



## **RSP Test Specification**

### **Version 1.6**

### **12 October 2019**

*This is a Non-binding Permanent Reference Document of the GSMA*

---

#### **Security Classification: Non-confidential**

Access to and distribution of this document is restricted to the persons permitted by the security classification. This document is confidential to the Association and 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 © 2019 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.

#### **Antitrust Notice**

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

## Table of Contents

	<b>Introduction</b>	<b>7</b>
	1.1 Overview	7
	1.2 Scope	7
1	1.3 Definition of Terms	8
	1.4 Abbreviations	12
	1.5 Document Cross-references	14
	1.6 Conventions	16
<b>2</b>	<b>Testing Rules</b>	<b>16</b>
	2.1 Applicability	16
	2.1.1 Format of the Optional Features Table	16
	2.1.2 Format of the Applicability Table	16
	2.1.3 Applicability and Notations	16
	2.1.4 Optional Features Table	17
	2.1.5 Applicability Table	18
	2.2 General Consideration	27
	2.2.1 Test Case Definition	27
	2.2.2 Test Cases Format	27
	2.2.3 General Rules for eUICC Testing	32
	2.2.4 General Rules for Device Testing	34
	2.2.5 Pass Criteria	35
	2.2.6 Future Study	35
<b>3</b>	<b>Testing Architecture</b>	<b>36</b>
	3.1 Testing Scope	36
	3.2 Testing Execution	37
	3.2.1 eUICC - Test Environment	38
	3.2.2 SM-DP+ and SM-DS - Test Environment	38
	3.2.3 Device/LPA - Test Environment	40
	3.2.4 End-to-End Testing	42
<b>4</b>	<b>Interface Compliance Testing</b>	<b>42</b>
	4.1 General Overview	42
	4.2 eUICC Interfaces	42
	4.2.1 ATR and ISD-R Selection	42
	4.2.2 ES6 (Operator -- eUICC): UpdateMetadata	43
	4.2.3 ES8+ (SM-DP+ -- eUICC): InitialiseSecureChannel	51
	4.2.4 ES8+ (SM-DP+ -- eUICC): ConfigureISDP	55
	4.2.5 ES8+ (SM-DP+ -- eUICC): StoreMetadata	57
	4.2.6 ES8+ (SM-DP+ -- eUICC): ReplaceSessionKeys	68
	4.2.7 ES8+ (SM-DP+ -- eUICC): LoadProfileElements	70
	4.2.8 ES10a (LPA -- eUICC): GetEuiccConfiguredAddresses	79
	4.2.9 ES10a (LPA -- eUICC): SetDefaultDpAddress	80
	4.2.10 ES10b (LPA -- eUICC): PrepareDownload	82
	4.2.11 ES10b (LPA -- eUICC): LoadBoundProfilePackage	89
	4.2.12 ES10b (LPA -- eUICC): GetEUICCChallenge	99

4.2.13	ES10b (LPA -- eUICC): GetEUICCInfo	100
4.2.14	ES10b (LPA -- eUICC): ListNotification	104
4.2.15	ES10b (LPA -- eUICC): RetrieveNotificationsList	118
4.2.16	ES10b (LPA -- eUICC): RemoveNotificationFromList	146
4.2.17	ES10b (LPA -- eUICC): LoadCRL	152
4.2.18	ES10b (LPA -- eUICC): AuthenticateServer	152
4.2.19	ES10b (LPA -- eUICC): CancelSession	174
4.2.20	ES10c (LPA -- eUICC): GetProfilesInfo	182
4.2.21	ES10c (LPA -- eUICC): EnableProfile	190
4.2.22	ES10c (LPA -- eUICC): DisableProfile	213
4.2.23	ES10c (LPA -- eUICC): DeleteProfile	235
4.2.24	ES10c (LPA -- eUICC): eUICCMemoryReset	246
4.2.25	ES10c (LPA -- eUICC): GetEID	251
4.2.26	ES10c (LPA -- eUICC): SetNickname	252
4.2.27	ES10b (LPA -- eUICC): GetRAT	256
4.3	SM-DP+ Interfaces	258
4.3.1	ES2+ (Operator -- SM-DP+): DownloadOrder	258
4.3.2	ES2+ (Operator -- SM-DP+): ConfirmOrder	258
4.3.3	ES2+ (Operator -- SM-DP+): CancelOrder	258
4.3.4	ES2+ (Operator -- SM-DP+): ReleaseProfile	258
4.3.5	ES2+ (Operator -- SM-DP+): HandleDownloadProgressInfo	258
4.3.6	ES2+ (Operator -- SM-DP+): TLS, Mutual Authentication, Server, Session Establishment	258
4.3.7	ES8+ (SM-DP+ -- eUICC): InitialiseSecureChannel	258
4.3.8	ES8+ (SM-DP+ -- eUICC): ConfigureISDP	258
4.3.9	ES8+ (SM-DP+ -- eUICC): StoreMetadata	259
4.3.10	ES8+ (SM-DP+ -- eUICC): ReplaceSessionKeys	259
4.3.11	ES8+ (SM-DP+ -- eUICC): LoadProfileElements	259
4.3.12	ES9+ (LPA -- SM-DP+): InitiateAuthentication	259
4.3.13	ES9+ (LPA -- SM-DP+): GetBoundProfilePackage	265
4.3.14	ES9+ (LPA -- SM-DP+): AuthenticateClient	303
4.3.15	ES9+ (LPA -- SM-DP+): HandleNotification	336
4.3.16	ES9+ (LPA -- SM-DP+): CancelSession	353
4.3.17	ES9+ (LPA -- SM-DP+): TLS, Server Authentication, Session Establishment	374
4.3.18	ES12 (SM-DP+ -- SM-DS): RegisterEvent	374
4.3.19	ES12 (SM-DP+ -- SM-DS): DeleteEvent	374
4.3.20	ES12 (SM-DP+ -- SM-DS): TLS, Mutual Authentication, Client, Session Establishment	374
4.4	LPAAd Interfaces	375
4.4.1	ES10a (LPA -- eUICC): GetEuiccConfiguredAddresses	375
4.4.2	ES10a (LPA -- eUICC): SetDefaultDpAddress	375
4.4.3	ES10b (LPA -- eUICC): PrepareDownload	375
4.4.4	ES10b (LPA -- eUICC): LoadBoundProfilePackage	375
4.4.5	ES10b (LPA -- eUICC): GetEUICCChallenge	375

4.4.6	ES10b (LPA -- eUICC): GetEUICCInfo	375
4.4.7	ES10b (LPA -- eUICC): ListNotification	375
4.4.8	ES10b (LPA -- eUICC): RetrieveNotificationsList	375
4.4.9	ES10b (LPA -- eUICC): RemoveNotificationFromList	375
4.4.10	ES10b (LPA -- eUICC): LoadCRL	375
4.4.11	ES10b (LPA -- eUICC): AuthenticateServer	375
4.4.12	ES10b (LPA -- eUICC): CancelSession	375
4.4.13	ES10c (LPA -- eUICC): GetProfilesInfo	375
4.4.14	ES10c (LPA -- eUICC): EnableProfile	376
4.4.15	ES10c (LPA -- eUICC): DisableProfile	376
4.4.16	ES10c (LPA -- eUICC): DeleteProfile	376
4.4.17	ES10c (LPA -- eUICC): eUICCMemoryReset	376
4.4.18	ES10c (LPA -- eUICC): GetEID	376
4.4.19	ES10c (LPA -- eUICC): SetNickname	376
4.4.20	ES10b (LPA -- eUICC): GetRAT	376
4.4.21	ES9+ (LPA -- SM-DP+): InitiateAuthentication	376
4.4.22	ES9+ (LPA -- SM-DP+): GetBoundProfilePackage	381
4.4.23	ES9+ (LPA -- SM-DP+): AuthenticateClient	390
4.4.24	ES9+ (LPA -- SM-DP+): HandleNotification	403
4.4.25	ES9+ (LPA -- SM-DP+): CancelSession	412
4.4.26	ES9+ (LPA -- SM-DP+): HTTPS	426
4.4.27	ES11 (LPA -- SM-DS): InitiateAuthentication	431
4.4.28	ES11 (LPA -- SM-DS): AuthenticateClient	435
4.4.29	ES11 (LPA -- SM-DS): HTTPS	443
4.5	SM-DS Interfaces	448
4.5.1	ES12 (SM-DP+ -- SM-DS): RegisterEvent	448
4.5.2	ES12 (SM-DS -- SM-DP+): DeleteEvent	457
4.5.3	ES15 (SM-DS -- SM-DS): RegisterEvent	465
4.5.4	ES15 (SM-DS -- SM-DS): DeleteEvent	467
4.5.5	ES11 (LPA -- SM-DS): InitiateAuthentication	469
4.5.6	ES11 (LPA -- SM-DS): Authenticate Client	470
4.5.7	ES15 (SM-DS -- SM-DS): TLS, Mutual Authentication, Client, Session Establishment	491
4.5.8	ES12 (SM-DS -- SM-DP+): TLS, Mutual Authentication, Server, Session Establishment	492
4.5.9	ES15 (SM-DS -- SM-DS): TLS, Mutual Authentication, Server, Session Establishment	492
4.5.10	ES11 (LPA -- SM-DS): TLS, Server Authentication, Session Establishment	493
4.6	TLS Interface	493
4.6.1	TLS, Mutual Authentication, Client, TLS Establishment	493
4.6.2	TLS, Mutual Authentication, Server, TLS Establishment	502
4.6.3	TLS, Server Authentication, TLS Establishment	510
4.7	LP Ae Interfaces	513
<b>5</b>	<b>Procedure - Behaviour Testing</b>	<b>514</b>

