DIS1 | response PendingNotification ::= otherSignedNotification : {    tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDisable  },  notificationAddress  #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_DIS5 | response PendingNotification ::= otherSignedNotification : {    tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDisable  },  notificationAddress  #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF5  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_DIS8 | response PendingNotification ::= otherSignedNotification : {    tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDisable  },  notificationAddress  #TEST\_DP\_ADDRESS8,  iccid #ICCID\_OP\_PROF8  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_EN1 | response PendingNotification ::= otherSignedNotification : {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationEnable  },  notificationAddress  #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_EN2 | response PendingNotification ::= otherSignedNotification : {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationEnable  },  notificationAddress  #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF2  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_EN5 | response PendingNotification ::= otherSignedNotification : {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationEnable  },  notificationAddress  #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF5  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PENDING\_NOTIF\_EN6 | response PendingNotification ::= otherSignedNotification : {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationEnable  },  notificationAddress  #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF6  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } | |
| PP\_VERSION | 0x01 00 00 | |
| PREP\_DOWNLOAD\_RESP\_8\_1\_6\_1 | resp PrepareDownloadResponse ::=  downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <BPP\_OTPK\_EUICC\_ECKA>  },  euiccSignature2 <EUICC\_SIGNATURE2\_INVALID>  } | |
| PREP\_DOWNLOAD\_RESP\_8\_2\_7\_3\_8 | resp PrepareDownloadResponse ::=  downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <BPP\_OTPK\_EUICC\_ECKA>,  hashCc <S\_HASHED\_CC\_ERROR>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } | |
| PREP\_DOWNLOAD\_RESP\_8\_10\_1\_3\_9 | resp PrepareDownloadResponse ::=  downloadResponseOk : {  euiccSigned2 {  transactionId <INVALID\_TRANSACTION\_ID>,  euiccOtpk <BPP\_OTPK\_EUICC\_ECKA>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } | |
| PREP\_DOWNLOAD\_RESP | resp PrepareDownloadResponse ::=   downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <BPP\_OTPK\_EUICC\_ECKA>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } | |
| PREP\_DOWNLOAD\_RESP\_CC | resp PrepareDownloadResponse ::=   downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <BPP\_OTPK\_EUICC\_ECKA>,  hashCc <S\_HASHED\_CC>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } | |
| PREP\_DOWNLOAD\_RESP\_NEW\_OTPK | resp PrepareDownloadResponse ::=  downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <OTPK\_EUICC\_ECKA\_NEW>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } | |
| PREP\_DOWNLOAD\_RESP\_NEW\_OTPK\_CC | resp PrepareDownloadResponse ::=  downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <OTPK\_EUICC\_ECKA\_NEW>,  hashCc <S\_HASHED\_CC>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } | |
| PROFILE\_VERSION | 0x02 01 00 | |
| RSP\_CAPABILITY | rspCapability RspCapability ::= {   additionalProfile, rpmSupport,   testProfileSupport } | |
| RSP\_CAPABILITY\_EXT | rspCapability RspCapability ::= {   additionalProfile, rpmSupport,   testProfileSupport, deviceInfoExtensibilitySupport,  serviceSpecificDataSupport  } | |
| S\_EUICC\_INFO2\_INSUF\_MEM\_ERROR | euiccInfo2 EUICCInfo2 ::= {  profileVersion #PROFILE\_VERSION,  svn #RSP\_SVN\_H,  euiccFirmwareVer #EUICC\_FIRMWARE\_VER,  extCardResource   #EXT\_CARD\_RESOURCE\_LIMITED\_SPACE,  uiccCapability #UICC\_CAPABILITY,  rspCapability #RSP\_CAPABILITY,  euiccCiPKIdListForVerification   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1},  euiccCiPKIdListForSigning   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1},  ppVersion #PP\_VERSION,  sasAcreditationNumber   #SAS\_ACREDITATION\_NUMBER } | |
| S\_EUICC\_INFO2\_PPR2 | euiccInfo2 EUICCInfo2 ::= {  profileVersion #PROFILE\_VERSION,  svn #RSP\_SVN\_H,  euiccFirmwareVer   #EUICC\_FIRMWARE\_VER,  extCardResource   #S\_EXT\_CARD\_RESOURCE,  uiccCapability #UICC\_CAPABILITY,  rspCapability #RSP\_CAPABILITY,  euiccCiPKIdListForVerification   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION\_1},  euiccCiPKIdListForSigning   {#EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING\_1},  forbiddenProfilePolicyRules { ppr2 },  ppVersion #PP\_VERSION,  sasAcreditationNumber   #SAS\_ACREDITATION\_NUMBER } | |
| S\_EXT\_CARD\_RESOURCE | The Extended Card Resource Information according to ETSI TS 102 226:  0x81 <L> #INSTALLED\_PROFILES  0x82 <L> #S\_NON\_VOLATILE\_MEM  0x83 <L> #S\_VOLATILE\_MEM | |
| S\_NON\_VOLATILE\_MEM | 0xA0 00 | |
| S\_PN\_PIR\_OK1 | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_INVALID\_TRANS\_ID | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <INVALID\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress   #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_INCORRECT\_INPUT\_VALUES | response PendingNotification ::= profileInstallationResult : profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId configureISDP,  errorReason incorrectInputValues  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_INVALID\_SIGN | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId initialiseSecureChannel,  errorReason invalidSignature }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_UNSUPPORTED\_CRT | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId initialiseSecureChannel,  errorReason unsupportedCrtValues  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_UNSUP\_REMOTE\_OP\_TYPE | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId initialiseSecureChannel,  errorReason   unsupportedRemoteOperationType  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_UNSUP\_PROFILE\_CLASS | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason unsupportedProfileClass  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_SCP03T\_STRUCTURE\_ERROR | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason scp03tStructureError  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_SCP03T\_SECURITY\_ERROR | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId replaceSessionKeys,  errorReason scp03tSecurityError  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_ICCID\_ALREADY\_EXISTS | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason   installFailedDueToIccidAlreadyExistsOnEuicc  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_INSUFFICIENT\_MEMORY | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason installFailedDueToInsufficientMemoryForProfile  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_INSTALL\_INTERRUPTION | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason  installFailedDueToInterruption  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_PE\_PROCESSING\_ERROR | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId loadProfileElements,  errorReason  installFailedDueToPEProcessingError  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_DATA\_MISMATCH | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId loadProfileElements,  errorReason  installFailedDueToDataMismatch  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_TEST\_PROFILE\_INVALID\_NAA\_KEY | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId loadProfileElements,  errorReason testProfileInstallFailedDueToInvalidNaaKey  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_PPR\_NOT\_ALLOWED | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason pprNotAllowed  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PN\_PIR\_UNKNOWN\_ERROR | response PendingNotification ::= profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  smdpOid #IUT\_SM\_DP\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason   installFailedDueToUnknownError  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| S\_PENDING\_NOTIF\_OTHER\_INST1 | response PendingNotification ::= otherSignedNotification :  {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation  {  notificationInstall  },  notificationAddress   #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature   <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| S\_PENDING\_NOTIF\_EN1 | response PendingNotification ::= otherSignedNotification : {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationEnable  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature   <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| S\_PENDING\_NOTIF\_DIS1 | response PendingNotification ::= otherSignedNotification : {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDisable  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature   <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| S\_PENDING\_NOTIF\_DE1 | response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationDelete  },  notificationAddress #IUT\_SM\_DP\_ADDRESS,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature   <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| S\_SMDP\_SIGNED2 | req SmdpSigned2 ::= {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  } | |
| S\_SMDP\_SIGNED2\_CC | req SmdpSigned2 ::= {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag TRUE  } | |
| S\_SMDP\_SIGNED2\_INV\_TRANSACTION\_ID | req SmdpSigned2 ::= {  transactionId <INVALID\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  } | |
| S**\_**VOLATILE\_MEM | '0x01 00' | |
| SAS\_ACREDITATION\_NUMBER | GSMA\_SAS\_123456789 | |
| UICC\_CAPABILITY | uiccCapability UICCCapability ::= {   contactlessSupport, usimSupport,   isimSupport,   akaMilenage, akaTuak128,   gbaAuthenUsim, eapClient,   javacard, multipleUsimSupport  } | |
| UICC\_CAPABILITY\_EXT | uiccCapability UICCCapability ::= {   contactlessSupport, usimSupport,   isimSupport,   akaMilenage, akaTuak128,   gbaAuthenUsim, eapClient,   javacard, multipleUsimSupport, berTlvFileSupport, dfLinkSupport, catTp, getIdentity, profile-a-x25519, profile-b-p256, suciCalculatorApi, unknownServiceSupport  }  Note: the definition of UICCCapability used above is equivalent to the definition in SGP.22 v2.3 (specific version of [2]) with the additional of a further field called “unknownServiceSupport” after the “suciCalculatorApi” field. | |

D.2.2 ES9+ Responses

|  |  |
| --- | --- |
| Name | Content |
| AUTH\_CLIENT\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "profileMetadata" :  #METADATA\_OP\_PROF1,  "smdpSigned2" : #S\_SMDP\_SIGNED2,  "smdpSignature2" :   <S\_SM\_DP+\_SIGNATURE2>,  "smdpCertificate" :  #CERT\_S\_SM\_DPpb\_ECDSA } |
| AUTH\_CLIENT\_OK\_CC | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "profileMetadata" :  #METADATA\_OP\_PROF1,  "smdpSigned2" : #S\_SMDP\_SIGNED2\_CC,  "smdpSignature2" :   <S\_SM\_DP+\_SIGNATURE2>,  "smdpCertificate" :  #CERT\_S\_SM\_DPpb\_ECDSA } |
| AUTH\_CLIENT\_INV\_PB\_CERT | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "profileMetadata" :  #METADATA\_OP\_PROF1,  "smdpSigned2" : #S\_SMDP\_SIGNED2,  "smdpSignature2" :   <S\_SM\_DP+\_SIGNATURE2>,  "smdpCertificate" :  #CERT\_S\_SM\_DPpb\_INV\_SIGN } |
| AUTH\_CLIENT\_INV\_CI | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "profileMetadata" :  #METADATA\_OP\_PROF1,  "smdpSigned2" : #S\_SMDP\_SIGNED2,  "smdpSignature2" :   <S\_SM\_DP+\_SIGNATURE2>,  "smdpCertificate" :  #CERT\_S\_SM\_DP2pb\_ECDSA } |
| AUTH\_CLIENT\_INV\_SIGN | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "profileMetadata" :  #METADATA\_OP\_PROF1,  "smdpSigned2" : #S\_SMDP\_SIGNED2,  "smdpSignature2" :   <S\_SM\_DP+\_SIGNATURE2>,  "smdpCertificate" :  #CERT\_S\_SM\_DPpb\_ECDSA }  The <S\_SM\_DP+\_SIGNATURE2> SHALL NOT be computed using the #SK\_S\_SM\_DPpb\_ECDSA *but SHALL have the same length as for a valid signature* |
| AUTH\_CLIENT\_INV\_TRANSACTION\_ID | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" :  <S\_TRANSACTION\_ID>,  "profileMetadata" :  #METADATA\_OP\_PROF1,  "smdpSigned2" : #S\_SMDP\_SIGNED2\_INV\_TRANSACTION\_ID,  "smdpSignature2" :   <S\_SM\_DP+\_SIGNATURE2>,  "smdpCertificate" :  #CERT\_S\_SM\_DPpb\_ECDSA } |
| CS\_OK\_EU\_LOAD\_BPP\_ERROR | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason loadBppExecutionError   },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } |
| CS\_OK\_EU\_POSTPONED | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason postponed  },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } |
| CS\_OK\_EU\_REJ | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason endUserRejection  },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } |
| CS\_OK\_PPR\_NOT\_ALLOWED | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason pprNotAllowed  },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } |
| CS\_OK\_TIMEOUT | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason timeout  },  euiccCancelSessionSignature   <EUICC\_CANCEL\_SESSION\_SIGNATURE>  } |
| GET\_BPP\_LOAD\_ERROR | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "boundProfilePackage" : BoundProfilePackage {  #S\_INIT\_SC\_PROF1,  firstSequenceOf87 {  #CONF\_ISDP\_PROF1  },  sequenceOf88 {  <METADATA\_OP\_PROF1\_SEG>  …  <METADATA\_OP\_PROF1\_SEG>   }  } }  NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1.  NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata. |
| GET\_BPP\_LOAD\_ERROR\_UNKNOWN\_TAG | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "boundProfilePackage" {  #S\_INIT\_SC\_PROF1,  #UNKNOWN\_BPP\_SEGMENT,  firstSequenceOf87 {  #CONF\_ISDP\_PROF1  },  sequenceOf88 {  <METADATA\_OP\_PROF1\_SEG>  …  <METADATA\_OP\_PROF1\_SEG>  },  sequenceOf86 {  <PPP\_OP\_PROF1\_SEG\_SK>  …  <PPP\_OP\_PROF1\_SEG\_SK>  }  }  }  NOTE 1: boundProfilePackage is encoded as base64 therefore the test tool shall decode boundProfilePackage to access the ASN.1.  NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata. |
| GET\_BPP\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "boundProfilePackage" : BoundProfilePackage {  #S\_INIT\_SC\_PROF1,  firstSequenceOf87 {  #CONF\_ISDP\_PROF1  },  sequenceOf88 {  <METADATA\_OP\_PROF1\_SEG>  …  <METADATA\_OP\_PROF1\_SEG>  },  sequenceOf86 {  <PPP\_OP\_PROF1\_SEG\_SK>  …  <PPP\_OP\_PROF1\_SEG\_SK>  }  }  }  NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1.  NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata. |
| GET\_BPP\_OK\_PPK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "boundProfilePackage" : BoundProfilePackage {  #S\_INIT\_SC\_PROF1,  firstSequenceOf87 {  0x87 <L> #CONF\_ISDP\_PROF1  },  sequenceOf88 {  <METADATA\_OP\_PROF1\_SEG>  …  <METADATA\_OP\_PROF1\_SEG>  },  secondSequenceOf87 {  0x87 <L> #REPLACE\_S\_KEYS\_REQ  },  sequenceOf86 {  <PPP\_OP\_PROF1\_SEG\_SK>  …  <PPP\_OP\_PROF1\_SEG\_SK>  }  }  } |
| GET\_BPP\_INV | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "boundProfilePackage" : BoundProfilePackage {  #S\_INIT\_SC\_PROF1,  firstSequenceOf87 {  0x87 <L> #CONF\_ISDP\_PROF1  },  sequenceOf88 {  <METADATA\_OP\_PROF1\_SEG>  …  <METADATA\_OP\_PROF1\_SEG>  },  sequenceOf86 {  <PPP\_OP\_PROF1\_SEG\_SK\_INV>  …  <PPP\_OP\_PROF1\_SEG\_SK\_INV>  }  }  } |
| PENDING\_NOTIF\_INST\_ADDRESS2 | response PendingNotification ::= otherSignedNotification :  {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation  {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| PENDING\_NOTIF\_INST1 | response PendingNotification ::= otherSignedNotification :  {  tbsOtherNotification {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation  {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  },  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_AUTH\_CLIENT\_META\_ABS | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_ABS,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_META\_ALL | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_ALL,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_META\_LARGE\_ICON | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_OP\_PROF1\_2\_SEG,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_META\_NON\_ASCII | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_NON\_ASCII,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_META\_NOTIF\_MULTI | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_NOTIF\_MULTI,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_META\_PN\_LONG | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_PN\_LONG,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_META\_SPN\_LONG | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_SPN\_LONG,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" :   #SMDP\_METADATA\_OP\_PROF1,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_OK\_ALL\_NOTIF | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" :   #SMDP\_METADATA\_ALL\_NOTIF,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_OK\_CC | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" :   #SMDP\_METADATA\_OP\_PROF1,  "smdpSigned2" : #SMDP\_SIGNED2\_CC,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_OK\_EN | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" :   #SMDP\_METADATA\_OP\_PROF1\_EN,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_OK\_PPR2 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" :   #SMDP\_METADATA\_OP\_PROF1\_PPR2,  "smdpSigned2" : #SMDP\_SIGNED2,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_RETRY\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" :   #SMDP\_METADATA\_OP\_PROF1,  "smdpSigned2" : #SMDP\_SIGNED2\_RETRY,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_CLIENT\_RETRY\_OK\_CC | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_AC>,  "profileMetadata" : #SMDP\_METADATA\_OP\_PROF1,  "smdpSigned2" : #SMDP\_SIGNED2\_CC\_RETRY,  "smdpSignature2" : <SMDP\_SIGNATURE2>,  "smdpCertificate" : #CERT\_SM\_DPpb\_ECDSA } |
| R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO | resp AuthenticateServerResponse ::authenticateResponseOk : {  euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present but not the values  ctxParams1 #CTX\_PARAMS1\_MATCH\_ID\_DEV\_INFO  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_AUTH\_SERVER\_DS\_MATCH\_ID\_DEV\_INFO\_1 | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_DS\_ADDRESS1,  serverChallenge <S\_SMDS\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present but not the values  ctxParams1 #CTX\_PARAMS1\_MATCH\_ID\_DEV\_INFO  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_AUTH\_SERVER\_MATCH\_ID\_DEV\_INFO | resp AuthenticateServerResponse ::= authenticateResponseOk {  euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present but not the values  ctxParams1 #CTX\_PARAMS1\_MATCH\_ID\_DEV\_INFO  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_GET\_BPP\_RESP\_OP1\_PPK  (Pre-generated PPP for Profiles) | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId": <TRANSACTION\_ID\_GBPP>,  "boundProfilePackage" : BoundProfilePackage {  #INIT\_SC\_PROF1,  firstSequenceOf87 {  <CONF\_ISDP\_PROF1\_ENC>  },  sequenceOf88 {  <SMDP\_METADATA\_SEG\_MAC>  …  <SMDP\_METADATA\_SEG\_MAC>  },  secondSequenceOf87 {  <REPLACE\_S\_KEYS\_REQ\_ENC>  },  sequenceOf86 {  <PPP\_OP\_PROF1\_SEG\_PPK>  …  <PPP\_OP\_PROF1\_SEG\_PPK>  }  }  }  NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1.  NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata. |
| R\_GET\_BPP\_RESP\_OP1\_SK  (Dynamically-generated PPP for Profiles) | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId": <TRANSACTION\_ID\_GBPP>,  "boundProfilePackage" : BoundProfilePackage {  #INIT\_SC\_PROF1,  firstSequenceOf87 {  <CONF\_ISDP\_PROF1\_ENC>  },  sequenceOf88 {  <SMDP\_METADATA\_SEG\_MAC>  …  <SMDP\_METADATA\_SEG\_MAC>  },  sequenceOf86 {  <PPP\_OP\_PROF1\_SEG\_SK>  …  <PPP\_OP\_PROF1\_SEG\_SK>  }  }  }  NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1.  NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata. |
| R\_HTTP\_204\_OK | HTTP/1.1 204 No Content  X-Admin-Protocol: gsma/rsp/v#RSP\_SVN  NOTE: If the HTTP response is being received from the server under test, then the "Content-type" header MAY be present. |
| R\_INITIATE\_AUTH\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_IA>,  "serverSigned1" : #SERVER\_SIGNED1,  "serverSignature1" : <SERVER\_SIGNATURE1>,  "euiccCiPKIdTobeUsed" : #CI\_PKI\_ID1,  "serverCertificate" : #CERT\_SM\_XXauth\_ECDSA } |
| R\_INITIATE\_AUTH\_OK\_2 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <TRANSACTION\_ID\_2>,  "serverSigned1" : #SERVER\_SIGNED1\_2,  "serverSignature1" : <SERVER\_SIGNATURE1\_2>,  "euiccCiPKIdTobeUsed" : #CI\_PKI\_ID1,  "serverCertificate" : #CERT\_SM\_XXauth\_ECDSA } |
| SERVER\_SIGNED1 | For InitiateAuthentication testing XX = IA, and for AuthenticateClient testing XX = AC:  ss1 ServerSigned1 ::= {   transactionId   <TRANSACTION\_ID\_SIGNED\_IA>,  euiccChallenge #S\_EUICC\_CHALLENGE,  serverAddress   #SERVER\_ADDRESS,  serverChallenge <SERVER\_CHALLENGE> } |
| SERVER\_SIGNED1\_2 | ss1\_2 ServerSigned1 ::= {   transactionId <TRANSACTION\_ID\_SIGNED\_2>,  euiccChallenge #S\_EUICC\_CHALLENGE\_2,  serverAddress #SERVER\_ADDRESS,  serverChallenge <SERVER\_CHALLENGE\_2> } |
| SMDP\_METADATA\_ALL\_NOTIF | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #IUT\_SM\_DP\_ADDRESS  }  }  } |
| SMDP\_METADATA\_OP\_PROF1 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  profileClass operational } |
| SMDP\_METADATA\_OP\_PROF1\_2\_SEG | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1\_2\_SEG,  profileClass operational,  notificationConfigurationInfo {  {   profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #IUT\_SM\_DP\_ADDRESS  }  },  profileOwner {   mccMnc #MCC\_MNC1  } } |
| SMDP\_METADATA\_OP\_PROF3 | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF3,  serviceProviderName #SP\_NAME3,  profileName #NAME\_OP\_PROF3,  profileClass operational,  profileOwner {  mccMnc #MCC\_MNC2  },  profilePolicyRules { ppr2 }  } |
| SMDP\_SIGNED2 | smdpSigned2 SmdpSigned2 ::= {  transactionId <TRANSACTION\_ID\_SIGNED\_AC>,  ccRequiredFlag FALSE } |
| SMDP\_SIGNED2\_CC | smdpSigned2 SmdpSigned2 ::= {  transactionId <TRANSACTION\_ID\_SIGNED\_AC>,  ccRequiredFlag TRUE } |
| SMDP\_SIGNED2\_CC\_RETRY | smdpSigned2 SmdpSigned2 ::= {  transactionId <TRANSACTION\_ID\_SIGNED\_AC>,  ccRequiredFlag TRUE,  bppEuiccOtpk <BPP\_OTPK\_EUICC\_ECKA> } |
| SMDP\_SIGNED2\_RETRY | smdpSigned2 SmdpSigned2 ::= {  transactionId <TRANSACTION\_ID\_SIGNED\_AC>,  ccRequiredFlag FALSE,  bppEuiccOtpk <BPP\_OTPK\_EUICC\_ECKA> } |