5.1	General Overview	514
5.2	eUICC Behaviour	514
5.2.1	Retry mechanism	514
5.2.2	Forbidden PPRs	520
5.2.3	eUICC's RAT	521
5.2.4	eUICC File Structure	523
5.2.5	eUICC Delete Profile Process	524
5.2.6	eUICC Enable Profile Process	525
5.2.7	eUICC Disable Profile Process	526
5.2.8	eUICC Notifications	528
5.3	Platform Procedures	529
5.3.1	Profile Download and Installation Procedure	529
5.3.2	Common Mutual Authentication Process	529
5.3.3	Profile Download and Installation Process	529
5.4	Device Procedures	532
5.4.1	Local Profile Management - Add Profile	532
5.4.2	Local Profile Management – ListProfiles	546
5.4.3	Local Profile Management - SetNickname	547
5.4.4	Local Profile Management - Delete Profile	550
5.4.5	Local Profile Management - Enable Profile	555
5.4.6	Local Profile Management - Disable Profile	559
5.4.7	Local eUICC Management - Retrieve EID Process	562
5.4.8	Local eUICC Management - eUICC Memory Reset Process	563
5.4.9	Local eUICC Management - eUICC Test Memory Reset Process	567
5.4.10	Local eUICC Management – Set/Edit Default SM-DP+ Address Process	568
5.4.11	Device Power On – Profile Discovery	570
6	<b>End-to-End Testing</b>	<b>573</b>
7	<b>External Test Specifications</b>	<b>573</b>
7.1	SIMAlliance eUICC Profile Package Test Specification	573
<b>Annex A</b>	<b>Constants</b>	<b>574</b>
A.1	Generic Constants	574
A.2	Test Certificates and Test Keys	583
<b>Annex B</b>	<b>Dynamic Content</b>	<b>592</b>
<b>Annex C</b>	<b>Methods And Procedures</b>	<b>602</b>
C.1	Methods	602
C.2	Procedures	615
<b>Annex D</b>	<b>Commands And Responses</b>	<b>633</b>
D.1	ES8+ Requests And Responses	633
D.1.1	ES8+ Requests	633
D.2	ES9+ Requests And Responses	647
D.2.1	ES9+ Requests	647
D.2.2	ES9+ Responses	675
D.3	ES10x Requests And Responses	689
D.3.1	ES10x Requests	689

D.3.2	ES10x Responses	700
D.4	APDU	725
D.4.1	APDU Commands	725
D.4.2	R-APDU Chaining	726
D.5	ES6 Requests And Responses	726
D.5.1	ES6 Requests	726
D.6	ES11 Requests And Responses	727
D.6.1	ES11 Requests	727
D.6.2	ES11 Responses	734
D.7	ES12 Requests And Responses	735
D.8	ES15 Requests And Responses	735
D.9	Common Server Responses	735
<b>Annex E</b>	<b>Profiles</b>	<b>744</b>
<b>Annex F</b>	<b>IUT Settings</b>	<b>749</b>
F.1	eUICC Settings	749
F.2	Platforms Settings	749
F.3	Device Settings	750
F.4	Common Settings	751
<b>Annex G</b>	<b>Initial States</b>	<b>752</b>
G.1	Device	752
G.1.1	Device (default)	752
G.1.2	Companion Device connected to a Primary Device	752
G.1.3	Test eUICC Settings	752
G.2	eUICC	753
G.2.1	Common Initial States	753
G.2.2	For eUICC supporting NIST P-256	754
G.2.3	For eUICC supporting BrainpoolP256r1	754
G.2.4	With default RAT configuration	754
G.2.5	With Additional PPARs in the RAT	754
G.2.6	Clean-up procedure	755
G.3	SM-DP+ and SM-DS	755
<b>Annex H</b>	<b>Icons and QR Codes</b>	<b>756</b>
<b>Annex I</b>	<b>Requirements</b>	<b>757</b>
<b>Annex J</b>	<b>Document Management</b>	<b>758</b>
J.1	Document History	758
J.2	Other Information	758

## Introduction

### 1.1 Overview

- 1 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 LPA 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 V2.1 [2a]
- GSMA RSP Technical Specification V2.2 [2b]
- GSMA RSP Technical Specification V2.2.1 [2]

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

### 1.3 Definition of Terms

Term	Description
Activation Code	Information issued by an Operator/Service Provider to an End User. It is used by the End User to request the download and installation of a Profile.
Activation Code Token	A part of the Activation Code information provided by the Operator/Service Provider to reference a Subscription.
Alternative SM-DS	SM-DS used in cascade mode with a Root SM-DS to redirect Event Registration from a SM-DP+ to the Root SM-DS.
Authenticated Confirmation	A mechanism by which the End User confirms their action through a method involving the input of personalised information (e.g. PIN, fingerprint).
Bound Profile Package	A Protected Profile Package that has been cryptographically linked to a particular eUICC.
Certificate Authority	A Certificate Authority is an entity that issues digital certificates.
Companion Device	A Device that relies on the capabilities of a Primary Device for the purpose of Remote SIM Provisioning.
Confirmation Code	A code entered by an End User required by the SM-DP+ to confirm the download of a Profile.
Confirmation Code Required Flag	A parameter to indicate whether the Confirmation Code is required.
Device	User equipment used in conjunction with an eUICC to connect to a mobile network. E.g. a tablet, wearable, smartphone or handset.
Disabled (Profile)	The state of a Profile where all files and applications (e.g. NAA) present in the Profile are not selectable.
Enabled (Profile)	The state of a Profile when its files and/or applications (e.g., NAA) are selectable.
End User	The person using the Device.
eUICC	A removable or non-removable UICC which enables the remote and/or local management of Profiles in a secure way. NOTE: The term originates from "embedded UICC".
eUICC Certificate	A certificate issued by the EUM for a specific eUICC. This Certificate can be verified using the EUM Certificate.
eUICC Memory Reset	An action that returns the eUICC to a state equivalent to a factory state.
eUICC Test Memory Reset	An action that deletes all post-issuance Test Profiles on an eUICC.
EUM Certificate	A certificate issued by a GSMA CI to a GSMA accredited EUM which can be used to verify eUICC Certificates.
Event	A Profile download which is set by an SM-DP+ on behalf of an Operator, to be processed by a specific eUICC.
EventID	Unique identifier of an Event for a specific EID generated by the SM-DP+ / SM-DS.



Term	Description
Event Record	The set of information stored on the SM-DS for a specific Event, via the Event Registration procedure. This information consists of either: <ul style="list-style-type: none"> <li>• the Event-ID, EID, and SM-DP+ address or</li> <li>• the Event-ID, EID, and SM-DS address.</li> </ul>
Event Registration	A process notifying the SM-DS on the availability of information on either a specific SM-DP+ or a specific SM-DS for a specific eUICC.
GSMA Certificate Issuer	A Certificate Authority accredited by GSMA.
Integrated Circuit Card ID	Unique number to identify a Profile in an eUICC as defined by ITU-T E.118 [10].
International Mobile Subscriber Identity	Unique identifier owned and issued by Mobile Operators as defined in 3GPP TS 23.003 [12] section 2.2.
Issuer Identifier Number	The first 8 digits of the EID identifying the EUM issuing the eUICC.
Issuer Security Domain	A security domain on the UICC as defined by GlobalPlatform Card Specification [6].
Local Profile Assistant	A functional element in the Device or in the eUICC that provides the Local Profile Download (LPD), Local Discovery Services (LDS) and Local User Interface (LUI) features. When the LPA is located in the Device, they are called LPA <sub>d</sub> , LPD <sub>d</sub> , LUI <sub>d</sub> , LDS <sub>d</sub> . When the LPA is located in the eUICC, they are called LPA <sub>e</sub> , LPD <sub>e</sub> , LUI <sub>e</sub> , LDS <sub>e</sub> . Where LPA, LPD, LDS or LUI are used, they apply to the element independent of its location in the Device or in the eUICC.
Local Profile Management	Local Profile Management are operations that are locally initiated on the End User (ESeu) interface.
Local Profile Management Operation	Local Profile Management Operations include enable Profile, Disable Profile, Delete Profile, query Profile Metadata, eUICC Memory Reset, eUICC Test Memory Reset, set/edit Nickname, add Profile and edit default SM-DP+ address.
MatchingID	Reference data for an RSP Server which could be an Activation Code Token or the EventID.
Mobile Network Operator	An entity providing access capability and communication services to its End User through a mobile network infrastructure.
Mobile Network Operator Security Domain (MNO-SD)	Part of the Profile, owned by the Operator, providing the Secured Channel to the Operator's Over The Air (OTA) Platform. It is used to manage the content of a Profile once the Profile is enabled.
Network Access Application	Application residing in a Profile providing authorisation to access a network.
Notification	A report about a Profile installation or Local Profile Management Operation processed by the eUICC.
Operational Profile	A combination of Operator data and applications to be provisioned on an eUICC for the purposes of providing services by the Operator. The Profile SHALL be in support of a Subscription with the relevant Operator and allow connectivity to a mobile network. Applications MAY be included to provide non-telecommunication services.

Term	Description
Operator	A Mobile Network Operator or Mobile Virtual Network Operator; a company providing wireless cellular network services.
Other Notification	Any Notification other than a Profile Installation Result.
Primary Device	A Device that can be used to provide some capabilities to a Companion Device for the purpose of Remote SIM Provisioning.
Profile	A combination of data and applications to be provisioned on an eUICC for the purpose of providing services.
Profile Component	A Profile Component is an element of the Profile, when installed in the eUICC, and MAY be one of the following: <ul style="list-style-type: none"> <li>• An element of the file system like an MF, EF or DF;</li> <li>• An Application, including NAA and Security Domain;</li> <li>• Profile Metadata, including Profile Policy Rules;</li> <li>• An MNO-SD.</li> </ul>
Profile Installation Result	A Notification that contains the result of a Profile installation.
Profile Management	A combination of local and remote management operations (e.g.: enable Profile, disable Profile, delete Profile, and query Profile Metadata).
Profile Management Operation	An operation related to the content and state update of a Profile in a dedicated ISD-P on the eUICC.
Profile Metadata	Information pertaining to a Profile used for the purpose of Local Profile Management.
Profile Nickname	Alternative name of the Profile set by the End User.
Profile Owner	The entity that controls the operations that can be performed upon its Profile. With the exception of Test Profiles, this is always the Operator.
Profile Package	A personalised Profile using an interoperable description format that is transmitted to an eUICC to load and install a Profile.
Profile Policy Authorisation Rule	A set of data that governs the ability of a Profile Owner to make use of a Profile Policy Rule in a Profile.
Profile Policy Rule	Defines a qualification for or enforcement of an action to be performed on a Profile when a certain condition occurs.
Protected Profile Package	A Profile Package which has been cryptographically protected for storage but not linked to a particular eUICC.
Provisioning Profile	A combination of Operator data and applications to be provisioned on an eUICC for the purposes of providing connectivity to a mobile network solely for the purpose of the provisioning of Profiles on the eUICC.  NOTE: Use of Provisioning Profiles for other system services in version 3 of this specification MAY require modifications of this definition.
Remote SIM Provisioning	The downloading, installing, enabling, disabling, and deleting of a Profile on an eUICC.
Roles	Roles are representing a logical grouping of functions.

Term	Description
Root SM-DS	A globally identified central access point for finding Events from one or more SM-DP+(s).
Rules Authorisation Table	A set of Profile Policy Authorisation Rules that, together, determines the ability of a Profile Owner to make use of a set of Profile Policy Rules in a Profile.
RSP Server	Either an SM-DS or SM-DP+.
Service Provider	The organization through which the End User obtains PLMN telecommunication services. This is usually the network Operator or possibly a separate body.
Simple Confirmation	A mechanism by which the End User confirms their action, e.g. by selecting Yes/No, OK/Cancel.
SM-DP+ Certificate	A Certificate issued by a GSMA CI to a GSMA accredited SM-DP+.
SM-DS Certificate	A Certificate used by a GSMA CI to a GSMA accredited SM-DS.
SM-DP+ OID	Identifier of the SM-DP+ that is globally unique and is included as part of the SM-DP+ Certificate.
SM-DS OID	Identifier of the SM-DS that is globally unique and is included as part of the SM-DS Certificate.
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.
Subscription	Describes the commercial relationship between the End User and the Service Provider.
Subscription Manager Data Preparation+ (SM-DP+)	<p>This role prepares Profile Packages, secures them with a Profile protection key, stores Profile protection keys in a secure manner and the Protected Profile Packages in a Profile Package repository, and allocates the Protected Profile Packages to specified EIDs.</p> <p>The SM-DP+ binds Protected Profile Packages to the respective EID and securely downloads these Bound Profile Packages to the LPA of the respective eUICC.</p>
Subscription Manager Discovery Server (SM-DS)	This is responsible for providing addresses of one or more SM-DP+(s) to a LDS.
Test Plan	Current document describing the test cases that allow the RSP ecosystem to be tested.
Test Profile	A combination of data and applications to be provisioned on an eUICC to provide connectivity to test equipment for the purpose of testing the Device and the eUICC. A Test Profile is not intended to store any Operator Credentials.
User Intent	Describes the direct, real time acquisition and validation of the manual End User instruction on the LUI to trigger locally a Profile download or Profile Management Operation. As defined in SGP.21 [3].

## 1.4 Abbreviations

Abbreviation	Description
AID	Application Identifier
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One
ATR	Answer To Reset
BPP	Bound Profile Package
C-APDU	Command APDU
CASD	Controlling Authority Security Domain
CERT.CI.ECDSA	Certificate of the CI for its Public ECDSA Key
CERT.DPauth.ECDSA	Certificate of the SM-DP+ for its Public ECDSA key used for SM-DP+ authentication
CERT.DPpb.ECDSA	Certificate of the SM-DP+ for its Public ECDSA key used for Profile Package Binding
CERT.DSauth.ECDSA	Certificate of the SM-DS for its Public ECDSA key used for SM-DS authentication
CERT.EUICC.ECDSA	Certificate of the eUICC for its Public ECDSA key
CERT.EUM.ECDSA	Certificate of the EUM for its Public ECDSA key
CERT.DP.TLS	Certificate of the SM-DP+ for securing TLS
CERT.DS.TLS	Certificate of the SM-DS for securing TLS
CI	Certificate Issuer
CRL	Certificate Revocation List
CRT	Control Reference Template
DER TLV	Distinguished Encoding Rules - Tag Length Value
DH	Diffie-Hellman
ECASD	eUICC Controlling Authority Security Domain
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve cryptography Digital Signature Algorithm
ECKA	Elliptic Curve cryptography Key Agreement algorithm
EID	eUICC-ID as defined in SGP.02 [1]
ETSI	European Telecommunications Standards Institute
EUM	eUICC Manufacturer
FCP	File Control Parameters
FFS	For Future Study
FQDN	Fully Qualified Domain Name
GID1	Group Identifier 1, as defined in 3GPP TS 31.102 [18]
GID2	Group Identifier 2, as defined in 3GPP TS 31.102 [18]
GSMA	GSM Association
HW	Hardware
ICCID	Integrated Circuit Card ID

Abbreviation	Description
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
ISD	Issuer Security Domain
ISD-P	Issuer Security Domain Profile
ISD-R	Issuer Security Domain Root
ISO	International Standards Organisation
ITU	International Telecommunications Union
IUT	Implementation Under Test
KVN	Key Version Number
LDS	Local Discovery Service
LDSd	Local Discovery Service when LPA is in the Device
LDS <sub>e</sub>	Local Discovery Service when LPA is in the eUICC
LPA	Local Profile Assistant
LPA <sub>d</sub>	Local Profile Assistant when LPA is in the Device
LPA <sub>e</sub>	Local Profile Assistant when LPA is in the eUICC
LPD	Local Profile Download
LPD <sub>d</sub>	Local Profile Download when LPA is in the Device
LPD <sub>e</sub>	Local Profile Download when LPA is in the eUICC
LTE	Long Term Evolution
LUI	Local User Interface
LUI <sub>d</sub>	Local User Interface when LPA is in the Device
LUI <sub>e</sub>	Local User Interface when LPA is in the eUICC
MAC	Message Authentication Code
MNO	Mobile Network Operator
MOC	Mandatory, Optional or Conditional
NAA	Network Access Application
OCE	Off-Card Entity
OTA	Over The Air
OS	Operating System
otPK.EUICC.ECKA	One-time Public Key of the eUICC for ECKA
otSK.EUICC.ECKA	One-time Private Key of the eUICC for ECKA
PE	Profile Element
PKI	Public Key Infrastructure
PIR	Profile Installation Result
POR	Proof Of Receipt
PPAR	Profile Policy Authorisation Rule
PPK-ENC	Profile Protection Key for message encryption/decryption
PPK-MAC	Profile Protection Key for message MAC generation/verification

Abbreviation	Description
PPP	Protected Profile Package
PPR	Profile Policy Rule
R-APDU	Response APDU
RAT	Rules Authorisation Table
RSA	Rivest / Shamir / Adleman asymmetric algorithm
RSP	Remote SIM Provisioning
SAS	Security Accreditation Scheme
SCP	Secure Channel Protocol
SD	Security Domain
S-ENC	Session key for message encryption/decryption
S-MAC	Session Key for message MAC generation/verification
SK.CI.ECDSA	Private key of the CI for signing certificates
SK.DPauth.ECDSA	Private Key of the of SM-DP+ for creating signatures for SM-DP+ authentication
SK.EUICC.ECDSA	Private key of the eUICC for creating signatures
SK.EUM.ECDSA	Private key of the EUM for creating signatures
SM-DP+	Subscription Manager Data Preparation (Enhanced compared to the SM-DP in SGP.02 [1])
SP	Service Provider
SSD	Supplemental Security Domain
SVN	SGP.22 Specification Version Number (referred to as 'eSVN' in SGP.21 [3]).
TAC	Type Allocation Code
TAR	Toolkit Application Reference
TLS	Transport Layer Security
UPP	Unprotected Profile Package
URI	Uniform Resource Identifier
URL	Uniform Resource locator
USIM	Universal Subscriber Identity Module

## 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 V2.2.x (x≥1)
[2a]	SGP.22	RSP Technical Specification V2.1
[2b]	SGP.22	RSP Technical Specification V2.2
[3]	SGP.21	RSP Architecture V2.2
[3a]	SGP.21	RSP Architecture V2.1

Ref	Document Number	Title
[4]	SIMalliance	SIMalliance eUICC Profile Package: Interoperable Format Technical Specification V2.0 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 <a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a>
[21]	SGP.11	Remote Provisioning Architecture for Embedded UICC Test Specification V3.2
[22]	3GPP TS 23.040	Technical realization of the Short Message Service (SMS)
[23]	SIMalliance Test	SIMalliance eUICC Profile Package: Interoperable Format Test Specification Version 2.2.2
[24]	RFC 4492	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)
[25]	SGP.26	RSP Test Certificates Definition v1.2