D.3 ES10x Requests And Responses

D.3.1 ES10x Requests

|  |  |
| --- | --- |
| Name | Content |
| AUTH\_SMDP\_MATCH\_ID | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  },  serverSignature1 <S\_SMDP\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DPauth\_ECDSA,  ctxParams1 #CTX\_PARAMS1\_MATCH\_ID  } |
| AUTH\_SMDP\_IMEI | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  },  serverSignature1 <S\_SMDP\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DPauth\_ECDSA,  ctxParams1 #CTX\_PARAMS1\_IMEI  } |
| AUTH\_SMDP\_INV\_CERT | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  },  serverSignature1 <S\_SMDP\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DPauth\_INV\_SIGN,  ctxParams1 #CTX\_PARAMS1  } |
| AUTH\_SMDP\_INV\_CURV | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  },  serverSignature1 <RANDOM\_SM\_DP+\_SIGN>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DPauth\_INV\_CURVE,  ctxParams1 #CTX\_PARAMS1  } |
| AUTH\_SMDP\_INV\_CHALLENGE | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge #S\_EUICC\_CHALLENGE,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  },  serverSignature1 <S\_SMDP\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DPauth\_ECDSA,  ctxParams1 #CTX\_PARAMS1  } |
| AUTH\_SMDP\_INV\_OID | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  },  serverSignature1 <S\_SMDP\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DPpb\_ECDSA,  ctxParams1 #CTX\_PARAMS1  } |
| AUTH\_SMDS\_IMEI | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>  },  serverSignature1 <S\_SMDS\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DSauth\_ECDSA,  ctxParams1 #CTX\_PARAMS1\_EVENT\_ID\_IMEI  } |
| AUTH\_SMDS\_INV\_CERT | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>  },  serverSignature1 <S\_SMDS\_SIGNATURE1>,  euiccCiPKIdToBeUsed <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DSauth\_INV\_SIGN,  ctxParams1 #CTX\_PARAMS1\_EVENT\_ID  } |
| AUTH\_SMDS\_INV\_CHALLENGE | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge #S\_EUICC\_CHALLENGE,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>  },  serverSignature1 <S\_SMDS\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DSauth\_ECDSA,  ctxParams1 #CTX\_PARAMS1\_EVENT\_ID  } |
| AUTH\_SMDS\_INV\_CURV | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>  },  serverSignature1 <RANDOM\_SM\_DS\_SIGN>,  euiccCiPKIdToBeUsed <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DSauth\_INV\_CURVE,  ctxParams1 #CTX\_PARAMS1\_EVENT\_ID  } |
| AUTHENTICATE\_SMDP | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>  },  serverSignature1 <S\_SMDP\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DPauth\_ECDSA,  ctxParams1 #CTX\_PARAMS1  } |
| AUTHENTICATE\_SMDS | req AuthenticateServerRequest ::= {  serverSigned1 {  transactionId <S\_TRANSACTION\_ID>,  euiccChallenge <EUICC\_CHALLENGE>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>  },  serverSignature1 <S\_SMDS\_SIGNATURE1>,  euiccCiPKIdToBeUsed  <EUICC\_CI\_PK\_ID\_TO\_BE\_USED>,  serverCertificate #CERT\_S\_SM\_DSauth\_ECDSA,  ctxParams1 #CTX\_PARAMS1\_EVENT\_ID  } |
| CANCEL\_SESSION\_INV\_TRANS\_ID | req CancelSessionRequest ::={  transactionId <INVALID\_TRANSACTION\_ID>,  reason endUserRejection  } |
| CANCEL\_SESSION\_REJECT | req CancelSessionRequest ::={  transactionId <S\_TRANSACTION\_ID>,  reason endUserRejection  } |
| CANCEL\_SESSION\_POSTPONED | req CancelSessionRequest ::={  transactionId <S\_TRANSACTION\_ID>,  reason postponed  } |
| CANCEL\_SESSION\_TIMEOUT | req CancelSessionRequest ::={  transactionId <S\_TRANSACTION\_ID>,  reason timeout  } |
| CANCEL\_SESSION\_PPR | req CancelSessionRequest ::={  transactionId <S\_TRANSACTION\_ID>,  reason pprNotAllowed  } |
| CANCEL\_SESSION\_METADATA | req CancelSessionRequest ::={  transactionId <S\_TRANSACTION\_ID>,  reason metadataMismatch  } |
| CANCEL\_SESSION\_LOAD\_BPP | req CancelSessionRequest ::={  transactionId <S\_TRANSACTION\_ID>,  reason loadBppExecutionError  } |
| CANCEL\_SESSION\_UNDEF | req CancelSessionRequest ::={  transactionId <S\_TRANSACTION\_ID>,  reason undefinedReason  } |
| EUICC\_MEMORY\_RESET | req EuiccMemoryResetRequest ::= {  resetOptions {  deleteOperationalProfiles,  resetDefaultSmdpAddress  }  } |
| EUICC\_MEMORY\_RESET\_DEF\_SMDPADDRESS | req EuiccMemoryResetRequest ::= {  resetOptions { resetDefaultSmdpAddress }  } |
| EUICC\_MEMORY\_RESET\_OP\_PRO | req EuiccMemoryResetRequest ::= {  resetOptions { deleteOperationalProfiles }  } |
| GET\_CONF\_OP\_PROF1 | opConfProf1Req ProfileInfoListRequest ::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '4FB8'H  } |
| GET\_EID | getEIDReq GetEuiccDataRequest ::= {  tagList '5A'H  } |
| GET\_EID\_INVALID | getEIDReq GetEuiccDataRequest ::= {  tagList '6B'H  } |
| GET\_EUICC\_CHALLENGE | request GetEuiccChallengeRequest ::= {} |
| GET\_EUICC\_CONFIGURED\_ADDRESSES | request EuiccConfiguredAddressesRequest ::={} |
| GET\_EUICC\_INFO1 | request GetEuiccInfo1Request::= { } |
| GET\_EUICC\_INFO2 | request GetEuiccInfo2Request::= { } |
| GET\_METADATA\_OP\_PROF1 | opConfProf1Req ProfileInfoListRequest ::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '5A9192939495B6B799'H  } |
| GET\_METADATA\_OP\_PROF1\_SERVICE\_SPECIFIC | opConfProf1Req ProfileInfoListRequest ::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '5A9192939495B6B799BF22'H  } |
| GET\_NEW\_METADATA | getupdate1Req ProfileInfoListRequest ::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '9192939499'H *-- names, icon and PPRs*  } |
| GET\_NOTIF\_CONF\_OP\_PROF1 | opConfProf1Req ProfileInfoListRequest ::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '5AB6'H  } |
| GET\_PPR\_OP\_PROF1 | opConfProf1Req ProfileInfoListRequest ::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '5A99'H  } |
| GET\_PROFILES\_INFO\_ALL | request ProfileInfoListRequest::= { } |
| GET\_PROFILES\_INFO\_ICCID\_TAGLIST1 | request ProfileInfoListRequest::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '9F70'H *--state* } |
| GET\_PROFILES\_INFO\_ICCID\_TAGLIST2 | request ProfileInfoListRequest::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '93'H *--icon type* } |
| GET\_PROFILES\_INFO\_ICCID\_TAGLIST3 | request ProfileInfoListRequest::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList '95'H *--Profile Class* } |
| GET\_PROFILES\_INFO\_ICCID\_TAGLIST4 | request ProfileInfoListRequest::= {  searchCriteria iccid: #ICCID\_OP\_PROF1,  tagList 'B6'H *--Notification configuration* } |
| GET\_PROFILES\_INFO\_ICCID\_TAGLIST5 | request ProfileInfoListRequest::= {  searchCriteria iccid: #ICCID\_OP\_PROF3,  tagList '99'H *--ppr* } |
| GET\_PROFILES\_INFO\_OPTAGLIST1 | request ProfileInfoListRequest::= {  searchCriteria profileClass: operational,  tagList '5A9F70'H *-- ICCID and State* } |
| GET\_PROFILES\_INFO\_OPTAGLIST2 | request ProfileInfoListRequest::= {  searchCriteria profileClass: operational,  tagList '909F70'H *--Nickname and State* } |
| GET\_PROFILES\_INFO\_OPTAGLIST3 | request ProfileInfoListRequest::= {  searchCriteria profileClass: operational,  tagList '9493'H *--Icon, Icon type* } |
| GET\_PROFILES\_INFO\_OPTAGLIST4 | request ProfileInfoListRequest::= {  searchCriteria profileClass: operational,  tagList '949F70'H *--Icon, state* } |
| GET\_PROFILES\_INFO\_PROFCLASS | request ProfileInfoListRequest::= {  searchCriteria profileClass: operational } |
| GET\_PROFILES\_INFO\_TAGLIST\_ICCID | request ProfileInfoListRequest::= {  tagList '5A'H } |
| GET\_PROFILES\_INFO\_TAGLIST\_ICON | request ProfileInfoListRequest::= {  tagList '94'H } |
| GET\_PROFILES\_INFO\_TAGLIST\_ISDPAID | request ProfileInfoListRequest::= {  tagList '4F'H } |
| GET\_PROFILES\_INFO\_TAGLIST\_PROFILE\_NAME | request ProfileInfoListRequest::= {  tagList '92'H } |
| GET\_PROFILES\_INFO\_TAGLIST\_PROFILE\_NICKNAME | request ProfileInfoListRequest::= {  tagList '90'H } |
| GET\_PROFILES\_INFO\_TAGLIST\_PROFILE\_OWNER | request ProfileInfoListRequest::= {  tagList 'B7'H } |
| GET\_PROFILES\_INFO\_TAGLIST\_SMDP\_PROP\_DATA | request ProfileInfoListRequest::= {  tagList 'B8'H } |
| GET\_PROFILES\_INFO\_TAGLIST\_SP\_NAME | request ProfileInfoListRequest::= {  tagList '91'H } |
| GET\_PROFILES\_INFO\_TAGLIST1 | request ProfileInfoListRequest::= {  tagList '5A9F70'H *-- ICCID and State* } |
| GET\_PROFILES\_INFO\_TAGLIST2 | request ProfileInfoListRequest::= {  tagList '909F70'H *--Nickname and State* } |
| GET\_PROFILES\_INFO\_TAGLIST3 | request ProfileInfoListRequest::= {  tagList '9493'H  *--Icon, Icon type* } |
| GET\_PROFILES\_INFO\_TAGLIST4 | request ProfileInfoListRequest::= {  tagList '949F70'H *--Icon, state* } |
| GET\_PROFILES\_OWNERS | request ProfileInfoListRequest::= {  tagList 'B7'H } |
| GET\_RAT | request GetRatRequest ::={} |
| LIST\_NOTIF\_ALL | request ListNotificationRequest ::= {  profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  } } |
| LIST\_NOTIF\_OMITTED | request ListNotificationRequest ::= {} |
| LIST\_NOTIF\_NONE | request ListNotificationRequest ::= {  profileManagementOperation {} } |
| LIST\_NOTIF\_INSTALL | request ListNotificationRequest ::= {  profileManagementOperation {  notificationInstall  } } |
| LIST\_NOTIF\_ENABLE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationEnable  } } |
| LIST\_NOTIF\_DISABLE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationDisable  } } |
| LIST\_NOTIF\_DELETE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationDelete  } } |
| LIST\_NOTIF\_INSTALL\_ENABLE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationInstall,  notificationEnable  } } |
| LIST\_NOTIF\_DISABLE\_DELETE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationDisable,  notificationDelete  } } |
| LIST\_NOTIF\_DISABLE\_ENABLE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationDisable,  notificationEnable  } } |
| LIST\_NOTIF\_INSTALL\_ENABLE\_DISABLE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable  } } |
| LIST\_NOTIF\_ENABLE\_DISABLE\_DELETE | request ListNotificationRequest ::= {  profileManagementOperation {  notificationEnable,  notificationDisable,  notificationDelete  } } |
| METADATA\_EN\_DI\_DE\_NOTIFS | metadataReq StoreMetadataRequest ::= {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  notificationConfigurationInfo {  { profileManagementOperation {  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  },  { profileManagementOperation {  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS2  }  }  } |
| PREP\_DOWNLOAD\_INVALID\_CC | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag TRUE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  smdpCertificate #CERT\_S\_SM\_DPpb\_ECDSA  } |
| RETRIEVE\_NOTIF\_ALL | request RetrieveNotificationsListRequest ::= {  searchCriteria profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  } } |
| RETRIEVE\_NOTIF\_OMITTED | request RetrieveNotificationsListRequest ::= {  } |
| RETRIEVE\_NOTIF\_NONE | request RetrieveNotificationsListRequest ::= {  searchCriteria profileManagementOperation {} } |
| RETRIEVE\_NOTIF\_INSTALL | request RetrieveNotificationsListRequest::= {  searchCriteria profileManagementOperation {  notificationInstall  } } |
| RETRIEVE\_NOTIF\_ENABLE | request RetrieveNotificationsListRequest::= {  searchCriteria profileManagementOperation {  notificationEnable  } } |
| RETRIEVE\_NOTIF\_DISABLE | request RetrieveNotificationsListRequest::= {  searchCriteria profileManagementOperation {  notificationDisable  } } |
| RETRIEVE\_NOTIF\_DELETE | request RetrieveNotificationsListRequest::= {  searchCriteria profileManagementOperation {  notificationDelete  } } |
| RETRIEVE\_NOTIF\_INSTALL\_ENABLE | request RetrieveNotificationsListRequest ::= {  searchCriteria profileManagementOperation {  notificationInstall,  notificationEnable  } } |
| RETRIEVE\_NOTIF\_DISABLE\_DELETE | request RetrieveNotificationsListRequest ::= {  searchCriteria profileManagementOperation {  notificationDisable,  notificationDelete  } } |
| RETRIEVE\_NOTIF\_DISABLE\_ENABLE | request RetrieveNotificationsListRequest ::= {  searchCriteria profileManagementOperation {  notificationDisable,  notificationEnable  } } |
| RETRIEVE\_NOTIF\_INSTALL\_ENABLE\_DISABLE | request RetrieveNotificationsListRequest ::= {  searchCriteria profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable  } } |
| PREP\_DOWN\_INV\_CURVE | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <RANDOM\_SM\_DP+\_SIGN>,  smdpCertificate #CERT\_S\_SM\_DPpb\_INV\_CURVE  } |
| PREP\_DOWNLOAD\_CERT\_SMDP2 | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  smdpCertificate #CERT\_S\_SM\_DP2pb\_ECDSA  } |
| PREP\_DOWNLOAD\_INV\_CERT | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  smdpCertificate #CERT\_S\_SM\_DPpb\_INV\_SIGN  } |
| PREP\_DOWNLOAD\_INV\_OID | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  smdpCertificate #CERT\_S\_SM\_DPauth\_ECDSA  } |
| PREP\_DOWNLOAD\_INV\_SIGN | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  smdpCertificate #CERT\_S\_SM\_DPpb\_ECDSA  }  NOTE: The <S\_SM\_DP+\_SIGNATURE2> SHALL NOT be computed using the #SK\_S\_SM\_DPpb\_ECDSA but SHALL have the same length as for a valid signature. |
| PREP\_DOWNLOAD\_INV\_TRANS\_ID | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <INVALID\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  smdpCertificate #CERT\_S\_SM\_DPpb\_ECDSA  } |
| PREP\_DOWNLOAD\_NO\_AUTH | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <RANDOM\_SM\_DP+\_SIGN>,  smdpCertificate #CERT\_S\_SM\_DPpb\_ECDSA  } |
| PREP\_DOWNLOAD\_NO\_CC | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag FALSE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  smdpCertificate #CERT\_S\_SM\_DPpb\_ECDSA  } |
| PREP\_DOWNLOAD\_RETRY\_CC | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag TRUE,  bppEuiccOtpk <OTPK\_EUICC\_ECKA>  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  hashCc <S\_HASHED\_CC>,  smdpCertificate #CERT\_S\_SM\_DPpb\_ECDSA  } |
| PREP\_DOWNLOAD\_WITH\_CC | req PrepareDownloadRequest ::= {  smdpSigned2 {  transactionId <S\_TRANSACTION\_ID>,  ccRequiredFlag TRUE  },  smdpSignature2 <S\_SM\_DP+\_SIGNATURE2>,  hashCc <S\_HASHED\_CC>,  smdpCertificate #CERT\_S\_SM\_DPpb\_ECDSA  } |
| SET\_EUICC\_CONFIGURED\_ADDRESS\_1 | request SetDefaultDpAddressRequest::={  defaultDpAddress #TEST\_DP\_ADDRESS1  } |
| SET\_EUICC\_CONFIGURED\_ADDRESS\_2 | request SetDefaultDpAddressRequest::={  defaultDpAddress #TEST\_DP\_ADDRESS2  } |
| SET\_EUICC\_CONFIGURED\_ADDRESS\_EMPTY | request SetDefaultDpAddressRequest::={  defaultDpAddress ""  } |
| SET\_NICKNAME\_EMPTY\_OP\_PROF1 | setNicknameReq SetNicknameRequest ::= {  iccid #ICCID\_OP\_PROF1,  profileNickname ""  } |
| SET\_NICKNAME\_ICCID\_UNKNOWN | setNicknameReq SetNicknameRequest ::= {  iccid #ICCID\_UNKNOWN,  profileNickname #NICKNAME2  } |
| SET\_NICKNAME\_OP\_PROF1 | setNicknameReq SetNicknameRequest ::= {  iccid #ICCID\_OP\_PROF1,  profileNickname #NICKNAME2  } |

D.3.2 ES10x Responses

|  |  |
| --- | --- |
| Name | Content |
| NOTIF\_METADATA\_DELETE1  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_DE1>,  profileManagementOperation { notificationDelete },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA2\_DELETE1  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO2\_DE1>,  profileManagementOperation { notificationDelete },  notificationAddress #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA\_DISABLE1  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_DI1>,  profileManagementOperation { notificationDisable },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA2\_DISABLE1  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO2\_DI1>,  profileManagementOperation { notificationDisable },  notificationAddress #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA\_ENABLE1  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_EN1>,  profileManagementOperation { notificationEnable },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA2\_ENABLE1  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO2\_EN1>,  profileManagementOperation { notificationEnable },  notificationAddress #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA\_ENABLE2  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_EN2>,  profileManagementOperation { notificationEnable },  notificationAddress #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF2  } |
| NOTIF\_METADATA\_INSTALL1  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_IN1>,  profileManagementOperation { notificationInstall },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA\_INSTALL1\_PIR  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_IN1\_PIR>,  profileManagementOperation {   notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  } |
| NOTIF\_METADATA\_INSTALL2  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_IN2>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF2  } |
| NOTIF\_METADATA\_INSTALL2\_PIR  (NotificationMetadata) | {  seqNumber <NOTIF\_SEQ\_NO\_IN2\_PIR>,  profileManagementOperation {   notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS2,  iccid #ICCID\_OP\_PROF2  } |
| PPR1\_WITH\_OWNER\_GID  (ProfilePolicyAuthorisationRule) | {  pprIds { ppr1 },  allowedOperators {  { mccMnc #MCC\_MNC2,  gid1 #GID1,  gid2 #GID2  }  },  pprFlags {consentRequired}  } |
| PPR1\_WITHOUT\_GID  (ProfilePolicyAuthorisationRule) | {  pprIds { ppr1 },  allowedOperators {  { mccMnc #MCC\_MNC4 }  },  pprFlags {consentRequired}  } |
| PPR2\_WITHOUT\_CONSENT  (ProfilePolicyAuthorisationRule) | {  pprIds { ppr2 },  allowedOperators {  { mccMnc '92EEEE'H, gid1 ''H, gid2 ''H}  },  pprFlags { }  } |
| PPRS\_ALLOWED  (ProfilePolicyAuthorisationRule) | {  pprIds { ppr1, ppr2 },  allowedOperators {  { mccMnc 'EEEEEE'H, gid1 ''H, gid2 ''H}  },  pprFlags {consentRequired}  } |
| PROFILE\_INFO1  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF1,  isdpAid <ISD\_P\_AID1>,  profileState enabled,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational  } |
| PROFILE\_INFO1\_DISABLED  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF1,  isdpAid <ISD\_P\_AID1>,  profileState disabled,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational  } |
| PROFILE\_INFO2  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF2,  isdpAid <ISD\_P\_AID2>,  profileState disabled,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF2,  profileClass operational  } |
| PROFILE\_INFO2\_ENABLED  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF2,  isdpAid <ISD\_P\_AID2>,  profileState enabled,  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF2,  profileClass operational  } |
| PROFILE\_INFO3  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF3,  isdpAid <ISD\_P\_AID3>,  profileState disabled,  profileNickname #NICKNAME3,  serviceProviderName #SP\_NAME3,  profileName #NAME\_OP\_PROF3,  iconType png,  icon #ICON\_OP\_PROF3,  profileClass operational  } |
| PROFILE\_INFO4  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF4,  isdpAid <ISD\_P\_AID4>,  profileState disabled,  serviceProviderName #SP\_NAME4,  profileName #NAME\_OP\_PROF4,  iconType png,  icon #ICON\_OP\_PROF4,  profileClass operational  } |
| PROFILE\_INFO4\_ENABLED  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF4,  isdpAid <ISD\_P\_AID4>,  profileState enabled,  serviceProviderName #SP\_NAME4,  profileName #NAME\_OP\_PROF4,  iconType png,  icon #ICON\_OP\_PROF4,  profileClass operational  } |
| PROFILES\_INFO\_ICCID\_TAGLIST1  (ProfileInfo) | {profileState enabled} |
| PROFILES\_INFO\_ICCID\_TAGLIST2  (ProfileInfo) | {iconType png} |
| PROFILES\_INFO\_ICCID\_TAGLIST3  (ProfileInfo) | {profileClass operational } |
| PROFILES\_INFO\_ICCID\_TAGLIST4  (ProfileInfo) | notificationConfigurationInfo from #METADATA\_OP\_PROF1 |
| PROFILES\_INFO\_ICCID\_TAGLIST5  (ProfileInfo) | profilePolicyRules from #METADATA\_OP\_PROF3 |
| PROFILES\_INFO\_TAGLIST\_ICCID  (ProfileInfo) | {iccid #ICCID\_OP\_PROF1},  {iccid #ICCID\_OP\_PROF2},  {iccid #ICCID\_OP\_PROF3} |
| PROFILES\_INFO\_TAGLIST\_ICON  (ProfileInfo) | {icon #ICON\_OP\_PROF1},  {icon #ICON\_OP\_PROF2},  {icon #ICON\_OP\_PROF3} |
| PROFILES\_INFO\_TAGLIST\_ISDPAID  (ProfileInfo) | {isdpAid <ISD\_P\_AID1>},  {isdpAid <ISD\_P\_AID2>},  {isdpAid <ISD\_P\_AID3>} |
| PROFILES\_INFO\_TAGLIST\_PROFILE\_NAME  (ProfileInfo) | {profileName #NAME\_OP\_PROF1},  {profileName #NAME\_OP\_PROF2},  {profileName #NAME\_OP\_PROF3} |
| PROFILES\_INFO\_TAGLIST\_PROFILE\_NICKNAME  (ProfileInfo) | {profileNickname #NICKNAME3} |
| PROFILES\_INFO\_TAGLIST\_PROFILE\_OWNER  (ProfileInfo) | {profileOwner #OWNER\_OP\_PROF1},  {profileOwner #OWNER\_OP\_PROF2},  {profileOwner #OWNER\_OP\_PROF2} |
| PROFILES\_INFO\_TAGLIST\_SMDP\_PROP\_DATA  (ProfileInfo) | {dpProprietaryData #SMDP\_PROP\_DATA1} |
| PROFILES\_INFO\_TAGLIST\_SP\_NAME  (ProfileInfo) | {serviceProviderName #SP\_NAME1},  {serviceProviderName #SP\_NAME2},  {serviceProviderName #SP\_NAME3} |
| PROFILES\_INFO\_TAGLIST1  (ProfileInfo) | {  iccid #ICCID\_OP\_PROF1,   profileState enabled  },  {  iccid #ICCID\_OP\_PROF2,   profileState disabled  },  {  iccid #ICCID\_OP\_PROF3,   profileState disabled  } |
| PROFILES\_INFO\_TAGLIST2  (ProfileInfo) | {  profileState enabled  },  {  profileState disabled  },  {  profileState disabled,  profileNickname #NICKNAME3 } |
| PROFILES\_INFO\_TAGLIST3  (ProfileInfo) | {  iconType png,  icon #ICON\_OP\_PROF1  },  {  iconType png,  icon #ICON\_OP\_PROF2  },  {  iconType png,  icon #ICON\_OP\_PROF3  } |
| PROFILES\_INFO\_TAGLIST4  (ProfileInfo) | {  profileState enabled,  icon #ICON\_OP\_PROF1  },  {  profileState disabled,  icon #ICON\_OP\_PROF2  },  {  profileState disabled,  icon #ICON\_OP\_PROF3  } |
| R\_AUTH\_SMDP\_MATCH\_ID | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present and has a valid TLV asn.1 structure  ctxParams1 #CTX\_PARAMS1\_MATCH\_ID  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_AUTH\_SMDP\_IMEI | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_DP\_ADDRESS1,  serverChallenge <S\_SMDP\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present and has a valid TLV asn.1 structure  ctxParams1 #CTX\_PARAMS1\_IMEI  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_AUTH\_SERVER\_INV\_CERT | resp AuthenticateServerResponse ::= authenticateResponseError : {  transactionId <S\_TRANSACTION\_ID>,  authenticateErrorCode invalidCertificate  } |
| R\_AUTH\_SERVER\_INV\_SIGN | resp AuthenticateServerResponse ::= authenticateResponseError : {  transactionId <S\_TRANSACTION\_ID>,  authenticateErrorCode invalidSignature  } |
| R\_AUTH\_SERVER\_INV\_CURV | resp AuthenticateServerResponse ::= authenticateResponseError : {  transactionId <S\_TRANSACTION\_ID>,  authenticateErrorCode unsupportedCurve  } |
| R\_AUTH\_SERVER\_INV\_CHALLENGE | resp AuthenticateServerResponse ::= authenticateResponseError : {  transactionId <S\_TRANSACTION\_ID>,  authenticateErrorCode euiccChallengeMismatch  } |
| R\_AUTH\_SERVER\_INV\_CI | resp AuthenticateServerResponse ::= authenticateResponseError : {  transactionId <S\_TRANSACTION\_ID>,  authenticateErrorCode ciPKUnknown  } |
| R\_AUTH\_SERVER\_INV\_OID | resp AuthenticateServerResponse ::= authenticateResponseError : {  transactionId <S\_TRANSACTION\_ID>,  authenticateErrorCode invalidOid  } |
| R\_AUTH\_SERVER\_NO\_SESSION | resp AuthenticateServerResponse ::= authenticateResponseError : {  transactionId <S\_TRANSACTION\_ID>,  authenticateErrorCode noSessionContext  } |
| R\_AUTH\_SMDS\_IMEI | resp AuthenticateServerResponse ::= authenticateResponseOk : {  euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present and has a valid TLV asn.1 structure  ctxParams1 #CTX\_PARAMS1\_EVENT\_ID\_IMEI  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_AUTHENTICATE\_SMDP | resp AuthenticateServerResponse ::= authenticateResponseOk: {  euiccSigned1 #EUICC\_SIGNED1,  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_AUTHENTICATE\_SMDS | resp AuthenticateServerResponse ::= authenticateResponseOk: {  euiccSigned1 {  transactionId <S\_TRANSACTION\_ID>,  serverAddress #TEST\_ROOT\_DS\_ADDRESS,  serverChallenge <S\_SMDS\_CHALLENGE>,  euiccInfo2 #R\_EUICC\_INFO2, -- check only that the field is present and has a valid TLV asn.1 structure  ctxParams1 #CTX\_PARAMS1\_EVENT\_ID  },  euiccSignature1 <EUICC\_SIGNATURE1>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  } |
| R\_CANCEL\_SESSION\_INV\_TRANS\_ID | resp CancelSessionResponse ::= cancelSessionResponseError : invalidTransactionId |
| R\_CANCEL\_SESSION\_METADATA | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason metadataMismatch  },  euiccCancelSessionSignature <EUICC\_CS\_SIGNATURE>  } |
| R\_CANCEL\_SESSION\_REJ | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason endUserRejection  },  euiccCancelSessionSignature <EUICC\_CS\_SIGNATURE>  } |
| R\_CANCEL\_SESSION\_POSTPONED | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason postponed  },  euiccCancelSessionSignature <EUICC\_CS\_SIGNATURE>  } |
| R\_CANCEL\_SESSION\_TIMEOUT | resp CancelSessionResponse ::= cancelSessionResponseOk {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason timeout  },  euiccCancelSessionSignature <EUICC\_CS\_SIGNATURE>  } |
| R\_CANCEL\_SESSION\_PPR | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason pprNotAllowed  },  euiccCancelSessionSignature <EUICC\_CS\_SIGNATURE>  } |
| R\_CANCEL\_SESSION\_LOAD\_BPP | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason loadBppExecutionError  },  euiccCancelSessionSignature <EUICC\_CS\_SIGNATURE>  } |
| R\_CANCEL\_SESSION\_UNDEF | resp CancelSessionResponse ::= cancelSessionResponseOk : {  euiccCancelSessionSigned {  transactionId <S\_TRANSACTION\_ID>,  smdpOid #S\_SM\_DP+\_OID,  reason undefinedReason  },  euiccCancelSessionSignature <EUICC\_CS\_SIGNATURE>  } |
| R\_CHALLENGE | response GetEuiccChallengeResponse ::=  {  euiccChallenge <EUICC\_CHALLENGE>  } |
| R\_CONF\_OP\_PROF1 | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  isdpAid <ISD\_P\_AID>,  dpProprietaryData {  dpOid #S\_SM\_DP+\_OID,  additionalSmdpData  #ADDITIONAL\_SMDP\_DATA\_MAX\_LENGTH  }  }  }  -- NOTE: Instead of  DpProprietaryData ::= SEQUENCE {  dpOid OBJECT IDENTIFIER  -- additional data objects defined by the  -- SM-DP+ MAY follow  }  -- the following structure is used to test the  -- DpProprietaryData size:  DpProprietaryData ::= SEQUENCE {  dpOid OBJECT IDENTIFIER,  additionalSmdpData OCTET STRING OPTIONAL  } |
| R\_DEFAULT\_RAT | response GetRatResponse ::= {  rat {  #PPRS\_ALLOWED  }  } |
| R\_DELETE\_PROFILE\_DISALLOWEDBYPOLICY | respDelProf DeleteProfileResponse ::= {  deleteResult disallowedByPolicy  } |
| R\_DELETE\_PROFILE\_NOTDISABLESTATE | respDelProf DeleteProfileResponse ::= {  deleteResult profileNotInDisabledState  } |
| R\_DELETE\_PROFILE\_OK | respDelProf DeleteProfileResponse ::= {  deleteResult ok  } |
| R\_DELETE\_PROFILE\_ICCID\_ISDP\_NOTFOUND | resp DeleteProfileResponse ::= {  deleteResult iccidOrAidNotFound  } |
| R\_DISABLE\_PROFILE\_DISALLOWEDbyPOLICY | resp DisableProfileResponse ::= {  disableResult disallowedByPolicy  } |
| R\_DISABLE\_PROFILE\_ICCID\_ISDP\_NOTFOUND | resp DisableProfileResponse ::= {  disableResult iccidOrAidNotFound  } |
| R\_DISABLE\_PROFILE\_NOT\_ENABLE\_STATE | resp DisableProfileResponse ::= {  disableResult profileNotInEnabledState  } |
| R\_DISABLE\_PROFILE\_OK | resp DisableProfileResponse ::= {  disableResult ok  } |
| R\_ENABLE\_PROFILE\_ICCID\_ISDP\_NOTFOUND | respEnaPro EnableProfileResponse ::= {  enableResult iccidOrAidNotFound  } |
| R\_ENABLE\_PROFILE\_NOT\_DISABLE\_STATE | respEnaPro EnableProfileResponse ::= {  enableResult profileNotInDisabledState  } |
| R\_ENABLE\_PROFILE\_DISALLOWEDbyPOLICY | respEnaPro EnableProfileResponse ::= {  enableResult disallowedByPolicy  } |
| R\_ENABLE\_PROFILE\_OK | resp EnableProfileResponse ::= {  enableResult ok  } |
| R\_ES10a\_GECA\_DS | response EuiccConfiguredAddressesResponse ::= {  *-- defaultDpAddress SHALL not be present*  rootDsAddress #TEST\_ROOT\_DS\_ADDRESS  } |
| R\_ES10a\_GECA\_DS\_DP\_1 | response EuiccConfiguredAddressesResponse ::= {  defaultDpAddress #TEST\_DP\_ADDRESS1,  rootDsAddress #TEST\_ROOT\_DS\_ADDRESS  } |
| R\_ES10a\_GECA\_DS\_DP\_2 | response EuiccConfiguredAddressesResponse ::= {  defaultDpAddress #TEST\_DP\_ADDRESS2,  rootDsAddress #TEST\_ROOT\_DS\_ADDRESS  } |
| R\_ES10a\_SD\_DP\_A\_OK | response SetDefaultDpAddressResponse::= {  setDefaultDpAddressResult ok  } |
| R\_EUICC\_INFO1 | response EUICCInfo1::=  {  svn <ANY\_SVN>  euiccCiPKIdListForVerification  <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION>,  euiccCiPKIdListForSigning  <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>  } |
| R\_EUICC\_INFO2 | response EUICCInfo2::=  {  profileVersion <ANY\_PROFILE\_VERSION>,  svn <ANY\_SVN>,  euiccFirmwareVer #IUT\_EUICC\_FIRMWARE\_VER,  extCardResource <EXT\_CARD\_RESOURCE>,  uiccCapability #IUT\_UICC\_CAPABILITY,  ts102241Version #IUT\_TS102241\_VERSION,  globalplatformVersion   #IUT\_GLOBALPLATFORM\_VERSION,  rspCapability <EUICC\_RSP\_CAPABILITY>,  euiccCiPKIdListForVerification  <EUICC\_CI\_PK\_ID\_LIST\_FOR\_VERIFICATION>,  euiccCiPKIdListForSigning  <EUICC\_CI\_PK\_ID\_LIST\_FOR\_SIGNING>,  euiccCategory #IUT\_EUICC\_CATEGORY, *-- OPTIONAL*  forbiddenProfilePolicyRules <PPR\_IDS>, *-- OPTIONAL*  ppVersion #IUT\_PP\_VERSION,  sasAcreditationNumber #IUT\_SAS\_ACCREDITATION\_NUMBER,  certificationDataObject {  platformLabel #IUT\_PLATFORM\_LABEL,  discoveryBaseURL #IUT\_DLOA\_URL  },-- OPTIONAL  treProperties <TRE\_PROPERTIES>, -- OPTIONAL  treProductReference <TRE\_REFERENCE>, -- OPTIONAL  additionalEuiccProfilePackageVersions <ANY\_ADD\_PP\_VERSIONS> -- OPTIONAL  -- <ANY\_ADD\_PP\_VERSIONS> shall be considered as “absent” if the additionalEuiccProfilePackageVersions field is not present  } |