Ref	Document Number	Title
[26]	3GPP TS 29.002	Mobile Application Part (MAP) specification

## 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].

## 2 Testing Rules

### 2.1 Applicability

#### 2.1.1 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**

#### 2.1.2 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, LPA <sub>d</sub> , LPA <sub>e</sub> or eUICC Entities under test that take in charge the functions used in the test case.
Version	This column indicates which test cases are applicable for the given SGP.22 version. 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**

#### 2.1.3 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.



Applicability code	Meaning
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**

### 2.1.4 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
Device Options	Mnemonic
The Device supports LPA <sub>d</sub>	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 and displaying the profile's Nickname	O_D_NICKNAME
The Device supports additional verification of TLS certificate content (i.e. key usage, extended key usage and certificate policy)	O_D_TLS_FULL_VERIFICATION
<b>SM-DP+ Options</b>	<b>Mnemonic</b>
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
<b>SM-DS Options</b>	<b>Mnemonic</b>
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
<p>NOTE 1: Devices which supports O_D_ONLY_CELLULAR_CONNECTIVITY are out of scope of the current version of this document.</p> <p>NOTE 2: For this version of test specification:</p> <ul style="list-style-type: none"> <li>• O_E_FRP is not applicable</li> <li>• The eUICC SHALL support either O_E_NIST or O_E_BRP or both</li> </ul>	

**Table 4: Options**

### 2.1.5 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	Roles	V2.1	V2.2.X (X≥0)	Test Env.
eUICC Interfaces Compliance Testing					
4.2.1.2.1	TC_eUICC_ATR_And_ISDR_Selection	eUICC	C006	C006	TE_eUICC
4.2.2.2.1	TC_eUICC_ES6.UpdateMetadata	eUICC	M	M	TE_eUICC
4.2.3.2.1	TC_eUICC_ES8+.InitialiseSecureChannel	eUICC	M	M	TE_eUICC

Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
4.2.4.2.1	TC_eUICC_ES8+.ConfigureISDP	eUICC	M	M	TE_eUICC
4.2.5.2.1	TC_eUICC_ES8+.StoreMetadata	eUICC	M	M	TE_eUICC
4.2.6.2.1	TC_eUICC_ES8+.ReplaceSessionKeys	eUICC	M	M	TE_eUICC
4.2.7.2.1	TC_eUICC_ES8+.LoadProfileElements	eUICC	M	M	TE_eUICC
4.2.8.2.1	TC_eUICC_ES10a.GetEuiccConfiguredAddresses	eUICC	M	M	TE_eUICC
4.2.9.2.1	TC_eUICC_ES10a.SetDefaultDpAddress	eUICC	M	M	TE_eUICC
4.2.10.2.1	TC_eUICC_ES10b.PrepareDownloadNIST	eUICC	C001	C001	TE_eUICC
4.2.10.2.2	TC_eUICC_ES10b.PrepareDownloadBRP	eUICC	C002	C002	TE_eUICC
4.2.10.2.3	TC_eUICC_ES10b.PrepareDownloadFRP	eUICC	C003	C003	TE_eUICC
4.2.10.2.4	TC_eUICC_ES10b.PrepareDownloadErrorCases	eUICC	M	M	TE_eUICC
4.2.11.2.1	TC_eUICC_ES10b.LoadBoundProfilePackageNIST	eUICC	C001	C001	TE_eUICC
4.2.11.2.2	TC_eUICC_ES10b.LoadBoundProfilePackageBRP	eUICC	C002	C002	TE_eUICC
4.2.11.2.3	TC_eUICC_ES10b.LoadBoundProfilePackageFRP	eUICC	C003	C003	TE_eUICC
4.2.11.2.4	TC_eUICC_ES10b.LoadBoundProfilePackage_ErrorCases	eUICC	M	M	TE_eUICC
4.2.12.2.1	TC_eUICC_ES10b.GetEUICCChallenge	eUICC	M	M	TE_eUICC
4.2.13.2.1	TC_eUICC_ES10b.GetEUICCInfo1	eUICC	M	M	TE_eUICC
4.2.13.2.2	TC_eUICC_ES10b.GetEUICCInfo2_RSP_V2.1	eUICC	M	N/A	TE_eUICC
4.2.13.2.3	TC_eUICC_ES10b.GetEUICCInfo2_RSP_V2.2.x	eUICC	N/A	M	TE_eUICC
4.2.13.2.4	TC_eUICC_ES10b.GetEUICCInfo2	eUICC	M	M	TE_eUICC
4.2.14.2.1	TC_eUICC_ES10b.ListNotification All test sequences except the sequence #5	eUICC	M	M	TE_eUICC
4.2.14.2.1	TC_eUICC_ES10b.ListNotification Only the test sequence #5	eUICC	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	TE_eUICC
4.2.15.2.1	TC_eUICC_ES10b.RetrieveNotificationsList Only the test sequences #5 and #15	eUICC	C025	C025	TE_eUICC
4.2.16.2.1	TC_eUICC_ES10b.RemoveNotificationFromList All test sequences except the sequence #5	eUICC	M	M	TE_eUICC
4.2.16.2.1	TC_eUICC_ES10b.RemoveNotificationFromList Only the test sequence #5	eUICC	C025	C025	TE_eUICC
4.2.18.2.1	TC_eUICC_ES10b.AuthenticateServer_SM-DP+_NIST	eUICC	C001	C001	TE_eUICC
4.2.18.2.2	TC_eUICC_ES10b.AuthenticateServer_SM-	eUICC	C002	C002	TE_eUICC

Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
	DP+_BRP				
4.2.18.2.3	TC_eUICC_ES10b.AuthenticateServer_SM-DP+_FRP	eUICC	C003	C003	TE_eUICC
4.2.18.2.4	TC_eUICC_ES10b.AuthenticateServer_SM-DP+_ErrorCases	eUICC	M	M	TE_eUICC
4.2.18.2.5	TC_eUICC_ES10b.AuthenticateServer_SM-DS_BRP	eUICC	C002	C002	TE_eUICC
4.2.18.2.6	TC_eUICC_ES10b.AuthenticateServer_SM-DS_NIST	eUICC	C001	C001	TE_eUICC
4.2.18.2.7	TC_eUICC_ES10b.AuthenticateServer_SM-DS_FRP	eUICC	C003	C003	TE_eUICC
4.2.18.2.8	TC_eUICC_ES10b.AuthenticateServer_SM-DS_ErrorCases	eUICC	M	M	TE_eUICC
4.2.19.2.1	TC_eUICC_ES10b.CancelSessionNIST	eUICC	C001	C001	TE_eUICC
4.2.19.2.2	TC_eUICC_ES10b.CancelSessionBRP	eUICC	C002	C002	TE_eUICC
4.2.19.2.3	TC_eUICC_ES10b.CancelSessionFRP	eUICC	C003	C003	TE_eUICC
4.2.19.2.4	TC_eUICC_ES10b.CancelSession_ErrorCase	eUICC	M	M	TE_eUICC
4.2.20.2.1	TC_eUICC_ES10c.GetProfilesInfo	eUICC	M	M	TE_eUICC
4.2.21.2.1	TC_eUICC_ES10c.EnableProfile_Case3	eUICC	M	M	TE_eUICC
4.2.21.2.2	TC_eUICC_ES10c.EnableProfile_ErrorCases_Case3	eUICC	M	M	TE_eUICC
4.2.21.2.3	TC_eUICC_ES10c.EnableProfile_Case4	eUICC	M	M	TE_eUICC
4.2.21.2.4	TC_eUICC_ES10c.EnableProfile_ErrorCases_Case4	eUICC	M	M	TE_eUICC
4.2.22.2.1	TC_eUICC_ES10c.DisableProfile_Case3	eUICC	M	M	TE_eUICC
4.2.22.2.2	TC_eUICC_ES10c.DisableProfile_ErrorCases_Case3	eUICC	M	M	TE_eUICC
4.2.22.2.3	TC_eUICC_ES10c.DisableProfile_Case4	eUICC	M	M	TE_eUICC
4.2.22.2.4	TC_eUICC_ES10c.DisableProfile_ErrorCases_Case4	eUICC	M	M	TE_eUICC
4.2.23.2.1	TC_eUICC_ES10c.DeleteProfile_Case3	eUICC	M	M	TE_eUICC
4.2.23.2.2	TC_eUICC_ES10c.DeleteProfile_ErrorCases_Case3	eUICC	M	M	TE_eUICC
4.2.23.2.3	TC_eUICC_ES10c.DeleteProfile_Case4	eUICC	M	M	TE_eUICC
4.2.23.2.4	TC_eUICC_ES10c.DeleteProfile_ErrorCases_Case4	eUICC	M	M	TE_eUICC
4.2.24.2.1	TC_eUICC_ES10c.eUICCMemoryReset	eUICC	M	M	TE_eUICC
4.2.24.2.2	TC_eUICC_ES10c.eUICCMemoryReset_ErrorCases	eUICC	M	M	TE_eUICC

Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
4.2.25.2.1	TC_eUICC_ES10c.GetEID	eUICC	M	M	TE_eUICC
4.2.26.2.1	TC_eUICC_ES10c.SetNickname	eUICC	M	M	TE_eUICC
4.2.27.2.1	TC_eUICC_ES10b.GetRAT	eUICC	M	M	TE_eUICC
SM-DP+ Interfaces Compliance Testing					
4.3.12.2.1	TC_SM-DP+_ES9+.InitiateAuthenticationNIST	SM-DP+	M	M	TE_P2
4.3.12.2.2	TC_SM-DP+_ES9+.InitiateAuthenticationFRP	SM-DP+	M	M	TE_P2
4.3.12.2.3	TC_SM-DP+_ES9+.InitiateAuthenticationBRP	SM-DP+	M	M	TE_P2
4.3.13.2.1	TC_SM-DP+_ES9+.GetBoundProfilePackageNIST Test sequences #1, #2 and #5	SM-DP+	C028	C028	TE_P2
4.3.13.2.1	TC_SM-DP+_ES9+.GetBoundProfilePackageNIST Test sequences #3, #4 and #6	SM-DP+	M	M	TE_P2
4.3.13.2.2	TC_SM-DP+_ES9+.GetBoundProfilePackageFRP	SM-DP+	M	M	TE_P2
4.3.13.2.3	TC_SM-DP+_ES9+.GetBoundProfilePackageBRP Test sequence #1	SM-DP+	C028	C028	TE_P2
4.3.13.2.3	TC_SM-DP+_ES9+.GetBoundProfilePackageBRP Test sequence #2	SM-DP+	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	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	TE_P2
4.3.13.2.7	TC_SM-DP+_ES9+.GetBoundProfilePackage_RetryCases_DifferentOTPK_NIST Test sequences #1 and #2	SM-DP+	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	TE_P2
4.3.13.2.10	TC_SM-DP+_ES9+.GetBoundProfilePackage_ErrorCasesNIST	SM-DP+	M	M	TE_P2
4.3.14.2.1	TC_SM-DP+_ES9+.AuthenticateClientNIST	SM-DP+	M	M	TE_P2
4.3.14.2.2	TC_SM-DP+_ES9+.AuthenticateClientNIST_ErrorCases	SM-DP+	M	M	TE_P2
4.3.14.2.3	TC_SM-DP+_ES9+.AuthenticateClientFRP	SM-DP+	M	M	TE_P2
4.3.14.2.5	TC_SM-DP+_ES9+.AuthenticateClientBRP	SM-DP+	M	M	TE_P2

Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
4.3.14.2.6	TC_SM-DP+_ES9+.AuthenticateClient_RetryCases_Reuse_OTPK	SM-DP+	C015	C015	TE_P2
4.3.15.2.1	TC_SM-DP+_ES9+_HandleNotificationNIST	SM-DP+	M	M	TE_P2
4.3.15.2.2	TC_SM_DP+_ES9+_HandleNotificationFRP	SM-DP+	M	M	TE_P2
4.3.15.2.3	TC_SM-DP+_ES9+_HandleNotificationBRP	SM-DP+	M	M	TE_P2
4.3.16.2.1	TC_SM-DP+_ES9+.CancelSession_After_AuthenticateClientNIST	SM-DP+	M	M	TE_P2
4.3.16.2.2	TC_SM-DP+_ES9+.CancelSession_After_GetBoundProfilePackageNIST	SM-DP+	M	M	TE_P2
4.3.16.2.3	TC_SM_DP+_ES9+.CancelSession_After_AuthenticateClientFRP	SM-DP+	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	TE_P2
4.3.16.2.5	TC_SM-DP+_ES9+.CancelSession_After_AuthenticateClientBRP	SM-DP+	M	M	TE_P2
4.3.16.2.6	TC_SM-DP+_ES9+.CancelSession_After_GetBoundProfilePackageBRP	SM-DP+	M	M	TE_P2
4.3.17.1	TC_SM-DP+_ES9+_Server_Authentication_for_HTTPS_EstablishmentNIST	SM-DP+	M	M	TE_P2
4.3.17.2	TC_SM-DP+_ES9+_Server_Authentication_for_HTTPS_EstablishmentBRP	SM-DP+	M	M	TE_P2
4.3.20.1	TC_SM-DP+_ES12_Client_Mutual_Authentication_for_HTTPS_EstablishmentNIST	SM-DP+	M	M	TE_P1
4.3.20.2	TC_SM-DP+_ES12_Client_Mutual_Authentication_for_HTTPS_EstablishmentBRP	SM-DP+	M	M	TE_P1
LPAd Interfaces Compliance Testing					
4.4.21.2.1	TC_LPAd_InitiateAuthentication_Nominal	LPAd	C007	C007	
4.4.21.2.2	TC_LPAd_InitiateAuthentication_ErrorCases	LPAd	C007	C007	
4.4.22.2.1	TC_LPAd_ES9+_GetBoundProfilePackage_Nominal	LPAd	C007	C007	
4.4.22.2.2	TC_LPAd_ES9+_GetBoundProfilePackage_Retry	LPAd	C005	C005	
4.4.22.2.3	TC_LPAd_ES9+_GetBoundProfilePackage_Error	LPAd	C007	C007	
4.4.23.2.1	TC_LPAd_AuthenticateClient_Nominal	LPAd	C007	C007	

Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
4.4.23.2.2	TC_LPAd_AuthenticateClient_ErrorCases	LPAd	C007	C007	
4.4.24.2.1	TC_LPAd_ES9+_HandleNotification_Nominal	LPAd	C007	C007	
4.4.25.2.1	TC_LPAd_ES9+_CancelSession_Nominal All test sequences except the sequence #02	LPAd	C007	C007	
4.4.25.2.1	TC_LPAd_ES9+_CancelSession_Nominal Only the test sequences #02	LPAd	C023	C023	
4.4.25.2.2	TC_LPAd_ES9+_CancelSession_EndUserPostponed_Nominal	LPAd	C008	C008	
4.4.25.2.3	TC_LPAd_ES9+_CancelSession_Error	LPAd	C026	C026	
4.4.25.2.4	TC_LPAd_ES9+_CancelSession_PPRs	LPAd	C0026	C026	
4.4.26.2.1	TC_LPAd_HTTPS_Nominal	LPAd	C007	C007	
4.4.26.2.2	TC_LPAd_HTTPS_ErrorCases All test sequences except the sequence #04, #05, #06	LPAd	C007	C007	
4.4.26.2.2	TC_LPAd_HTTPS_ErrorCases Only the test sequences #04, #05, #06	LPAd	C031	C031	
4.4.27.2.1	TC_LPAd_ES11_InitiateAuthentication_Nominal	LPAd	C007	C007	
4.4.27.2.2	TC_LPAd_ES11_InitiateAuthentication_ErrorCases	LPAd	C007	C007	
4.4.28.2.1	TC_LPAd_ES11_AuthenticateClient_Nominal	LPAd	C007	C007	
4.4.28.2.2	TC_LPAd_ES11_AuthenticateClient_ErrorCases	LPAd	C007	C007	
4.4.29.2.1	TC_LPAd_HTTPS_Nominal	LPAd	C007	C007	
4.4.29.2.2	TC_LPAd_HTTPS_Error All test sequences except the sequence #04, #05, #06	LPAd	C007	C007	
4.4.29.2.2	TC_LPAd_HTTPS_Error Only the test sequences #04, #05, #06	LPAd	C031	C031	
<b>SM-DS Interfaces Compliance Testing</b>					
4.5.1.2.1	TC_ROOT_SM_DS_ES12.RegisterEvent	SM-DS	C024	C024	TE_S3
4.5.1.2.2	TC_ALT_SM_DS_ES12.RegisterEvent	SM-DS	C021	C021	TE_SA1
4.5.2.2.1	TC_ROOT_SM_DS_ES12.DeleteEvent	SM-DS	C024	C024	TE_S3
4.5.2.2.2	TC_ALT_SM_DS_ES12.DeleteEvent	SM-DS	C021	C021	TE_SA1
4.5.2.2.3	TC_ALT_SM_DS_ES12.DeleteEvent_Error_Nonexistent_EventID	SM-DS	C021	C021	TE_S2
4.5.3.2.1	TC_ROOT_SM_DS_ES15.RegisterEvent	SM-DS	C024	C024	TE_SR2
4.5.4.2.1	TC_ROOT_SM_DS_ES15.DeleteEvent	SM-DS	C024	C024	TE_SR2
4.5.5.2.1	TC_SM_DS_ES11.InitiateAuthenticationNIST	SM-DS	M	M	TE_S1

Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
4.5.6.2.1	TC_SM_DS_ES11.AuthenticateClientNIST	SM-DS	M	M	TE_S1
4.5.6.2.2	TC_SM_DS_ES11.AuthenticateClientBRP	SM-DS	M	M	TE_S1
4.5.7.1	TC_ALT_SM_DS_ES15_Client_Mutual_Authentication_for_HTTPS_EstablishmentNIST	SM-DS	C021	C021	TE_SA1
4.5.7.2	TC_ALT_SM_DS_ES15_Client_Mutual_Authentication_for_HTTPS_EstablishmentBRP	SM-DS	C021	C021	TE_SA1
4.5.8.1	TC_SM_DS_ES12_Server_Mutual_Authentication_for_HTTPS_EstablishmentNIST	SM-DS	M	M	TE_S2
4.5.8.2	TC_SM_DS_ES12_Server_Mutual_Authentication_for_HTTPS_EstablishmentBRP	SM-DS	M	M	TE_S2
4.5.9.1	TC_ROOT_SM_DS_ES15_Server_Mutual_Authentication_for_HTTPS_EstablishmentNIST	SM-DS	C024	C024	TE_SR1
4.5.9.2	TC_ROOT_SM_DS_ES15_Server_Mutual_Authentication_for_HTTPS_EstablishmentBRP	SM-DS	C024	C024	TE_SR1
4.5.10.1	TC_SM_DS_ES11_Server_Mutual_Authentication_for_HTTPS_EstablishmentNIST	SM-DS	M	M	TE_S1
4.5.10.2	TC_SM_DS_ES11_Server_Mutual_Authentication_for_HTTPS_EstablishmentBRP	SM-DS	M	M	TE_S1
Procedure - Behaviour Testing					
5.2.1.2.1	TC_eUICC_PrepareDownload_Retry_ReuseOTKeys	eUICC	C019	C019	TE_eUICC
5.2.1.2.2	TC_eUICC_PrepareDownload_Retry_NewOTKeys	eUICC	C020	C020	TE_eUICC
5.2.2.2.1	TC_eUICC_ForbiddenPPRs	eUICC	M	M	TE_eUICC
5.2.3.2.1	TC_eUICC_GetProfilesInfo_GetRAT_RSPSession	eUICC	M	M	TE_eUICC
5.2.4.2.1	TC_eUICC_Default_FileSystem	eUICC	M	M	TE_eUICC
5.2.5.2.1	TC_eUICC_DeleteProfile_ISDP_And_Components	eUICC	M	M	TE_eUICC
5.2.6.2.1	TC_eUICC_EnableProfile_Twice_Notifications	eUICC	M	M	TE_eUICC
5.2.7.2.1	TC_eUICC_DisableProfile_ApplicationManagement	eUICC	M	M	TE_eUICC
5.2.8.2.1	TC_eUICC_Enable_Disable_Delete_Notifications	eUICC	M	M	TE_eUICC
5.3.3.2.1	TC_SM-DP+_ProfileMetadata	SM-DP+	M	M	
5.4.1.2.1	TC_LPAd_AddProfile_Manual_Entry	LPAd	C007	C007	
5.4.1.2.2	TC_LPAd_AddProfile_QRcode_scanning	LPAd	C007	C007	
5.4.1.2.3	TC_LPAd_AddProfile_ActivationCode_InvalidForm at_QRcode	LPAd	C007	C007	
5.4.1.2.4	TC_LPAd_AddProfile_ActivationCode_InvalidForm at_ManualEntry	LPAd	C007	C007	
5.4.1.2.5	TC_LPAd_AddProfile_ConfirmationCode_smdpSigned2_QR	LPAd	C007	C007	



Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
5.4.1.2.6	TC_LPAd_AddProfile_ConfirmationCode_smdpSigned2_Manual_Entry	LPAd	C007	C007	
5.4.1.2.7	TC_LPAd_AddProfile_default_SM-DP+_address	LPAd	C007	C007	
5.4.1.2.8	TC_LPAd_AddProfile_QRCode_with_ConfirmationCode	LPAd	C007	C007	
5.4.1.2.9	TC_LPAd_AddProfile_PPRs	LPAd	C007	C007	
5.4.1.2.10	TC_LPAd_LUI_access_protected	LPAd	C007	C007	
5.4.1.2.11	TC_LPAd_AddProfile_Security_Errors	LPAd	C007	C007	
5.4.1.2.12	TC_LPAd_AddProfile_Empty_MatchingID	LPAd	C007	C007	
5.4.2.2.1	TC_LPAd_ListProfiles	LPAd	C007	C007	
5.4.3.2.1	TC_LPAd_SetNickname	LPAd	C027	C027	
5.4.3.2.2	TC_LPAd_EditNickname	LPAd	C027	C027	
5.4.4.2.1	TC_LPAd_DeleteProfile_Disabled_without_PPR	LPAd	C007	C007	
5.4.4.2.2	TC_LPAd_DeleteProfile_Enabled_without_PPR	LPAd	C009	C009	
5.4.4.2.3	TC_LPAd_DeleteProfile_Error_with_PPR1	LPAd	C012	C012	
5.4.4.2.4	TC_LPAd_DeleteProfile_Error_Disabled_with_PPR2	LPAd	C013	C013	
5.4.4.2.5	TC_LPAd_DeleteProfile_Error_Enabled_with_PPR2	LPAd	C014	C014	
5.4.4.2.6	TC_LPAd_DeleteProfile_Security_Errors	LPAd	C007	C007	
5.4.5.2.1	TC_LPAd_EnableProfile	LPAd	C009	C009	
5.4.5.2.2	TC_LPAd_EnableProfile_ImplicitDisable	LPAd	C009	C009	
5.4.5.2.3	TC_LPAd_EnableProfile_Error_ProfileAlreadyEnabled	LPAd	C010	C010	
5.4.5.2.4	TC_LPAd_EnableProfile_Error_PPR1Set	LPAd	C011	C011	
5.4.5.2.5	TC_LPAd_EnableProfile_Security_Errors	LPAd	C007	C007	
5.4.6.2.1	TC_LPAd_DisableProfile	LPAd	C009	C009	
5.4.6.2.2	TC_LPAd_DisableProfile_Error_ProfileAlreadyDisabled	LPAd	C017	C017	
5.4.6.2.3	TC_LPAd_DisableProfile_Error_PPR1Set	LPAd	C018	C018	
5.4.6.2.4	TC_LPAd_DisableProfile_Security_Errors	LPAd	C007	C007	
5.4.7.2.1	TC_LPAd_RetrieveEID	LPAd	C004	C004	
5.4.8.2.1	TC_LPAd_eUICCMemoryReset	LPAd	C007	C007	
5.4.8.2.2	TC_LPAd_eUICCMemoryResetWithPINVerification	LPAd	C009	C009	
5.4.10.2.1	TC_LPAd_Set/Edit Default SM-DP+ Address	LPAd	C007	C007	
5.4.11.2.1	TC_LPAd_DevicePowerOnProfileDiscovery_SM-	LPAd	C022	C022	

Test case	Name	Roles	V2.1	V2.2.X (X≥0)	Test Env.
	DP+_address				
5.4.11.2.2	TC_LPAd_DevicePowerOnProfileDiscovery_SM-DS	LPAd	C022	C022	
Test Specifications					
7.1	SIMAlliance eUICC Profile Package Test Specification	eUICC	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_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_ENPREVPPR1 AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEN M ELSE N/A
C012	IF (O_D_LPAD AND O_D_DISDELPPR1 AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEN M ELSE N/A
C013	IF (O_D_LPAD AND O_D_DELPFR2 AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEN M ELSE N/A
C014	IF (O_D_LPAD AND O_D_PIN AND O_D_DELPFR2 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) THEN M ELSE N/A
C018	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_DISPPR1) 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

Conditional item	Condition
	O_D_ONLY_CELLULAR_CONNECTIVITY) 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	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_TLS_FULL_VERIFICATION) THEN M ELSE N/A

**Table 6: Conditional Items Referenced by Table 5**

## 2.2 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.

### 2.2.1 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.

### 2.2.2 Test Cases Format

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

<b>4.X.Y.Z Test Cases</b>	
<b>4.X.Y.Z.1 TC_IUT_TestName1</b>	
<b>General Initial Conditions</b>	
<b>Entity</b>	<b>Description of the general initial condition</b>

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.

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.

### **2.2.2.1 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.

### **2.2.2.2 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.

### 2.2.2.3 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.

NOTE: see exception in section 2.2.3.3 for `ProfileInfo.profileClass` when the eUICC supports SGP.22 v2.1 [2a].

#### 2.2.2.4 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).

#### 2.2.2.5 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 SIMalliance 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.

### 2.2.2.6 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.

### 2.2.2.7 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]

## 2.2.3 General Rules for eUICC Testing

### 2.2.3.1 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>)

### 2.2.3.2 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

### 2.2.3.3 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 two different DER encodings if the eUICC supports SGP.22 V2.1 [2a]:

- either a tag 0x95 containing the integer value 2
- or an absent tag

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.

## 2.2.4 General Rules for Device Testing

### 2.2.4.1 Default Profile Download, install and enable 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 <EUIICC\_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)

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]) 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 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.

#### **2.2.4.2 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 Confirmation is enforced upon entry to the LUI as defined in section 3.2 Local Profile Management of SGP.22 [2].

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

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. E.g.: upon installation of a new Profile, the LPA MAY 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 SHALL select the confirmation only for the single actual operation (i.e. select “Install only”).

#### **2.2.4.3 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.

#### **2.2.5 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.

#### **2.2.6 Future Study**

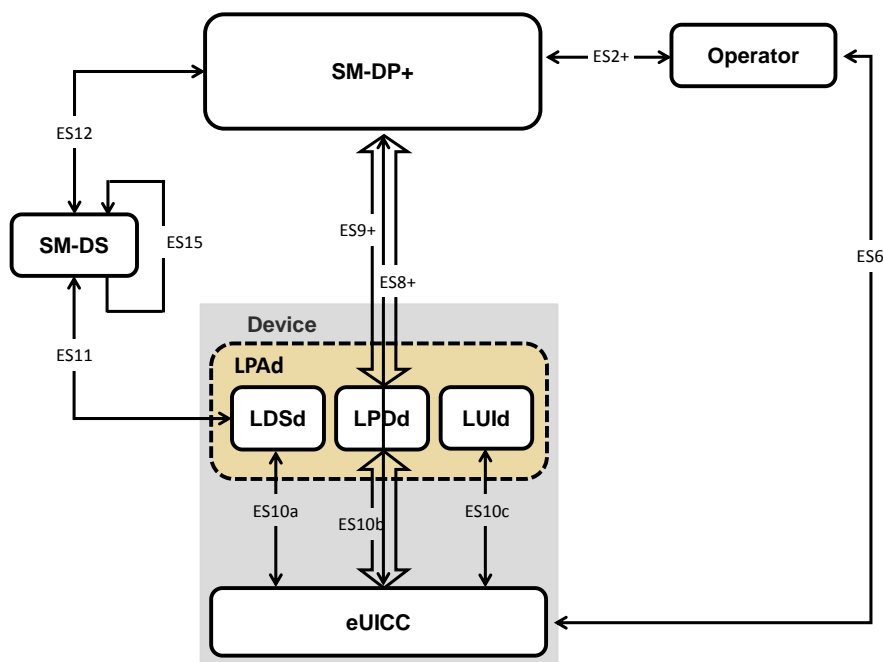
Some of the test cases or test sequences described in this Test Plan are FFS (For Future Study). This MAY mean that some clarifications are expected at the requirement level to

conclude on a test method. As consequence, the corresponding test SHALL not be executed.