| R\_EUICC\_MEMORY\_RESET\_OK | resp EuiccMemoryResetResponse ::= {  resetResult ok  } |
| R\_GET\_UPDATE\_N1 | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profilePolicyRules { ppr2 }  }  } |
| R\_GET\_UPDATE\_N2 | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType jpg,  icon #ICON\_JPG,  profilePolicyRules { ppr1 }  }  } |
| R\_GET\_UPDATE\_N3 | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF1  *-- profilePolicyRules SHALL not be present*  }  } |
| R\_GET\_UPDATE\_N4 | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  *-- serviceProviderName SHALL not be present*  *-- profileName SHALL not be present*  iconType png,  icon #ICON\_OP\_PROF1  *-- profilePolicyRules SHALL not be present*  }  } |
| R\_GET\_UPDATE\_N6 | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType png,  icon #ICON\_OP\_PROF1  *-- profilePolicyRules SHALL not be present*  }  } |
| R\_LIST\_NOTIF\_DI1\_EN2 | response ListNotificationResponse ::= notificationMetadataList : {  #NOTIF\_METADATA\_DISABLE1,  #NOTIF\_METADATA\_ENABLE2  } |
| R\_METADATA\_UNCHANGED | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profilePolicyRules {ppr1,ppr2}  }  } |
| R\_PIR\_DATA\_MISMATCH | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  ...  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId loadProfileElements,  errorReason installFailedDueToDataMismatch,  ...  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_OK\_PROF9 | response ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF9  },  smdpOid #S\_SM\_DP+\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_PPR\_NOT\_ALLOWED | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  ...  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason pprNotAllowed  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_GET\_METADATA\_OP\_PROF1 | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1}  }  } |
| R\_GET\_METADATA\_OP\_PROF1\_SERVICE\_SPECIFIC | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  iccid #ICCID\_OP\_PROF1,  serviceProviderName #SP\_NAME1,  profileName #NAME\_OP\_PROF1,  iconType png,  icon #ICON\_OP\_PROF1,  profileClass operational,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall,  notificationEnable,  notificationDisable,  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  }  },  profileOwner {  mccMnc #MCC\_MNC1  },  profilePolicyRules {ppr1},  serviceSpecificDataStoredInEuicc #VENDOR\_SPECIFIC\_EXTENSION1  }  } |
| R\_GET\_PROF\_NOTIF\_CONF | resp ProfileInfoListResponse ::=  profileInfoListOk :{  {  iccid #ICCID\_OP\_PROF1,  notificationConfigurationInfo {  { profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS3  },  { profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS2  },  { profileManagementOperation {  notificationEnable  },  notificationAddress #TEST\_DP\_ADDRESS2  },  { profileManagementOperation {  notificationEnable  },  notificationAddress #TEST\_DP\_ADDRESS3  },  { profileManagementOperation {  notificationDisable  },  notificationAddress #TEST\_DP\_ADDRESS3  },  { profileManagementOperation {  notificationDisable  },  notificationAddress #TEST\_DP\_ADDRESS4  },  { profileManagementOperation {  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS1  },  { profileManagementOperation {  notificationDelete  },  notificationAddress #TEST\_DP\_ADDRESS3  }  }  }  } |
| R\_ISDR\_SELECTION | resp ISDRProprietaryApplicationTemplate::= {  svn <ANY\_SVN>  } |
| R\_LIST\_NOTIF\_DE1 | response ListNotificationResponse ::= notificationMetadataList : {  #NOTIF\_METADATA\_DELETE1  } |
| R\_LIST\_NOTIF\_DE1\_DE1 | response ListNotificationResponse ::= notificationMetadataList : {  #NOTIF\_METADATA\_DELETE1,  #NOTIF\_METADATA2\_DELETE1  } |
| R\_LIST\_NOTIF\_DI1 | response ListNotificationResponse ::= notificationMetadataList : {  #NOTIF\_METADATA\_DISABLE1  } |
| R\_LIST\_NOTIF\_DI1\_DE1 | response ListNotificationResponse ::= notificationMetadataList : {  #NOTIF\_METADATA\_DISABLE1,  #NOTIF\_METADATA\_DELETE1  } |
| R\_LIST\_NOTIF\_DI1\_DI1 | response ListNotificationResponse ::= notificationMetadataList : {  #NOTIF\_METADATA\_DISABLE1,  #NOTIF\_METADATA2\_DISABLE1  } |
| R\_LIST\_NOTIF\_EN1 | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_ENABLE1  } |
| R\_LIST\_NOTIF\_EN1\_EN1 | response ListNotificationResponse ::= notificationMetadataList : {  #NOTIF\_METADATA\_ENABLE1,  #NOTIF\_METADATA2\_ENABLE1  } |
| R\_LIST\_NOTIF\_IN1 | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_INSTALL1  } |
| R\_LIST\_NOTIF\_IN1\_IN1\_PIR | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_INSTALL1,  #NOTIF\_METADATA\_INSTALL1\_PIR  } |
| R\_LIST\_NOTIF\_IN1\_PIR | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_INSTALL1\_PIR  } |
| R\_LIST\_NOTIF\_IN1\_EN1 | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_INSTALL1,  #NOTIF\_METADATA\_ENABLE1  } |
| R\_LIST\_NOTIF\_IN1\_PIR\_EN1 | response ListNotificationResponse ::= notificationMetadataList: {   #NOTIF\_METADATA\_INSTALL1\_PIR,  #NOTIF\_METADATA\_ENABLE1  } |
| R\_LIST\_NOTIF\_IN2\_PIR | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_INSTALL2\_PIR  } |
| R\_LIST\_NOTIF\_IN2\_PIR\_IN2 | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_INSTALL2\_PIR,  #NOTIF\_METADATA\_INSTALL2  } |
| R\_LIST\_NOTIF\_IN1\_PIR\_IN2\_PIR | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_INSTALL1\_PIR,  #NOTIF\_METADATA\_INSTALL2\_PIR  } |
| R\_LIST\_NOTIF\_NONE | response ListNotificationResponse ::=  notificationMetadataList: {} |
| R\_LIST\_NOTIF\_UNDEFINED\_ERROR | response ListNotificationResponse ::= listNotificationsResultError : undefinedError |
| R\_LIST\_NOTIF\_EN1\_IN2\_PIR | response ListNotificationResponse ::= notificationMetadataList: {  #NOTIF\_METADATA\_ENABLE1,  #NOTIF\_METADATA\_INSTALL2\_PIR  } |
| R\_PIR\_ICCID\_ALREADY\_EXIST | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason  installFailedDueToIccidAlreadyExistsOnEuicc  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_INVALID\_CRT | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId initialiseSecureChannel,  errorReason unsupportedCrtValues  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_INVALID\_DATA | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId configureISDP,  errorReason incorrectInputValues  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_INVALID\_OP\_ID | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId initialiseSecureChannel,  errorReason unsupportedRemoteOperationType  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_INVALID\_SIGN | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId initialiseSecureChannel,  errorReason invalidSignature }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_INVALID\_TRANS\_ID | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <INVALID\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId initialiseSecureChannel,  errorReason invalidTransactionId  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_METADATA\_INVALID | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  …  },  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId storeMetadata,  errorReason  scp03tStructureError  OR  incorrectInputValues  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_OK | response ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata {  seqNumber <SEQ\_NUMBER>,  profileManagementOperation {  notificationInstall  },  notificationAddress #TEST\_DP\_ADDRESS1,  iccid #ICCID\_OP\_PROF1  },  smdpOid #S\_SM\_DP+\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_PPK\_INV | resp ProfileInstallationResult ::= {  profileInstallationResultData {  ...  finalResult errorResult : {  bppCommandId replaceSessionKeys,  errorReason  incorrectInputValues  OR  scp03tStructureError  OR  scp03tSecurityError  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PIR\_SECU\_INVALID | resp ProfileInstallationResult ::= {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  …  smdpOid #S\_SM\_DP+\_OID,  finalResult errorResult : {  bppCommandId loadProfileElements,  errorReason incorrectInputValues  OR  scp03tStructureError  OR  scp03tSecurityError  …  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  } |
| R\_PREP\_DOWN\_INV\_CURVE | resp PrepareDownloadResponse ::= downloadResponseError : {  transactionId <S\_TRANSACTION\_ID>,  downloadErrorCode unsupportedCurve  } |
| R\_PREP\_DOWN\_INV\_TRANS\_ID | resp PrepareDownloadResponse ::= downloadResponseError : {  transactionId <INVALID\_TRANSACTION\_ID>,  downloadErrorCode invalidTransactionId  } |
| R\_PREP\_DOWN\_NO\_SESSION | resp PrepareDownloadResponse ::= downloadResponseError : {  transactionId <S\_TRANSACTION\_ID>,  downloadErrorCode noSessionContext  } |
| R\_PREP\_DOWNLOAD\_INV\_CERT | resp PrepareDownloadResponse ::= downloadResponseError : {  transactionId <S\_TRANSACTION\_ID>,  downloadErrorCode invalidCertificate  } |
| R\_PREP\_DOWNLOAD\_INV\_SIGN | resp PrepareDownloadResponse ::= downloadResponseError : {  transactionId <S\_TRANSACTION\_ID>,  downloadErrorCode invalidSignature  } |
| R\_PREP\_DOWNLOAD\_NO\_CC | resp PrepareDownloadResponse ::= downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <OTPK\_EUICC\_ECKA>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } |
| R\_PREP\_DOWNLOAD\_WITH\_CC | resp PrepareDownloadResponse ::= downloadResponseOk : {  euiccSigned2 {  transactionId <S\_TRANSACTION\_ID>,  euiccOtpk <OTPK\_EUICC\_ECKA>,  hashCc <S\_HASHED\_CC>  },  euiccSignature2 <EUICC\_SIGNATURE2>  } |
| R\_RAT\_WITH\_OTHER\_RULES | response GetRatResponse ::= {  rat {  #PPR1\_WITH\_OWNER\_GID,  #PPR1\_WITHOUT\_GID,  #PPR2\_WITHOUT\_CONSENT,  #PPRS\_ALLOWED  }  } |
| R\_REMOVE\_NOTIF\_NOTHING\_TO\_DELETE | response NotificationSentResponse ::= {  deleteNotificationStatus nothingToDelete  } |
| R\_REMOVE\_NOTIF\_OK | response NotificationSentResponse ::= {  deleteNotificationStatus ok  } |
| R\_RETRIEVE\_NOTIF\_IN1\_IN1\_PIR | resp RetrieveNotificationsListResponse ::=  notificationList : {  profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata #NOTIF\_METADATA\_INSTALL1\_PIR,  smdpOid #S\_SM\_DP+\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  },  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_INSTALL1,   euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_IN1\_PIR | resp RetrieveNotificationsListResponse ::=  notificationList : {  profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata #NOTIF\_METADATA\_INSTALL1\_PIR,  smdpOid #S\_SM\_DP+\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  }  } |
| R\_RETRIEVE\_NOTIF\_IN1 | resp RetrieveNotificationsListResponse ::=  notificationList : {  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_INSTALL1,  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_EN1 | resp RetrieveNotificationsListResponse ::=  notificationList : {  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_ENABLE1,  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_IN2\_PIR | resp RetrieveNotificationsListResponse ::=  notificationList : {  profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata #NOTIF\_METADATA\_INSTALL2\_PIR,  smdpOid #S\_SM\_DP+\_OID2,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  }  } |
| R\_RETRIEVE\_NOTIF\_DI1 | resp RetrieveNotificationsListResponse ::=  notificationList : {  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_DISABLE1,  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_DE1 | resp RetrieveNotificationsListResponse ::=  notificationList : {  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_DELETE1,  euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_NONE | resp RetrieveNotificationsListResponse ::=  notificationList : {} |
| R\_RETRIEVE\_NOTIF\_IN1\_PIR\_EN1 | resp RetrieveNotificationsListResponse ::=  notificationList : {  profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata #NOTIF\_METADATA\_INSTALL1\_PIR,  smdpOid #S\_SM\_DP+\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  },  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_ENABLE1,   euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_IN1\_PIR\_IN2\_PIR | resp RetrieveNotificationsListResponse ::=  notificationList : {  profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata #NOTIF\_METADATA\_INSTALL1\_PIR,  smdpOid #S\_SM\_DP+\_OID,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  },  profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata #NOTIF\_METADATA\_INSTALL2\_PIR,  smdpOid #S\_SM\_DP+\_OID2,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  }  } |
| R\_RETRIEVE\_NOTIF\_DI1\_DE1 | resp RetrieveNotificationsListResponse ::=  notificationList : {  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_DISABLE1,   euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA   },  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_DELETE1,   euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_IN1\_EN1 | resp RetrieveNotificationsListResponse ::=  notificationList : {  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_INSTALL1,   euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA   },  otherSignedNotification : {  tbsOtherNotification #NOTIF\_METADATA\_ENABLE1,   euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| R\_RETRIEVE\_NOTIF\_EN1\_IN2\_PIR | resp RetrieveNotificationsListResponse ::=  notificationList : {  profileInstallationResult : {  profileInstallationResultData {  transactionId <S\_TRANSACTION\_ID>,  notificationMetadata #NOTIF\_METADATA\_INSTALL2\_PIR,  smdpOid #S\_SM\_DP+\_OID2,  finalResult successResult : {  aid <ISD\_P\_AID>,  simaResponse #SIMA\_RESULT\_OK  }  },  euiccSignPIR <EUICC\_SIGN\_PIR>  },  otherSignedNotification : {  tbsOtherNotification#NOTIF\_METADATA\_ENABLE1,   euiccNotificationSignature <TBS\_EUICC\_NOTIF\_SIG>,  euiccCertificate #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA  }  } |
| SMDP\_PROP\_DATA1  (DpProprietaryData) | {  dpOid #S\_SM\_DP+\_OID  } |