## Testing Architecture

### 3.1 Testing Scope

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



↔ ↔ Interface tunneled within other interfaces

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

Interface	Between		Description
			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:

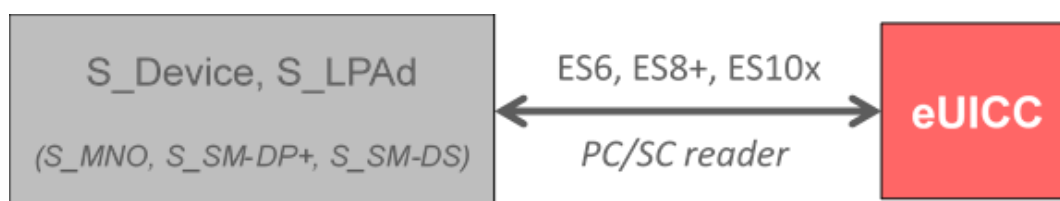
- S\_ComponentName for a simulated component

- `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 card™
- 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
- LPA / 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)



Test Environment reference:

- TE\_P2 (SM-DP+ on ES9+)



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

Test Environment reference:

- TE\_S1 (SM-DS on ES11)



Test Environment reference:

- TE\_S2 (SM-DS on ES12)



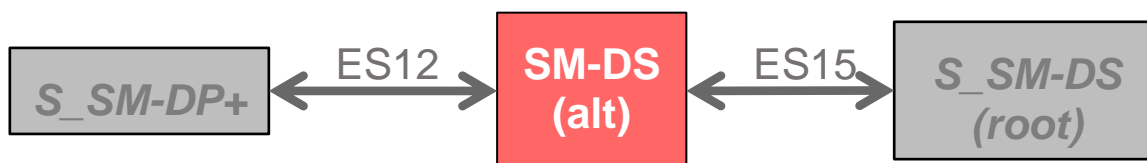
Test Environment reference:

- TE\_S3 (SM-DS on ES12 and ES11)



Test Environment reference:

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



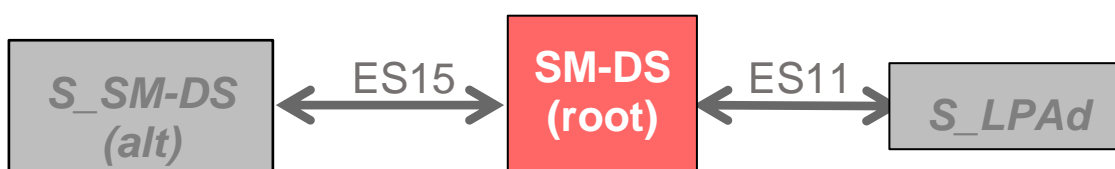
Test Environment reference:

- TE\_SR1 (Root SM-DS on ES15)



Test Environment reference:

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



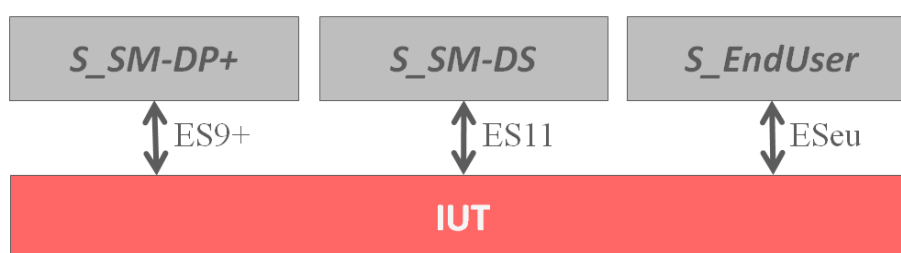
### 3.2.3 Device/LPAd - Test Environment

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 LPAd
- 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.

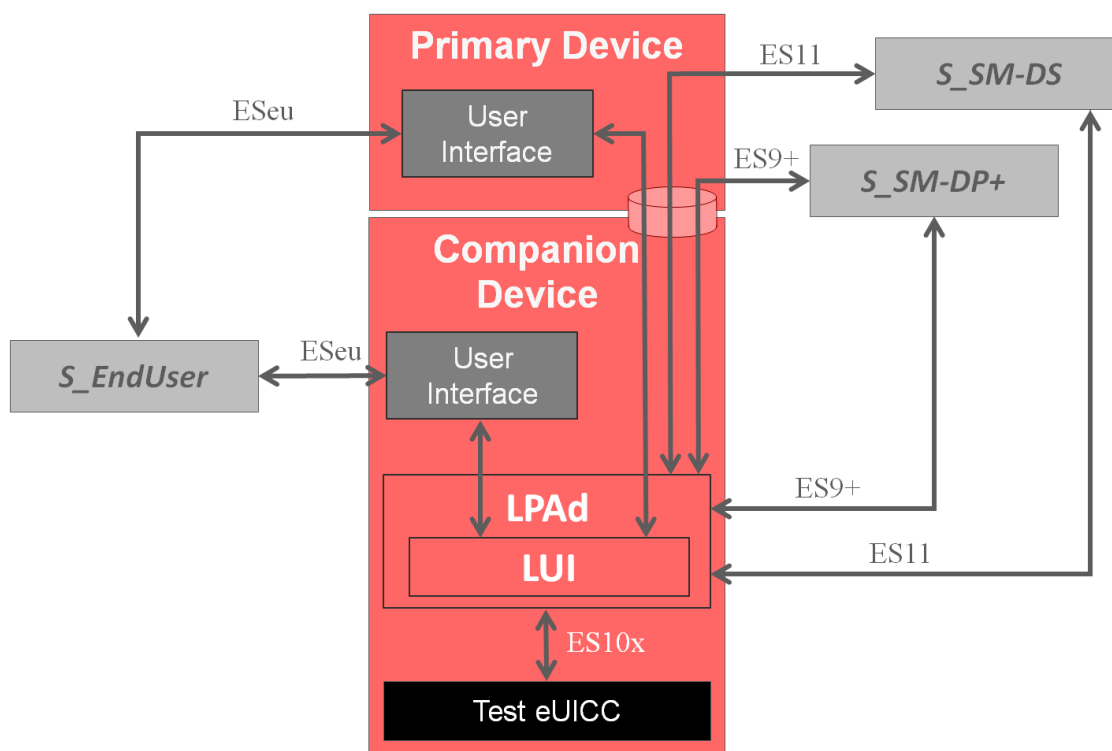
Device Vendors SHALL provide the mechanism to connect the Primary Device to the Companion Device.

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.

## 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

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

#### 4.2.1.2.1 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

#### TC\_eUICC\_ES6.UpdateMetadata

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

4.2.2.2.1

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_009 RQ54_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_009 RQ54_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(	SW=0x91XX	RQ54_001 RQ54_002

		[INSTALL_PERSO_RES_ISDP]; MTD_STORE_DATA_SCRIPT( #UPD_NAMES_REM_PPRS, FALSE))		RQ54_003 RQ54_004 RQ54_005 RQ54_006 RQ54_007 RQ54_009 RQ54_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_1 RQ57_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_009 RQ54_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( #R_GET_UPDATE_N4	#R_GET_UPDATE_N4	RQ54_013

		#GET_NEW_METADATA)	SW=0x9000	RQ54_013_1 RQ54_009 RQ57_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_UNCHAN GED 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	RQ54_014 RQ57_120 RQ57_122

			SW=0x9000	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

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

General Initial Conditions	
Entity	Description of the general initial condition
4.2.3.2.1 eUICC	The PROFILE_OPERATIONAL1 is not loaded on the eUICC.
eUICC	<p>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+</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul> <p>Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

#### 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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>		
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_013 RQ57_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

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

4 General Initial Conditions	
Entity	Description of the general initial condition
eUICC	The PROFILE_OPERATIONAL1 is not loaded on the eUICC.
eUICC	<p>The communication between the S_Device and the eUICC has been initialized and the S_LPAd has selected the ISD-R.</p> <p>Common Mutual Authentication procedure has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul> <p>Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

###### 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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> </ul>		

	• <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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_A1>)	SW=0x9000 without response data for all STORE DATA commands	
3	S_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_INIT>)	SW=0x9000 without response data for all STORE DATA commands	
1	S_LPA <sub>d</sub> → 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**

**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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul> Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+ <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

**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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> </ul>		

			<ul style="list-style-type: none"> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>	
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_LPAAd → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_INIT>)	SW=0x9000 without response data for all STORE DATA commands	
IC5	S_LPAAd → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_A0>)	SW=0x9000 without response data for all STORE DATA commands	
1	S_LPAAd → 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_LPAAd → 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_LPAAd → 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		<p>The communication between the S_Device and the eUICC has been initialized and the S_LPAd has selected the ISD-R.</p> <p>Common Mutual Authentication procedure has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>#GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>the same GSMA CI has been chosen for signing and for verification</li> </ul> <p>Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>#PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>		
IC2		Generate the <OTPK_S_SM_DP+_ECKA> and <OT_SK_S_SM_DP+_ECKA>		
IC3		<pre>&lt;BPP&gt; = MTD_GENERATE_BPP(     #S_INIT_SC_PROF1,     #CONF_ISDP_PROF1,     #METADATA_OP_PROF1,     NO_PARAM,     #UPP_OP_PROF1)</pre>		
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>)	<p>SW=0x9000 without response data for all STORE DATA commands except for the last one</p> <p>SW=0x9000 with the response data #R_PIR_ICCID_ALREADY_EXIST for the last STORE DATA command</p> <p>The euiccSignPIR SHALL be verified with the #PK_EUICC_ECDSA</p>	<p>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</p>