D.4 APDU

D.4.1 APDU Commands

|  |  |
| --- | --- |
| Name | Content |
| DELETE\_SSD | - CLA = 80, INS = E4, P1 = 00, P2 = 80, LC = <L>  - Data = 4F <L> #SSD\_AID  - LE = 00 |
| GET\_RESPONSE | - CLA = 0x (x = <CHANNEL\_NUMBER>), INS = C0,  P1 = 00, P2 = 00, LE = <L> |
| GET\_MNO\_SD | - CLA = 80, INS = F2, P1 = 80, P2 = 02, LC = <L>  - Data = 4F 00  - LE = 00 |
| INSTALL\_PERSO\_RES\_ISDP | - CLA = 80, INS = E6, P1 = 20, P2 = 00, LC = 16  - Data = 00 00 10 A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 0F 00 00 00 00  - LE = 00 |
| MANAGE\_CHANNEL\_OPEN | - CLA = 00, INS = 70, P1 = 00, P2 = 00, LE = 01 |
| READ\_BINARY | - CLA = 00, INS = B0, P1 = 00, P2 = 00, LE = <L> |
| SELECT\_MF | - CLA = 00, INS = A4, P1 = 00, P2 = 04, LC = <L>  - Data = 3F 00  - LE = 00 |
| SELECT\_ICCID | - CLA = 00, INS = A4, P1 = 00, P2 = 0C, LC = 02  - Data = 2F E2 |
| SELECT\_USIM | - CLA = 00, INS = A4, P1 = 04, P2 = 04, LC = <L>  - Data = #USIM\_AID  - LE = 00 |
| TERMINAL\_CAPABILITY\_LPAd | - CLA = 80, INS = AA, P1 = 00, P2 = 00, LC = <L>  - Data = A9 05 81 00 83 01 07 |
| TERMINAL\_PROFILE | - CLA = 80, INS = 10, P1 = 00, P2 = 00, LC = <L>  - Data = FF FF FF FF 7F 9D 00 DF BF 00 00 1F E2 00 00 00 C7 EB 00 00 00 01 68 00 50 00 00 00 00 00 02 00 |
| TERMINAL\_PROFILE\_eUICCProfileStateChanged | - CLA = 80, INS = 10, P1 = 00, P2 = 00, LC = <L>  - Data = FF FF FF FF FF FF 1F FF FF 03 02 FF FF 9F FF EF DF FF 0F FF 0F FF FF 0F FF 03 00 3F 7F FF 03 FF FF 20 |

D.4.2 R-APDU Chaining

During the execution of all sequences related to the eUICC testing (i.e. section 4.2), for commands where the response exceeds 256 bytes, the chaining mechanism defined in ISO/IEC 7816-4 [7], using the 61XX status word and multiple GET RESPONSE commands, SHALL be used.

As an example, the following generic sequence, which describes this mechanism, SHALL apply.

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Direction | Sequence / Description | Result |
| 1 | OCE 🡪 eUICC | Send APDU command on logical channel x | <R\_APDU\_PART1>  SW=0x61XX |
| 2 | OCE 🡪 eUICC | Send [GET\_RESPONSE] on logical channel x with LE='XX' | <R\_APDU\_PART2>  SW=0x61XX |
| 3 | OCE 🡪 eUICC | Send [GET\_RESPONSE] on logical channel x with LE='XX' | <R\_APDU\_PART3>  SW=0x61XX |
| 4 | OCE 🡪 eUICC | Send [GET\_RESPONSE] on logical channel x with LE='XX' | <R\_APDU\_PART4>  SW=0x9000  The complete response is the result of the concatenation of all R-APDUs from <R\_APDU\_PART1> to <R\_APDU\_PART4> |

D.5 ES6 Requests And Responses

D.5.1 ES6 Requests

|  |  |
| --- | --- |
| Name | Content |
| REMOVE\_PPR1 | metadataReq UpdateMetadataRequest ::= {  profilePolicyRules {ppr2}  } |
| UPD\_ICON\_REM\_PPR2 | metadataReq UpdateMetadataRequest ::= {  iconType jpg,  icon #ICON\_JPG,  profilePolicyRules {ppr1}  } |
| UPD\_NAMES\_REM\_PPRS | metadataReq UpdateMetadataRequest ::= {  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  profilePolicyRules {}  } |
| REMOVE\_NAMES\_PPRS | metadataReq UpdateMetadataRequest ::= {  serviceProviderName "",  profileName "",  profilePolicyRules {}  } |
| UPD\_PPR\_CONTROL | metadataReq UpdateMetadataRequest ::= {  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType jpg,  icon #ICON\_JPG,  profilePolicyRules {pprUpdateControl, ppr1}  } |
| UPD\_NO\_METADATA | metadataReq UpdateMetadataRequest ::= { } |
| UPD\_ICON\_NO\_TYPE | metadataReq UpdateMetadataRequest ::= {  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  icon #ICON\_JPG,  profilePolicyRules {}  } |
| UPD\_ICON\_TYPE\_ONLY | metadataReq UpdateMetadataRequest ::= {  serviceProviderName #SP\_NAME2,  profileName #NAME\_OP\_PROF2,  iconType jpg,  profilePolicyRules {}  } |

D.6 ES11 Requests And Responses

D.6.1 ES11 Requests

|  |  |
| --- | --- |
| Name | Content |
| AUTH\_SERVER\_RESP\_MATCHING\_ID\_EMPTY | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_MATCHING\_ID\_EVENT\_ID | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EVENT\_ID  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_MATCHING\_ID\_EVENT\_ID\_R | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EVENT\_ID\_R  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_MATCHING\_ID\_OMITTED | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_OMITTED  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_BC\_cA | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_BC\_cA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_BC\_PLC | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_BC\_PLC } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_CP | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_CP } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_EX\_KU | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_EX\_KU } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_1\_SIG | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_INVALID\_SIG } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_2\_6\_3 | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate   #CERT\_EUM\_ECDSA\_EXPIRED } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_EX\_CP | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_EX\_CP,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_EX\_KU | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_EX\_KU,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_SIG | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_SIG,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_SUB\_ORG | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_SUB\_ORG,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_1\_SUB\_SN | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_INVALID\_SUB\_SN,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_3\_6\_3 | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA\_EXPIRED,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_6\_1\_CHA | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE\_INVALID>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_1\_6\_1\_SIG | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <S\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1\_INVALID>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } |
| AUTH\_SERVER\_RESP\_SMDS\_8\_10\_1\_3\_9 | resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {  transactionId   <INVALID\_TRANSACTION\_ID>,  serverAddress   #IUT\_SM\_DS\_ADDRESS\_ES11,  serverChallenge   <SMDS\_CHALLENGE>,  euiccInfo2 #S\_EUICC\_INFO2,  ctxParams1   #CTX\_PARAMS1\_MATCHING\_ID\_EMPTY  },  euiccSignature1   <EUICC\_SIGNATURE1>,  euiccCertificate   #CERT\_EUICC\_ECDSA,  eumCertificate #CERT\_EUM\_ECDSA } |
| CTX\_PARAMS1\_MATCHING\_ID\_EVENT\_ID (CtxParams1) | ctxParamsForCommonAuthentication : {  matchingId #EVENT\_ID\_1,  deviceInfo #S\_DEVICE\_INFO } |
| CTX\_PARAMS1\_MATCHING\_ID\_EVENT\_ID\_R (CtxParams1) | ctxParamsForCommonAuthentication : {  matchingId <EVENT\_ID\_R>,  deviceInfo #S\_DEVICE\_INFO } |
| CTX\_PARAMS1\_MATCHING\_ID\_OMITTED  (CtxParams1) | ctxParamsForCommonAuthentication : {  deviceInfo #S\_DEVICE\_INFO } |

D.6.2 ES11 Responses

|  |  |
| --- | --- |
| Name | Content |
| AUTH\_CLIENT\_DS\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : #EVENT\_ENTRY  } |
| AUTH\_CLIENT\_DS\_OK1 | {  "header" :{  "functionExecutionStatus":{  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : [ #EVENT\_ENTRY\_1 ]  } |
| AUTH\_CLIENT\_DS\_OK2 | {  "header" :{  "functionExecutionStatus":{  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : [ #EVENT\_ENTRY\_2 ]  } |
| AUTH\_CLIENT\_DS\_OK\_DSADDR1 | {  "header" :{  "functionExecutionStatus":{  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : [ #EVENT\_ENTRY\_DSADDR1 ]  } |
| EVENT\_ENTRY | {  "eventId" : <EVENT\_ID>,  "rspServerAddress" : <RSP\_SERVER\_ADDRESS>  } |
| EVENT\_ENTRY\_1 | {  "eventId" : #EVENT\_ID\_1,  "rspServerAddress" : #TEST\_DP\_ADDRESS1 } |
| EVENT\_ENTRY\_1\_ALT\_DS | {  "eventId" : #EVENT\_ID\_1,  "rspServerAddress" : #TEST\_ALT\_DS\_ADDRESS } |
| EVENT\_ENTRY\_2 | {  "eventId" : #EVENT\_ID\_2,  "rspServerAddress" : #TEST\_DP\_ADDRESS1  } |
| EVENT\_ENTRY\_DSADDR1 | {  "eventId" : #EVENT\_ID\_1,  "rspServerAddress" : #TEST\_DS\_ADDRESS1  } |
| EVENT\_ENTRY\_MULTI | {  "eventId" : #EVENT\_ID\_1,  "rspServerAddress" : #TEST\_DP\_ADDRESS1 },  {  "eventId" : #EVENT\_ID\_2,  "rspServerAddress" : #TEST\_DP\_ADDRESS2 } |
| R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_ALT\_DS\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : [#EVENT\_ENTRY\_1\_ALT\_DS]  } |
| R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_1\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : [#EVENT\_ENTRY\_1]  } |
| R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_EMPTY\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : []  } |
| R\_AUTH\_CLIENT\_DS\_EVENT\_ENTRY\_MULTI\_OK | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  },  "transactionId" : <S\_TRANSACTION\_ID>,  "eventEntries" : [#EVENT\_ENTRY\_MULTI]  } |

D.7 ES12 Requests And Responses

There are no specific ES12 requests or responses defined in the present document.

D.8 ES15 Requests And Responses

There are no specific ES15 requests or responses defined in the present document.

D.9 Common Server Responses

For all responses with a JSON component the “subjectIdentifier” and “message” are optional and may or may not be present in the response received from the RSP server.

|  |  |
| --- | --- |
| Name | Content |
| R\_ERROR\_1\_2\_4\_2 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "1.2",  "reasonCode" : "4.2"  }  }  } } |
| R\_ERROR\_8\_1\_1\_2\_2 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1.1",  "reasonCode" : "2.2"  }  }  }  } |
| R\_ERROR\_8\_1\_1\_3\_8 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1.1",  "reasonCode" : "3.8"  }  }  } } |
| R\_ERROR\_8\_1\_1\_3\_10 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1.1",  "reasonCode" : "3.10"  }  }  }  } |
| R\_ERROR\_8\_1\_2\_6\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1.2",  "reasonCode" : "6.1"  }  }  } } |
| R\_ERROR\_8\_1\_2\_6\_3 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1.2",  "reasonCode" : "6.3"  }  }  } } |
| R\_ERROR\_8\_1\_3\_6\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1.3",  "reasonCode" : "6.1"  }  }  } } |
| R\_ERROR\_8\_1\_3\_6\_3 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1.3",  "reasonCode" : "6.3"  }  }  } } |
| R\_ERROR\_8\_1\_4\_8 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1",  "reasonCode" : "4.8"  }  }  } } |
| R\_ERROR\_8\_1\_6\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.1",  "reasonCode" : "6.1"  }  }  } } |
| R\_ERROR\_8\_2\_1\_2 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.2",  "reasonCode" : "1.2"  }  }  } } |
| R\_ERROR\_8\_2\_1\_3\_3 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData" : {  "subjectCode" : "8.2.1",  "reasonCode" : "3.3"  }  }  } } |
| R\_ERROR\_8\_2\_1\_3\_9 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData" : {  "subjectCode" : "8.2.1",  "reasonCode" : "3.9"  }  }  } } |
| R\_ERROR\_8\_2\_3\_7 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData" : {  "subjectCode" : "8.2",  "reasonCode" : "3.7"  }  }  }  } |
| R\_ERROR\_8\_2\_5\_4\_3 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.2.5",  "reasonCode" : "4.3"  }  }  } } |
| R\_ERROR\_8\_2\_6\_3\_3 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.2.6",  "reasonCode" : "3.3"  }  }  }  } |
| R\_ERROR\_8\_2\_6\_3\_8 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.2.6",  "reasonCode" : "3.8"  }  }  } } |
| R\_ERROR\_8\_2\_6\_3\_10 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.2.6",  "reasonCode" : "3.10"  }  }  }  } |
| R\_ERROR\_8\_2\_7\_2\_2 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData" : {  "subjectCode" : "8.2.7",  "reasonCode" : "2.2"  }  }  }  } |
| R\_ERROR\_8\_2\_7\_3\_8 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData" : {  "subjectCode" : "8.2.7",  "reasonCode" : "3.8"  }  }  }  } |
| R\_ERROR\_8\_2\_7\_6\_4 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.2.7",  "reasonCode" : "6.4"  }  }  }  } |
| R\_ERROR\_8\_8\_1\_3\_8 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.8.1",  "reasonCode" : "3.8"  }  }  } } |
| R\_ERROR\_8\_8\_2\_3\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.8.2",  "reasonCode" : "3.1"  }  }  } } |
| R\_ERROR\_8\_8\_3\_3\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.8.3",  "reasonCode" : "3.1"  }  }  } } |
| R\_ERROR\_8\_8\_3\_10 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.8",  "reasonCode" : "3.10"  }  }  } } |
| R\_ERROR\_8\_8\_4\_3\_7 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.8.4",  "reasonCode" : "3.7"  }  }  } } |
| R\_ERROR\_8\_8\_5\_4\_10 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.8.5",  "reasonCode" : "4.10"  }  }  } } |
| R\_ERROR\_8\_8\_5\_6\_4 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.8.5",  "reasonCode" : "6.4"  }  }  } } |
| R\_ERROR\_8\_9\_1\_3\_8 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9.1",  "reasonCode" : "3.8"  }  }  } } |
| R\_ERROR\_8\_9\_2\_3\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9.2",  "reasonCode" : "3.1"  }  }  } } |
| R\_ERROR\_8\_9\_3\_3\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9.3",  "reasonCode" : "3.1"  }  }  } } |
| R\_ERROR\_8\_9\_4\_2 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9",  "reasonCode" : "4.2"  }  }  } } |
| R\_ERROR\_8\_9\_4\_3\_7 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9.4",  "reasonCode" : "3.7"  }  }  } } |
| R\_ERROR\_8\_9\_5\_1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9",  "reasonCode" : "5.1"  }  }  } } |
| R\_ERROR\_8\_9\_5\_3\_3 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9.5",  "reasonCode" : "3.3"  }  }  } } |
| R\_ERROR\_8\_9\_5\_3\_9 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.9.5",  "reasonCode" : "3.9"  }  }  } } |
| R\_ERROR\_8\_10\_1\_3\_9 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.10.1",  "reasonCode" : "3.9"  }  }  } } |
| R\_ERROR\_8\_11\_1\_3\_9 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : "8.11.1",  "reasonCode" : "3.9"  }  }  } } |
| R\_ERROR\_ANY | {  "header" : {  "functionExecutionStatus" : {  "status" : "Failed",  "statusCodeData” : {  "subjectCode" : <SUBJECT\_CODE\_ANY>,  "reasonCode" : <REASON\_CODE\_ANY>  }  }  }  } |
| R\_ERROR\_SMXX\_1\_3\_8 | The error response will be as follows dependent on the entity under test:   for SM-DP+ testing on ES9+ SHALL be #R\_ERROR\_8\_8\_1\_3\_8   for SM-DS testing on ES11 SHALL be #R\_ERROR\_8\_9\_1\_3\_8 |
| R\_ERROR\_SMXX\_2\_3\_1 | The error response will be as follows dependent on the entity under test:   for SM-DP+ testing on ES9+ SHALL be #R\_ERROR\_8\_8\_2\_3\_1   for SM-DS testing on ES11 SHALL be #R\_ERROR\_8\_9\_2\_3\_1 |
| R\_ERROR\_SMXX\_3\_3\_1 | The error response will be as follows dependent on the entity under test:   for SM-DP+ testing on ES9+ SHALL be #R\_ERROR\_8\_8\_3\_3\_1   for SM-DS testing on ES11 SHALL be #R\_ERROR\_8\_9\_3\_3\_1 |
| R\_ERROR\_SMXX\_4\_3\_7 | The error response will be as follows dependent on the entity under test:   for SM-DP+ testing on ES9+ SHALL be #R\_ERROR\_8\_8\_4\_3\_7   for SM-DS testing on ES11 SHALL be #R\_ERROR\_8\_9\_4\_3\_7 |
| R\_SUCCESS | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  } } |

D.10 ES2+ Requests And Responses

**D.10.1 ES2+ Requests**

**D.10.2 ES2+ Responses**

|  |  |  |
| --- | --- | --- |
| Name | | Content |
| R\_SUCCESS\_ICCID1 | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  }  "iccid" : "#ICCID\_OP\_PROF1\_NON\_SWAP " } | |
| R\_SUCCESS\_MATCHING\_ID | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  }  "matchingId" : <MATCHING\_ID>  } | |
| R\_SUCCESS\_MATCHING\_ID\_EID | {  "header" : {  "functionExecutionStatus" : {  "status" : "Executed-Success"  }  }  "matchingId" : <MATCHING\_ID>  "eid" : "#EID1"  } | |

Annex E Profiles

|  |  |
| --- | --- |
| Profile | GENERIC\_PROFILE\_STRUCTURE |
| Description | Generic Operational Profile ASN.1 structure to be used as a basis for all Profiles used in this specification. |
| Details | headerValue ProfileElement ::= header : {  major-version 2,  minor-version 3,  profileType "GSMA Profile Package",  iccid '89019990001234567893'H,  eUICC-Mandatory-services {  usim NULL,  milenage NULL  },  eUICC-Mandatory-GFSTEList {  -- see Note 1  id-MF,  id-USIM  }  }  mfValue ProfileElement ::= mf : {  mf-header {  mandated NULL,  identification 1  },  templateID id-MF,  mf {  fileDescriptor : {  pinStatusTemplateDO '01020A'H  }  },  ef-pl {  fileDescriptor : {  -- EF PL modified to use Access Rule 15 within EF ARR  securityAttributesReferenced '0F'H  }  },  ef-iccid {  -- swapped ICCID: 98109909002143658739  fillFileContent '98109909002143658739'H  },  ef-dir {  fileDescriptor {  -- Shareable Linear Fixed File  -- 4 records, record length: 38 bytes  fileDescriptor '42210026'H,  efFileSize '98'H  },  -- USIM AID: A0000000871002FF33FF018900000100  fillFileContent  '61184F10A0000000871002FF33FF01890000010050045553494D'H  },  ef-arr {  fileDescriptor : {  fileDescriptor '42210025'H,  lcsi '05'H,  efFileSize '022B'H  },  fillFileContent : '8001019000800102A406830101950108800158A40683010A950108'H,  fillFileOffset : 10,  fillFileContent : '800101A40683010195010880015AA40683010A950108'H,  fillFileOffset : 15,  fillFileContent : '80015BA40683010A950108'H,  fillFileOffset : 26,  fillFileContent : '800101900080015A9700'H,  fillFileOffset : 27,  fillFileContent : '800103A406830101950108800158A40683010A950108'H,  fillFileOffset : 15,  fillFileContent : '800111A40683010195010880014AA40683010A950108'H,  fillFileOffset : 15,  fillFileContent : '800103A406830101950108800158A40683010A950108840132A406830101950108'H,  fillFileOffset : 4,  fillFileContent : '800101A406830101950108800102A406830181950108800158A40683010A950108'H,  fillFileOffset : 4,  fillFileContent : '800101900080011AA406830101950108800140A40683010A950108'H,  fillFileOffset : 10,  fillFileContent : '800101900080015AA40683010A950108'H,  fillFileOffset : 21,  fillFileContent : '8001019000800118A40683010A9501088001429700'H,  fillFileOffset : 16,  fillFileContent : '800101A40683010195010880015A9700'H,  fillFileOffset : 21,  fillFileContent : '800113A406830101950108800148A40683010A950108'H,  fillFileOffset : 15,  fillFileContent : '80015EA40683010A950108'H,  fillFileOffset : 26,  fillFileContent '8001019000800102A010A40683010195  0108A406830102950108800158A40683  010A950108'H  }  }  pukVal ProfileElement ::= pukCodes : {  puk-Header {  mandated NULL,  identification 2  },  pukCodes {  {  keyReference pukAppl1,  pukValue '3030303030303030'H,  -- maxNumOfAttemps:9, retryNumLeft:9  maxNumOfAttemps-retryNumLeft 153  },  {  keyReference pukAppl2,  pukValue '3132333435363738'H  },  {  keyReference secondPUKAppl1,  pukValue '3932393435363738'H,  -- maxNumOfAttemps:8, retryNumLeft:8  maxNumOfAttemps-retryNumLeft 136  }  }  }  pinVal ProfileElement ::= pinCodes : {  pin-Header {  mandated NULL,  identification 3  },  pinCodes pinconfig : {  {  keyReference pinAppl1,  pinValue '31323334FFFFFFFF'H,  unblockingPINReference pukAppl1  },  {  keyReference pinAppl2,  pinValue '30303030FFFFFFFF'H,  unblockingPINReference pukAppl2  },  {  keyReference adm1,  pinValue '35363738FFFFFFFF'H,  pinAttributes 1  }  }  }  usimValue ProfileElement ::= usim : {  usim-header {  mandated NULL,  identification 4  },  templateID id-USIM,  adf-usim {  fileDescriptor : {  fileID '7FF1'H,  dfName 'A0000000871002FF33FF018900000100'H,  pinStatusTemplateDO '01810A'H  }  },  ef-imsi {  -- numerical format: 234101943787656  fillFileContent '082943019134876765'H  },  ef-arr {  fileDescriptor {  linkPath '2F06'H  }  },  ef-ust {  -- Service Dialling Numbers, Short Message Storage…  fillFileContent '0A2E178CE73204000000000000'H  },  ef-spn {  -- ASCII format: "GSMA eUICC"  fillFileContent '0247534D41206555494343FFFFFFFFFFFF'H  },  ef-est {  -- Services deactivated  fillFileContent '00'H  },  ef-acc {  -- Access class 4  fillFileContent '0040'H  },  ef-ecc {  -- Emergency Call Code 911  fillFileContent '19F1FF01'H  }  }  usimPin ProfileElement ::= pinCodes : {  pin-Header {  mandated NULL,  identification 5  },  pinCodes pinconfig : {  {  keyReference secondPINAppl1,  pinValue '39323338FFFFFFFF'H  unblockingPINReference secondPUKAppl1,  -- PIN is Enabled  pinAttributes 1,  -- maxNumOfAttemps:2, retryNumLeft:2  maxNumOfAttemps-retryNumLeft 34  }  }  }  akaParamValue ProfileElement ::= akaParameter : {  aka-header {  mandated NULL,  identification 6  },  algoConfiguration algoParameter : {  algorithmID milenage,  -- RES and MAC 64 bits, CK and IK 128 bits  algorithmOptions '01'H,  key '000102030405060708090A0B0C0D0E0F'H,  opc '0102030405060708090A0B0C0D0E0F00'H,  -- rotationConstants uses default: '4000204060'H  -- xoringConstants uses default value  authCounterMax '010203'H  }  -- sqnOptions uses default: '02'H  -- sqnDelta uses default: '000010000000'H  -- sqnAgeLimit uses default: '000010000000'H  -- sqnInit uses default: all bytes zero  }  mnoSdValue ProfileElement ::= securityDomain : {  sd-Header {  mandated NULL,  identification 7  },  instance {  applicationLoadPackageAID 'A0000001515350'H,  classAID 'A000000151535041'H,  instanceAID 'A000000151000000'H,  applicationPrivileges '82FC80'H,  -- Secured  lifeCycleState '0F'H,  -- SCP80 supported  applicationSpecificParametersC9 '810280008201F08701F0'H,  -- other parameters MAY be necessary  applicationParameters {  -- TAR: B20100, MSL: 12  uiccToolkitApplicationSpecificParametersField  '0100000100000002011203B2010000'H  }  },  keyList {  {  -- C-ENC + R-ENC  keyUsageQualifier '38'H,  -- ENC key  keyIdentifier '01'H,  keyVersionNumber '01'H,  keyCompontents {  {  -- DES mode implicitly known (as an example)  keyType '80'H,  -- This value MAY be freely changed  keyData '112233445566778899AABBCCDDEEFF10'H  }  }  },  {  -- C-MAC + R-MAC  keyUsageQualifier '34'H,  -- MAC key  keyIdentifier '02'H,  keyVersionNumber '01'H,  keyCompontents {  {  -- DES mode implicitly known (as an example)  keyType '80'H,  -- This value MAY be freely changed  keyData '112233445566778899AABBCCDDEEFF10'H  }  }  },  {  -- C-DEK + R-DEK  keyUsageQualifier 'C8'H,  -- data ENC key  keyIdentifier '03'H,  keyVersionNumber '01'H,  keyCompontents {  {  -- DES mode implicitly known (as an example)  keyType '80'H,  -- This value MAY be freely changed  keyData '112233445566778899AABBCCDDEEFF10'H  }  }  },  -- AES Token Key (as an example)  -- This value MAY be freely changed  keyUsageQualifier '81'H,  -- MAY be used by SD  keyAccess '01'H,  -- Key Id 01  keyIdentifier '01'H,  keyVersionNumber '70'H,  keyCompontents {  {  -- AES (16 bytes key length)  -- This value MAY be freely changed  keyType '88'H,  -- This value MAY be freely changed  keyData 'CDFE56B7B72FAE6A047341F003D7A48D'H  }  }  },  {  -- Receipt (the AES scheme SHALL be supported)  keyUsageQualifier '44'H,  -- MAY be used by SD  keyAccess '01'H,  -- Key Id 01  keyIdentifier '01'H,  keyVersionNumber '71'H,  keyCompontents {  {  -- AES (16 bytes key length)  keyType '88'H,  -- This value MAY be freely changed  keyData '11121314212223243132333441424344'H  }  }  }  }  }  ssdValue ProfileElement ::= securityDomain : {  sd-Header {  mandated NULL,  identification 8  },  instance {  applicationLoadPackageAID 'A0000001515350'H,  classAID 'A000000151535041'H,  instanceAID 'A00000055910100102736456616C7565'H,  -- by default extradited under MNO-SD  -- Privileges: Security Domain + Trusted Path  applicationPrivileges '808000'H,  -- Personalized  lifeCycleState '0F'H,  -- SCP80 supported, extradiction supported  applicationSpecificParametersC9 '810280008201F0'H,  applicationParameters {  -- TAR: 6C7565, MSL: 12  uiccToolkitApplicationSpecificParametersField  '01000001000000020112036C756500'H  }  },  keyList {  {  -- C-ENC + R-ENC  keyUsageQualifier '38'H,  keyIdentifier '01'H,  keyVersionNumber '01'H,  keyCompontents {  {  -- DES mode implicitly known (as an example)  keyType '80'H,  -- This value MAY be freely changed  keyData '11223344556677881122334455667788'H  }  }  },  {  -- C-MAC + R-MAC  keyUsageQualifier '34'H,  -- MAC key  keyIdentifier '02'H,  keyVersionNumber '01'H,  keyCompontents {  {  -- DES mode implicitly known (as an example)  keyType '80'H,  -- This value MAY be freely changed  keyData '11223344556677881122334455667788'H  }  }  },  {  -- C-DEK + R-DEK  keyUsageQualifier 'C8'H,  -- data ENC key  keyIdentifier '03'H,  keyVersionNumber '01'H,  keyCompontents {  {  -- DES mode implicitly known (as an example)  keyType '80'H,  -- This value MAY be freely changed  keyData '11223344556677881122334455667788'H  }  }  }  }  }  rfmUicc ProfileElement ::= rfm : {  rfm-header {  identification 11  },  -- Instance AID  instanceAID ' A00000055910100001'H,  tarList {  'B00000'H  },  -- cryptographic checksum + counter higher  minimumSecurityLevel '12'H,  -- full access  uiccAccessDomain '00'H,  -- full access  uiccAdminAccessDomain '00'H  }  rfmUsim ProfileElement ::= rfm : {  rfm-header {  identification 12  },  -- Instance AID  instanceAID 'A00000055910100002'H,  tarList {  'B00020'H  },  -- cryptographic checksum + counter higher  minimumSecurityLevel '12'H,  -- full access  uiccAccessDomain '00'H,  -- full access  uiccAdminAccessDomain '00'H,  adfRFMAccess {  adfAID 'A0000000871002FF33FF018900000100'H,  -- UICC access condition: ADM1  adfAccessDomain '02000100'H,  -- UICC access condition: ADM1  adfAdminAccessDomain '02000100'H  }  }  endValue ProfileElement ::= end : {  end-header {  mandated NULL,  identification 99  }  } |
| *Note 1: The following OIDs are used:*  *id-MF OBJECT IDENTIFIER ::=*  *{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) mf(1)}*  *id-USIM OBJECT IDENTIFIER ::=*  *{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) usim(4)}* | |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL1 |
| Description | Operational Profile  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF1, except if defined differently in the test sequence.  The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF1 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF1   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF1   the pinAttributes of pinAppl1 present in the PE\_PIN SHALL be set to 6   the SCP80 encryption key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO\_SCP80\_ENC\_KEY   the SCP80 message authentication key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO\_SCP80\_AUTH\_KEY   the SCP80 data encryption key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO\_SCP80\_DATA\_ENC\_KEY   the instance AID configured in the PE-SecurityDomain that corresponds to the Supplementary Security Domain PE\_SSD SHALL be set to #SSD\_AID   the ef-dir present in the PE-MF SHALL be configured with the AID #USIM\_AID   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H  The PROFILE\_OPERATIONAL1 UPP is named #UPP\_OP\_PROF1 in the scope of this document. |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL2 |
| Description | Operational Profile  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF2, except if defined differently in the test sequence.  The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF2 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF2   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF2   The pinAttributes of pinAppl1 present in the PE\_PIN SHALL be set to 6   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H  The PROFILE\_OPERATIONAL2 UPP is named #UPP\_OP\_PROF2 in the scope of this document. |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL3 |
| Description | Operational Profile with PPR2 but without notification  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF3, except if defined differently in the test sequence.  The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF3 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF3   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF3   the pinAttributes of pinAppl1 present in the PE\_PIN SHALL be set to 6   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H  The PROFILE\_OPERATIONAL3 UPP is named #UPP\_OP\_PROF3 in the scope of this document. |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL4 |
| Description | Operational Profile with PPR1 and notification  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF4, except if defined differently in the test sequence.  The Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE] except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF4 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF4   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF4   the pinAttributes of pinAppl1 present in the PE\_PIN SHALL be set to 6   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H  The PROFILE\_OPERATIONAL4 UPP is named #UPP\_OP\_PROF4 in the scope of this document. |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL5 |
| Description | Operational Profile with pinAppl1 enabled.  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF5, except if defined differently in the test sequence.  The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF5 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF5   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF5   the pinAppl1 present in the PE\_PIN SHALL be enabled and has the value #PO1\_PIN1   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL6 |
| Description | Operational Profile with pinAppl1 enabled.  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF6, except if defined differently in the test sequence.  The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF6 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF6   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF6   The pinAppl1 present in the PE\_PIN SHALL be enabled and has the value #PO2\_PIN1   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL7 |
| Description | Operational Profile with PPR2 and notification  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF7, except if defined differently in the test sequence.  The Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF7 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF7   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF7   the pinAttributes of pinAppl1 present in the PE\_PIN SHALL be set to 6   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL8 |
| Description | Operational Profile with PPR2, pinAppl1 enabled and notification  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF8, except if defined differently in the test sequence.  The Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF8 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF8   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF8   The pinAppl1 present in the PE\_PIN SHALL be enabled and has the value #PO2\_PIN1   the ef-ust SHALL be set in accordance to #EF\_UST1 (service 17 and 18 are not available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H |

|  |  |
| --- | --- |
| Profile | PROFILE\_OPERATIONAL9 |
| Description | Operational Profile with GID1 and GID2 set  This Profile acts as an Operational Profile in the scope of this specification.  NOTE: Milenage algorithm is used in this Profile |
| Details | The Profile Metadata SHALL be set to #METADATA\_OP\_PROF9, except if defined differently in the test sequence.  The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC\_PROFILE\_STRUCTURE except that:   the *iccid* field SHALL be set to #ICCID\_OP\_PROF9 in the *ProfileHeader* element, in non-swapped format   the ef-iccid present in the PE-MF SHALL be set to #ICCID\_OP\_PROF9   the ef-imsi present in the PE-USIM SHALL be set to #IMSI\_OP\_PROF9   the pinAppl1 present in the PE\_PIN SHALL be enabled and has the value #PO1\_PIN1   the ef-ust SHALL be set to #EF\_UST2 (service 17 and 18 are available)   the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H   * the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD    the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H   the following new Profile Element PE\_OPT\_USIM SHALL be inserted right after PE\_USIM:   |  | | --- | | PE\_OPT\_USIM | | optusimValue ProfileElement ::= opt-usim : {  optusim-header {  mandated NULL,  identification 15  },  templateID id-OPT-USIM,  ef-gid1 {  fileDescriptor {        efFileSize '04'H      },  fillFileContent #GID1  },  ef-gid2 {  fileDescriptor {        efFileSize '04'H      },  fillFileContent #GID2  }  } | | NOTE : The following OIDs are used:  id-OPT-USIM OBJECT IDENTIFIER ::=  {joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-usim(5)} |   The PROFILE\_OPERATIONAL9 UPP is named #UPP\_OP\_PROF9 in the scope of this document. |

Annex F IUT Settings

F.1 eUICC Settings

In order to execute the test cases defined in this document, the eUICC Manufacturer SHALL deliver following settings:

|  |  |
| --- | --- |
| eUICC Setting name | Description |
| IUT\_DLOA\_URL | Discovery Base URL of the SE default DLOA Registrar as defined in GlobalPlatform DLOA specification [19] (optional) |
| IUT\_EUICC\_ADD\_PP\_VERSIONS | The expected content, if any, of the additionalEuiccProfilePackageVersions field in EUICCInfo2, coded as the concatenation of the binary values of the value parts of the contained VersionType(s) – i.e. without tags and lengths.  Examples:   * 0x030200 (encoding v3.2)   0x030100030200 (encoding v3.1 and v3.2; note that this example is correct syntactically, but not semantically, as an eUICC should only indicate a single v3.x version)  This setting is only applicable for eUICCs supporting SGP.22 v2.3 or later. |
| IUT\_EUICC\_CATEGORY | The category, if provided, SHALL be either not present or:   other(0)   or basicEuicc(1)   or mediumEuicc(2)   or contactlessEuicc(3) |
| IUT\_EUICC\_FIRMWARE\_VER | eUICC Firmware version coded as binary value (3 bytes representing major/minor/revision). |
| IUT\_GLOBALPLATFORM\_VERSION | GlobalPlatform version coded as binary value (3 bytes representing major/minor/revision, 2.3.0 or higher). The support of GlobalPlatform is considered as mandatory in the scope of this specification. |
| IUT\_PLATFORM\_LABEL | Platform\_Label as defined in GlobalPlatform DLOA specification [19] (optional) |
| IUT\_PP\_VERSION | Protection Profile version coded as binary value (3 bytes representing major/minor/revision). |
| IUT\_SAS\_ACCREDITATION\_NUMBER | SAS Accreditation Number, coded as ASN.1 UTF8String |
| IUT\_SIMA\_VERSION | Version of eUICC Profile Package Specification [4] supported by the eUICC (3 bytes representing major/minor/revision) e.g. 0x020100 |
| IUT\_TS102241\_VERSION | The ts102241 version field is coded as binary value (3 bytes representing major/minor/revision, 9.0.0 or higher). The support of Java Card is considered as mandatory in the scope of this specification.  The ts102241 Version field indicates the latest version of ETSI TS102 241[17] supported by the eUICC |
| IUT\_UICC\_CAPABILITY | Sequence is derived from ServicesList[] defined in eUICC Profile Package PEDefinitions, coded as ASN.1 BIT STRING. |

F.2 Platforms Settings

In order to execute the test cases defined in this document, the Platform (i.e. SM-DP+ or SM-DS) provider SHALL deliver following settings:

|  |  |
| --- | --- |
| SM-DP+ Setting name | Description |
| IUT\_SM\_DP\_ADDRESS | FQDN of the SM-DP+ Under Test. |
| IUT\_SM\_DP\_HOST\_ID | SM-DP+ Host ID of the SM-DP+ Under Test coded as an ASN.1 octet string. |
| IUT\_SM\_DP\_OID | SM-DP+ OID (as defined in section 1.3) of the SM-DP+ Under Test. |
| IUT\_SM‑DP+\_MAX\_NUMBER\_DOWNLOAD\_ATTEMPTS | Maximum number of download attempts allowed by the SM-DP+. After this number, no further download is allowed. |
| IUT\_SM\_DP\_ADDRESS\_ES2\_PLUS | FQDN, or FQDN:<port> of the SM-DP+ Under Test ES2+ Interface.  This Value SHALL be different from the IUT\_SM\_DP\_ADDRESS |
| SM-DS Setting name | Description |
| IUT\_SM\_DS\_ADDRESS\_ES11 | FQDN of the SM-DS Under Test for access on ES11. |
| IUT\_SM\_DS\_ADDRESS\_ES12 | FQDN of the SM-DS Under Test for access on ES12. |
| IUT\_SM\_DS\_ADDRESS\_ES15 | FQDN of the SM-DS Under Test for access on ES15. |
| Shared Setting name | Description |
| IUT\_CLIENT\_TLS\_VER | Applicability: this IUT setting is applicable for SM-DP+ and Alternative SM-DS. It is not applicable for Root SM-DS.  Highest TLS protocol version supported by the Client (SM-DP+ on ES12 or Alternative SM-DS on ES15) under test, which SHALL be at least v1.2. For versions higher than TLS v1.2 backwards compatibility is assumed. |

F.3 Device Settings

| Device Setting name | Description |
| --- | --- |
| IUT\_CDMA2000\_1X\_REL | If cdma2000 1X is supported, this SHALL be encoded as the octet string {1, 0, 0}. |
| IUT\_CDMA2000\_EHRPD\_REL | If cdma2000 eHRPD, is supported this SHALL be the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}. |
| IUT\_CDMA2000\_HRPD\_REL | If cdma2000 HRPD is supported, this SHALL be encoded as the octet string {R, 0, 0}. The value R SHALL represent the EVDO revision as follows: Rev 0 SHALL be encoded as 1 Rev A SHALL be encoded as 2 Rev B SHALL be encoded as 3 |
| IUT\_EU\_CONFIRMATION\_TIMEOUT | Timeout in seconds for LPAd for the End User Intent confirmation starting when the LPAd displays the dialog for confirmation. |
| IUT\_GSM\_GERAN\_REL | If GSM/GERAN is supported, this is the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}. |
| IUT\_IMEI | International Mobile Equipment Identity value of the Device in human readable format, including the check digit. The value is used as a reference for verification of the TAC (mandatory) and IMEI (optional) retrieved from DeviceInfo. |
| IUT\_LPAd\_Confirmation | Description of the way to perform Authenticated Confirmation (devices supporting SGP.22 v2.2.2 or earlier) or Strong Confirmation (devices supporting SGP.22 v2.3 or later). |
| IUT\_LPAd\_CI | CI subjectPublicKeyInfo of CERT.CI.ECDSA (used to verify CERT.DP.TLS) stored in LPAd. Based on NIST [11] in this version of specification. |
| IUT\_LPAd\_NOTIFICATION\_TIMEOUT | Timeout in seconds for LPAd to send a Notification to the SM-DP+ on ES9+ interface assuming IP connection is available. |
| IUT\_LPAd\_READY\_AFTER\_REBOOT\_TIMEOUT | Timeout in seconds for the LPAd to be ready after a reboot. The time starts from the power off at the start of the reboot and ends when the LPAd is ready after the reboot. |
| IUT\_LPAd\_SESSION\_CLOSE\_TIMEOUT | Timeout in seconds for LPAd to send a next command for Profile Download to the SM-DP+ (or SM-DS) on ES9+ (or ES11) interface assuming IP connection is available. The timeout SHALL start after sending of the previous command by the LPAd. |
| IUT\_LTE\_EUTRAN\_REL | If LTE/E-UTRAN is supported, this SHALL be the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}. |
| IUT\_NFC\_REL | If NFC is supported, this SHALL be the highest (version, revision) number of TS.26 [15], encoded as the octet string {version, revision, 0}. |
| IUT\_TLS\_VERSION | Highest TLS protocol version supported by LPAd, at least v1.2. By versions higher then TLS v1.2 backwards compatibility is assumed. |
| IUT\_UMTS\_UTRAN\_REL | If UMTS/UTRAN is supported, this SHALL be the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}. |

F.4 Common Settings

In order to execute the test cases defined in this document, the IUT provider SHALL deliver following settings:

|  |  |
| --- | --- |
| IUT Setting name | Description |
| IUT\_RSP\_VERSION | Version of SGP.22 supported by the IUT encoded as a string of three integers separated with dots (for example: 2.1.0).  In the scope of this specification, this value SHALL indicate one of the versions of SGP.22 for which this specification contains test cases, as specified in section 1.2. |

Annex G Initial States

Unless it is defined differently in a particular test case, the IUTs SHALL be set in the following initial state before the test case execution.

G.1 Device

G.1.1 Device (default)

The Device is “powered on”.

The Device is in the normal execution mode after Device boot-up and Device initial configuration. The Device is NOT in the Test Mode.

The LPAd has access to the root CI key #CERT\_CI\_ECDSA (or the CI public key) for verification of the TLS certificates of SM-DP+ or SM-DS. No CRL is loaded.

The Device contains a Test eUICC pre-configured as defined below in G.1.3.

G.1.2 Companion Device connected to a Primary Device

The Companion Device is connected to the Primary Device as defined by the Device vendor.

Companion Device and the connected Primary Device are “powered on”.

The Companion Device and Primary Device are in the normal execution mode (NOT in the boot-up mode).

The LPAd of the Companion Device has access to the root CI #CERT\_CI\_ECDSA (or the CI public key) for verification of the TLS certificates of SM-DP+ or SM-DS. No CRL is loaded.

The Companion Device contains a Test eUICC preconfigured as defined below in G.1.3.

G.1.3 Test eUICC Settings

Depending on the test cases and on the supported options, the Test eUICC SHALL be configured according to the following Initial States.

 The Test eUICC is configured with the ISD-R AID #ISD\_R\_AID and the EID #EID1.

 The Test eUICC does not contain any Profile.

 The Test eUICC is configured with the default SM-DS address #TEST\_ROOT\_DS\_ADDRESS.

 The Test eUICC contains #TEST\_DP\_ADDRESS1 as default SM-DP+ address.

The ECASD is configured with at least the following Keys and Certificates based on NIST P-256 [11] or on brainpoolP256r1 [8] for this version of the SGP.23:

 The Test eUICC’s Private Key #SK\_EUICC\_ECDSA (for creating ECDSA signatures)

 The Test eUICC’s Certificate #CERT\_EUICC\_ECDSA (for eUICC authentication) containing the eUICC’s Public Key #PK\_EUICC\_ECDSA

 The GSMA Certificate Issuer’s Public Key #PK\_CI\_ECDSA (for verifying off-card entities certificates)

 The Certificate of the EUM #CERT\_EUM\_ECDSA

Other Certificates and Keys MAY be present. No CRL is loaded on the Test eUICC.

The CI, identified as highest priority in euiccCiPKIdListForSigning, is also selectable in the euiccCiPKIdListForVerification (i.e. all EUM and eUICC Certificates lead to a Root CI certificate linked to a #PK\_CI\_ECDSA contained in the eUICC).

This CI corresponds to the SubjectKeyIdentifier of one of the #CERT\_CI\_ECDSA defined in sections G.2.2 and G.2.3.

For devices supporting O\_D\_REMOVABLE\_DOWNLOAD\_PPR, the Test eUICC SHALL contain the RAT configuration specified in #PPRS\_ALLOWED.

For devices supporting a removable eUICC but not supporting O\_D\_REMOVABLE\_DOWNLOAD\_PPR, the Test eUICC can be configured with any RAT.

For devices supporting a non-removable eUICC:

* For some combinations of device options, RAT configurations with certain constraints are required for some sequences, as specified below. These constraints can be satisfied using any valid RAT table; for example, Allowed Operators can be specified explicitly or using wildcards.

|  |  |
| --- | --- |
| **Device option(s) supported** | **RAT configuration of Test eUICC** |
| O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ | PPR1 is allowed and End User Consent is required for #MCC\_MNC4 with gid1 and gid2 absent. |
| O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_REQ | PPR2 is allowed and End User Consent is required for #MCC\_MNC2 with gid1 and gid2 absent. |
| NOT O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_REQ AND O\_D\_EMB\_ALLOWS\_PPR1\_EUC\_NOT\_REQ | PPR1 is allowed and End User Consent is not required for #MCC\_MNC4 with gid1 and gid2 absent. |
| NOT O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_REQ AND O\_D\_EMB\_ALLOWS\_PPR2\_EUC\_NOT\_REQ | PPR2 is allowed and End User Consent is not required for #MCC\_MNC2 with gid1 and gid2 absent. |

* If none of the constraints above apply, the Test eUICC can be configured with any RAT.
* Note: in the current version of this document, it is possible to satisfy the relevant constraints above with a single RAT configuration. It is recommended to supply a single device for testing with the RAT configuration satisfying all of the relevant constraints above, rather than to supply multiple devices.

A separate Test eUICC needs to be provided for each additional RAT configuration (not used in this version of the test specification). In case the Test eUICC is non-removable the additional Device SHALL contain the same software and hardware except the Test eUICC configuration.

G.2 eUICC

Depending on the test cases and on the supported options, the EUM SHALL configure the eUICC according to the following Initial States. The initial conditions SHALL be restored, as described in the following subsections, after each test sequence.

G.2.1 Common Initial States

The following initial states apply for all test cases defined in this Test Plan whatever the options supported by the eUICC:

 The eUICC is configured with the ISD-R AID #ISD\_R\_AID and the EID #EID1.

 The eUICC does not contain any Profile.

 The default files system does not contain EF\_ICCID.

 The eUICC’s Pending Notifications List is empty.

 No RSP session is ongoing.

 The eUICC is configured with the default SM-DS address #TEST\_ROOT\_DS\_ADDRESS.

 The eUICC is configured without Default SM-DP+ address.

 No CRL is loaded on the eUICC.

 The ECASD is configured as defined in section G.2.2 and/or G.2.3 depending on the support of the options O\_E\_NIST and O\_E\_BRP.

o If the eUICC only supports O\_E\_NIST, the ECASD is configured as defined in section G.2.2.

o If the eUICC only supports O\_E\_BRP, the ECASD is configured as defined in section G.2.3.

o If the eUICC supports O\_E\_NIST and O\_E\_BRP, the ECASD is configured as defined in sections G.2.2 and G.2.3 (i.e. several EUM / eUICC Certificates and Keys are configured in the eUICC).

The CI, identified as highest priority in euiccCiPKIdListForSigning, is also selectable in the euiccCiPKIdListForVerification (i.e. all EUM and eUICC Certificates lead to a Root CI certificate linked to a #PK\_CI\_ECDSA contained in the eUICC).

This CI corresponds to the SubjectKeyIdentifier of one of the #CERT\_CI\_ECDSA defined in sections G.2.2 and G.2.3.

The default RAT configuration defined in section G.2.4 applies for all test sequences except if the Test Case overrides it. Particular RAT configurations for those specific Test Cases are defined in section G.2.5.

G.2.2 For eUICC supporting NIST P-256

If the eUICC supports O\_E\_NIST, the ECASD contains at least:

 The eUICC’s Private Key #SK\_EUICC\_ECDSA (for creating ECDSA signatures) based on NIST P-256 [11]

 The eUICC’s Certificate #CERT\_EUICC\_ECDSA (for eUICC authentication) containing the eUICC’s Public Key #PK\_EUICC\_ECDSA based on NIST P-256 [11]

 The GSMA Certificate Issuer’s Public Key #PK\_CI\_ECDSA (for verifying off-card entities certificates) based on NIST P-256 [11]

 The Certificate of the EUM #CERT\_EUM\_ECDSA based on NIST P-256 [11]

Other Certificates and Keys MAY be present.

G.2.3 For eUICC supporting BrainpoolP256r1

If the eUICC supports O\_E\_BRP, the ECASD contains at least:

 The eUICC’s Private Key #SK\_EUICC\_ECDSA (for creating ECDSA signatures) based on brainpoolP256r1 [8]

 The eUICC’s Certificate #CERT\_EUICC\_ECDSA (for eUICC authentication) containing the eUICC’s Public Key #PK\_EUICC\_ECDSA based on brainpoolP256r1 [8]

 The GSMA Certificate Issuer’s Public Key #PK\_CI\_ECDSA (for verifying off-card entities certificates) based on brainpoolP256r1 [8]

 The Certificate of the EUM #CERT\_EUM\_ECDSA based on brainpoolP256r1 [8]

 Other Certificates and Keys MAY be present.

G.2.4 With default RAT configuration

The eUICC’s RAT is configured as detailed in SGP.21 Annex H:

 Only one PPAR authorizing PPR1 and PPR2 for all MNOs with End User consent required i.e. #PPRS\_ALLOWED

G.2.5 With Additional PPARs in the RAT

The eUICC’s RAT is configured as below (following this order):

 Additional PPARs representing custom agreements between MNOs and OEMs:

o #PPR1\_WITH\_OWNER\_GID

o #PPR1\_WITHOUT\_GID

o #PPR2\_WITHOUT\_CONSENT

 The last PPAR authorizes PPR1 and PPR2 for all MNOs with End User consent required i.e. #PPRS\_ALLOWED

G.2.6 Clean-up procedure

Unless differently specified in the test case, the following procedure SHALL be executed after each test sequence to bring the eUICC back to its Common Initial State:

* eUICC Memory Reset to delete all profiles and reset the SM-DP+ Address
* Retrieve and Remove all pending notifications

Where necessary, in addition to the above, other steps may be executed to restore the initial state specified in this Annex.

G.3 SM-DP+ and SM-DS

The SM-DP+ SHALL be configured with #CERT\_SM\_DPauth\_ECDSA, #CERT\_SM\_DPpb\_ECDSA and #CERT\_SM\_DP\_TLS for both NIST and BRP unless it is specified differently to verify specific configuration (e.g. test cases dedicated for NIST or BRP only).

The SM-DP+ provider SHALL provide the capability to provision the SM-DP+ with Profiles as required by the specific test cases, with the following associated data where required:

 Profile Metadata

 MatchingID

 EID

 Confirmation Code

 Protected with random keys in advance, or with session keys during an RSP session, as required

 Number of retries for receipt of a valid Confirmation Code.

The SM-DP+ provider SHALL provide the capability to expire a download order.

NOTE: as ES2+ is out of scope in the current version of the present document, proprietary means MAY be used to provide these capabilities.

The SM-DS SHALL be configured with #CERT\_SM\_DSauth\_ECDSA and #CERT\_SM\_DS\_TLS for both NIST and BRP, unless it is specified differently to verify specific configuration (e.g. test cases dedicated for NIST or BRP only).

For TLS level and for the SM-XX testing, NIST shall be used unless it is specified otherwise.

The SM-DS provider SHALL provide the capability to register an event.

The SM-DS provider SHALL provide the capability to remove the record of a particular EventID having been used from the SM-DS.

Annex H Icons and QR Codes

The files for the eUICC Consumer Devices Icons and QR Codes are provided within in SGP.23\_AnnexH\_Icons.zip and SGP.23\_AnnexH\_QRCodes.zip packages, which accompany the present document.

Annex I Requirements

The requirements used in the specified test cases are provided within SGP\_23\_AnnexI\_Requirements\_v1\_3.zip package, which accompanies the present document.

Annex J Integrated eUICC Testing (Normative)

**J.1 Overview (Informative)**

An Integrated eUICC hardware resides in an SoC along with other subsystems such as general processing and mobile broadband modem, all connected through a proprietary SoC interconnect channel. Alternatively, an Integrated eUICC may communicate with a mobile broadband modem external to the SoC via an external interface, which may be proprietary or based on a standard not associated with UICC. As such, Integrated eUICC may not include a physical UICC-Terminal interface [5].

In order to test the functionality and compliancy of an Integrated eUICC, hardware and OS, Integrated eUICC manufacturers need to provide and support a test interface to which testing equipment can be connected to. Having a standardized testing interface, will increase interoperable and reusability between different manufacturer of Integrated eUICC and test equipment.

For Integrated eUICC with a USB CCID [29] test interface, this annex describes its properties. In cases where a USB interface is not available in a device containing an Integrated eUICC, an adapter to USB CCID needs to be provided, e.g. Bluetooth to USB CCID. The functionality needed to provide and support the test interface, shall be considered part of the test environment and not the IUT.



**Figure 1 Integrated eUICC with USB CCID [29] Test Interface**

Note: The mechanism providing USB CCID to the RSP eUICC Test System, and described in this Annex, is implementation specific. As such, it may be implemented in the SoC, on-Device, off-Device or any combination thereof.

**J.2 Integrated eUICC test requirements**

An Integrated eUICC manufacturer shall provide a USB CCID test interface implementing the functionality specified in J.3.

The test interface shall maintain the integrity and order of the data between the Integrated eUICC and the test system.

The Integrated eUICC manufacturer shall ensure that during testing no other clients or SoC subsystems interfere with the testing.

The Integrated eUICC may use any physical or logical interface between the Integrated eUICC and the test system, as long as a USB CCID is provided to the test system and the channel is reliable (i.e. maintain integrity and order).

**J.3 USB CCID test interface**

The Integrated eUICC USB CCID test interface shall operate in a card reader mode.

The Integrated eUICC USB CCID test interface shall support the following [29] section 6 messages:

* [29] section 6.1 Messages:
  + PC\_to\_RDR\_IccPowerOn
  + PC\_to\_RDR\_IccPowerOff
  + PC\_to\_RDR\_GetSlotStatus
  + PC\_to\_RDR\_Escape
  + PC\_to\_RDR\_XfrBlock
  + PC\_to\_RDR\_T0APDU
  + PC\_to\_RDR\_Secure
  + PC\_to\_RDR\_Abort
* [29] section 6.2 Messages:
  + RDR\_to\_PC\_SlotStatus
  + RDR\_to\_PC\_Escape
  + RDR\_to\_PC\_DataBlock

Note: For test systems using wincard.h/PCSC lite APIs to connect to the Integrated eUICC USB CCID test interface, the following APIs are expected to be used:

* SCardEstablishContext
* SCardListReaders[A|W]
* SCardConnect[A|W]
* SCardControl
* SCardTransmit
* SCardDisconnect
* SCardStatus[A|W]
* SCardReleaseContext
* SCardReconnect
* SCardBeginTransaction
* SCardEndTransaction
* SCardGetStatusChange
* SCardFreeMemory
* SCardGetAttrib

Annex K Document Management

J.1 Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date | CR Number | Brief Description of Change | Approval Authority | Editor / Company |
| v1.0 | 9th June 2017 |  | Initial version of SGP.23 v1.0 Test Specification | PSMC | Yolanda Sanz, GSMA |
| v1.1 | 28th Sept 2017 |  | Minor version of SGP.23 Test specifications | RSPLEN | Yolanda Sanz, GSMA |
| v1.2 | 3rd Jan 2018 |  | Minor version of SGP.23 Test specifications | RSPLEN | Yolanda Sanz, GSMA |
| V1.3 | 01th August |  | Minor version of SGP.23 Test specification | RSPLEN | Yolanda Sanz, GSMA |
| V1.4 | 18th Dec |  | Minor version of SGP.23 Test specification | RSPLEN | Yolanda Sanz, GSMA |
| V1.5 | 30 April 2019 |  | Minor version of SGP.23 Test specification | RSPLEN | Marcin Kulczycki, GSMA |
| V1.6 | 15 July 2019 |  | Minor version of SGP.23 Test specification | eSIM Group | Marcin Kulczycki, GSMA |
| V1.7 | 07 July 2020 |  | Minor version of SGP.23 Test specification | ISAG | Yolanda Sanz, GSMA |
| V1.8 | 22  October 2020 |  | Minor version of SGP.23 Test specification | ISAG | Yolanda Sanz, GSMA |
| V1.9 | 11 February 2021 | CR1900R01 | LPA: To remove non validated TLS test | ISAG | Yolanda Sanz, GSMA |
| CR1901R01 | eUICC: Aligment of Applicability table |
| CR1902R03 | LPA: remove TLS critical extension presence test sequence |
| V1.10 | 30 June 2021 | CR11006R00 | SM-DP+\_ES2+ TestEnviorment | ISAG | Yolanda Sanz, GSMA |
| CR11003R04 | SM-DP+\_ES2+\_DownloadOrder-1. |
| CR11004R02 | SM-DP+ ES2+\_DownloadOrder-2,3 |
| CR11007R02 | LPA Simplified End User Confirmation |
| CR11010R01 | LPA Revising User Confirmations in Add Profile with Confirmation Code input |
| CR11012R01 | LPA Set\_EditDefaultSM-DP+Address\_User\_Intent\_Verification\_Removal |
| CR11015R01 | eUICC Test Enviorment for the eUICC |
| CR11017R01 | eUICC Mirror on integrated eUICCInfo2 |
| CR11018R01 | eUICC Introduction of v2.3 |
| NA | Fix ES2+ test enviorment |
| CR11019R02 | LPA\_Update\_DeleteProfile\_Strong\_Confirmation |
| CR11011R02 | Update\_eUICCMemoryReset\_Strong\_Confirmation |
| CR11016R01 | Optional support of Set/Edit Default SM-DP+ Address |
| CR11020R04 | Update\_for\_LPA45 |
| CR11005R03 | SM-DP+\_ES2+\_DownloadOrderWithRetry |
| CR11021R06 | Update\_eUICCMemoryReset |
| CR11027R00 | Editorial\_Definitions\_Abbreviations\_Reference\_SGP.22 |
| CR11024R04 | SM-DP+ ES2+\_ConfirmOrder 1-8 |
| CR11026R00 | SM-DP+ ES2+ Test Environment with SM-DS |
| CR11030R03 | SM-DP+ ES2+ ConfirmOrder Errors 1-6 |
| CR11031R02 | SM-DP+ ES2+ ConfirmOrderRetry1-4 |
| CR11032R00 | SM-DP+ Entity SM-DP+ |
| CR11028R04 | SM-DP+\_ES2+\_CancelOrder-1-3 |
| CR11029R03 | SM-DP+\_ES2+\_CancelOrderErrors-1-5 |
| CR11033R01 | Update reference to SGP.26 |
| NA | To include the missing part of CR11033R01 |
| CR11023R06 | Update of applicability of PPR sequences |
| CR11022R06 | CancelSession: remove unnecessary usage of profile already installed with PPR1 |
| CR11034R02 | eUICC\_Clarify\_EuiccInfo2\_TRE |
| CR11035R01 | Operator TLS Certificate |
| CR11036R00 | Updates and corrections related to SGP.22 versions |
| CR11037R01 | SIMAlliance references |
| CR11038R01 | LPA fixes |
| CR11039R01 | Optional\_support\_of\_Brainpool\_and\_FRP\_for\_TLS |
| CR11040R01 | DpProprietaryData\_Additional\_Data\_Must\_be\_allowed |
| CR11041R00 | Further SIMAlliance clean up |
| CR11042R02 | eUICCMemoryReset\_fixes |
| CR11043R04 | eUICC Profile Package versions |
| SGP.23 v1.11 | 28 October 2021 | CR11101R02 | Fix\_ASN1\_field\_name\_and\_remove\_restriction | ISAG | Yolanda Sanz, GSMA |
| CR11103R01 | Update\_AddProfile\_with\_an\_enabled\_PPR1\_Profile |
| CR11104R03 | Reference\_TCA\_Test\_Spec\_v3.1. |
| CR11105R00 | Remove\_SGP22v2.1\_References |
| CR11106R00 | Update\_C005 |
| CR11107R02 | Clarify higher major version in additionalEuiccProfilePackageVersions |
| CR11108r03 | RSPCapability bit for Profile Metadata Extensibilty |
| CR11109R01 | Case-insensitivity in Content-Type |
| CR11111R01 | Apple\_Extended euiccInfo2 - Alt |
| CR11112R00 | Apple\_Typo in extended deviceInfo |
| CR11113R01 | Vodafone\_SM-DP+\_ES2+\_Fixes |
| CR11114R00 | SGP.22 v2.4 |
| CR11115R01 | SM-DP+\_ES2+\_Fixes-2 |
| CR11116R02 | eUICC\_support of vendor-specific metadata |
| SPG.223 v1.12 | 28 March 2022 | CR11201R01 | SMDP FQDN | ISAG | Yolanda Sanz, GSMA |
| CR11202R00 | EUICC\_IUT setting for eUICC Profile Package version |
| CR11203R00 | LPA Adding conditions for SM-DS test cases |
| CR11204R02\_ | LPA\_LPM\_on\_disabled\_profile |
| CR11205R02 | LPA\_CancelSession\_with\_PPRs |
| CR11206R00\_ | LPA\_Avoiding\_state\_verification |
| CR11207R00 | eUICC\_TCA\_eUICC\_Test\_Specification\_Version\_Update |
| CR11208R02 | LPA\_Conditions\_for\_EnabledDisabled\_profile |
| CR11209R01 | LPA\_Adding\_simple\_delete\_profile |
| CR11210R02 | LPA\_EnableDisableProfile |
| CR11211R01\_ | LPA\_EnableProfile\_additional\_simplified\_TS |
| CR11212R01 | LPA\_Adding conditions for Default SM-DP+ test cases |
| CR11214R02 | LPA\_HandleNotification\_additionalTS |
| CR11215R00 | LPA\_Clarification\_on\_disabled\_state |
| NA | To fix some correction after CR implementation |  |  |
| NA | To fix some correction after CR implementation |  |  |
| SGP.23 v1.13 | 19 October 2022 | CR11303R00 | LPAd Clarification of TLS Alerts | ISAG | Yolanda Sanz /GSMA |
| CR11304r02 | LPAd Helper Profile Conditions |
| CR11305R00 | LPAd Helper Profile – deletion after initial conditions |
| CR11306R00 | LPAd Helper Profile Enable Profile Applicability |
| CR11307R00 | LPAd Helper profiles – removing notifications |
| CR11308R02 | LPAd - eUICC Memory Reset Specify SM-DP+ |
| CR11309R00 | eUICC\_Test\_Specification\_Version\_Update |
| CR11310R03 | eUICC\_TRE\_Properties\_GetEuiccInfo2\_for\_IntegratedEuicc |
| SGP.23 v1.14 | 19 October 2022 | NA | First Draf | ISAG | Yolanda Sanz/GSMA |
| SGP.23 v1.14  2 | 05 July 2023 | CR11401R00 | LPA Ensure Profile content and Profile Metadata consistent | ISAG | Yolanda Sanz / GSMA |
| CR11402R00 | eUICC\_TCA\_eUICC\_Test\_Specification\_Version\_Update |
| CR11403R01 | eUICC\_Reduction of svn checking to two test cases |
| CR11405R02 | eUICC\_LPA\_Server Support SGP.22 v2.5 |
| CR11406R02 | eUICC\_Clarification on eUICC Memory Reset |
| CR11407R02 | eUICC\_AddtionalPPVersion |
| CR11408R02 | eUICC\_Add TCA v3.3 |
| CR11410R01 | SM-DP+\_ICCID |
| NA | Final version for eSIMG approval | ISAG | Yolana Sanz/  GSMA |

J.2 Other Information

|  |  |
| --- | --- |
| Type | Description |
| Document Owner | eSIMG |
| Editor / Company | Yolanda Sanz, GSMA |

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at [prd@gsma.com](mailto:prd@gsma.com).

Your comments or suggestions & questions are always welcome.