**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_PPR_NO_OWNER, NO_PARAM, #UPP_OP_PROF1)		

	#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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_A0>)	SW=0x9000 without response data for all STORE DATA commands	
1	S_LPA <sub>d</sub> → 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

#### 4.2.6 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	<p>The communication between the S_Device and the eUICC has been initialized and the S_LPA<sub>d</sub> has selected the ISD-R.</p> <p>Common Mutual Authentication procedure has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification and for verification</li> </ul> <p>Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

**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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A2&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>		
IC4	S_LPA → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_INIT>)	SW=0x9000 without response data for all STORE DATA commands	
IC5	S_LPA → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_A0>)	SW=0x9000 without response data for all STORE DATA commands	
IC6	S_LPA → eUICC	MTD_STORE_DATA_SCRIPT( <BPP_SEG_A1>)	SW=0x9000 without response data for all STORE DATA commands	
1	S_LPA → 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

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

4 General Initial Conditions	
Entity	Description of the general initial condition
eUICC	The PROFILE_OPERATIONAL1 is not loaded on the eUICC.
eUICC	<p>The communication between the S_Device and the eUICC has been initialized and the S_LPAd has selected the ISD-R.</p> <p>Common Mutual Authentication procedure has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul> <p>Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

#### 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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>			
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 →	MTD_STORE_DATA_SCRIPT( <BPP_SEG_A0>)	SW=0x9000 without response data	

	eUICC	<BPP_SEG_A0>	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 ( <ul style="list-style-type: none"> <li>#S_INIT_SC_PROF1,</li> <li>#CONF_ISDP_PROF1,</li> <li>#METADATA_OP_PROF1,</li> <li>#REPLACE_S_KEYS_REQ,</li> <li>#UPP_OP_PROF1)</li> </ul> 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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A2&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>		
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	



		<BPP_SEG_A2>		
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 <i>iccid</i> field SHALL be set to #ICCID_OP_PROF2 in the <i>ProfileHeader</i> element		
IC3		Split the <BPP> into several segments arrays named: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>		
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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> </ul>		

			<ul style="list-style-type: none"> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>	
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_023 RQ25_024 RQ55_045_2
2	S_LPAd → eUICC	MTD_STORE_DATA(#GET_PROFILES_OWNERS)	<pre> resp ProfileInfoListResponse ::=   profileInfoListOk :{     {       profileOwner {         mccMnc #MCC_MNC9,         gid1 #GID1,         gid2 #GID2       }     }   }                     </pre> 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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>			
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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>			
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

### 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_LPAAd → eUICC	MTD_STORE_DATA( #GET_EUICC_CONFIGURED_ADDRESS ES)	#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_LPAAd → eUICC	MTD_STORE_DATA( #GET_EUICC_CONFIGURED_ADDRES SES)	#R_ES10a_GECA_DS_D P_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

##### TC\_eUICC\_ES10a.SetDefaultDpAddress

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

4.2.9.2.1

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_ADD RESS_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_ADD RESSES)	#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_ADD RESS_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_ADD RESSES)	#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_ADDRESSES_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_ADDRESSES_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**

**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	<p>The communication between the S_Device and the eUICC has been initialized and the S_LPAd has selected the ISD-R.</p> <p>Common Mutual Authentication procedure has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI based on NIST P-256 curve has been chosen for signing and for verification</li> </ul>

**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)	<p>#R_PREP_DOWNLOAD_NO_CC</p> <p>SW=0x9000</p> <p>The &lt;EUICC_SIGNATURE2&gt; SHALL be verified with the #PK_EUICC_ECDSA.</p> <p>Verify that the &lt;S_TRANSACTION_ID&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_NO_CC.</p>	<p>RQ31_130</p> <p>RQ31_131</p> <p>RQ31_132</p> <p>RQ31_133</p> <p>RQ31_134</p> <p>RQ31_135</p> <p>RQ31_139</p> <p>RQ31_140</p> <p>RQ31_141</p> <p>RQ57_025</p> <p>RQ57_026</p> <p>RQ57_027</p> <p>RQ57_028</p> <p>RQ57_029</p> <p>RQ57_030</p> <p>RQ57_034</p> <p>RQ57_035</p> <p>RQ57_036</p> <p>RQ57_037</p> <p>RQ57_038</p> <p>RQ57_039</p> <p>RQ26_029</p> <p>RQ26_030</p> <p>RQ26_011</p> <p>RQ26_034</p> <p>RQ26_035</p> <p>RQ31_062</p>

**Test Sequence #02 Nominal: With Confirmation Code**

Step	Direction	Sequence / Description	Expected result	REQ
IC1				
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				
IC2				
1	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT( #PREP_DOWNLOAD_RETRY_C C)	#R_PREP_DOWNLOAD_WITH_C C 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_C C.	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

			Verify that the <S_HASHED_CC> present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_RETRY_C C.  Verify that the <OTPK_EUICC_ECKA> present in the euiccSigned2 is not the same as in #PREP_DOWNLOAD_RETRY_C C.	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
--	--	--	--	--

**TC\_eUICC\_ES10b.PrepareDownloadBRP**

4 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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI based on BrainpoolP256r1 curve has been chosen for signing and for verification</li> </ul>

**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 4.2.10.2.3 – TC\_eUICC\_ES10b.PrepareDownloadNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

**TC\_eUICC\_ES10b.PrepareDownloadFRP**

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

**TC\_eUICC\_ES10b.PrepareDownloadErrorCases**

General Initial Conditions	
Entity	Description of the general initial condition
421024 eUICC	The PROFILE_OPERATIONAL1 is not loaded on the eUICC.
eUICC	<p>The communication between the S_Device and the eUICC has been initialized and the S_LPAd has selected the ISD-R.</p> <p>Common Mutual Authentication procedure has been successfully executed between the eUICC and the S_SM-DP+</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul>

**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_S MDP2)	#R_PREP_DOWNLOAD_INV_CE RT SW=0x9000  Verify that the <S_TRANSACTION_ID> present in the response is the same as in #PREP_DOWNLOAD_CERT_SM DP2.	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_SIG N)	#R_PREP_DOWNLOAD_INV_SIG N 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_VERI FICATION> and choose #CERT_S_SM_DPpb_ECDSA according to this CI curve.	
1	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT( #PREP_DOWNLOAD_NO_AUT H)	#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_CE RT 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

###### 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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI based on NIST P-256 curve has been chosen for signing and for verification</li> </ul>

	Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+ <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>
--	---

**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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul> NOTE: In this test sequence, the data resulting of this operation SHALL be composed of the following TLV arrays: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt; contains the tag and length fields of the BoundProfilePackage TLV plus the #S_INIT_SC_PROF1 command</li> <li>• &lt;BPP_SEG_A0&gt; contains the tag and length fields of the firstSequenceOf87 TLV plus the first 0x87 TLV containing #CONF_ISDP_PROF1 command</li> <li>• &lt;BPP_SEG_A1&gt; contains the tag and length fields of the sequenceOf88 TLV and each of the '88' TLVs containing #METADATA_OP_PROF1 command</li> <li>• &lt;BPP_SEG_A3&gt; contains the tag and length fields of the sequenceOf86 TLV and each of the '86' TLVs containing #UPP_OP_PROF1 protected with &lt;S_ENC&gt; and &lt;S_MAC&gt;</li> </ul>		
1	S_LPA → 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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A2&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul> NOTE: In this test sequence, the data resulting of this operation SHALL be composed of the following TLV arrays: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt; contains the tag and length fields of the BoundProfilePackage TLV plus the #S_INIT_SC_PROF1 command</li> <li>• &lt;BPP_SEG_A0&gt; contains the tag and length fields of the firstSequenceOf87 TLV plus the first 0x87 TLV containing #CONF_ISDP_PROF1 command</li> <li>• &lt;BPP_SEG_A1&gt; contains the tag and length fields of the sequenceOf88 TLV and each of the '88' TLVs containing #METADATA_OP_PROF1 command</li> <li>• &lt;BPP_SEG_A2&gt; contains the tag and length fields of the secondSequenceOf87 TLV plus the first '87' TLV, containing the #REPLACE_S_KEYS_REQ command</li> <li>• &lt;BPP_SEG_A3&gt; 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</li> </ul>		
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 eiccSignPIR 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))	<pre> resp ProfileInfoListResponse ::= profileInfoListOk :{     {         ...         iccid #ICCID_OP_PROF1,         isdpAid &lt;ISD_P_AID&gt;,         profileState disabled,         ...     } } SW=0x9000                 </pre>	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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI based on BrainpoolP256r1 curve has been chosen for signing and for verification</li> </ul> Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+ <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

**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.

**TC\_eUICC\_ES10b.LoadBoundProfilePackageFRP**

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

**4.2.11.2.3 TC\_eUICC\_ES10b.LoadBoundProfilePackage\_ErrorCases**

General Initial Conditions	
4 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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul> Sub-procedure Profile Download and Installation – End User Confirmation has been successfully executed between the eUICC and the S_SM-DP+ <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

**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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>		
IC4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT(<BPP_SEG_INIT>)	SW=0x9000 without response data for all STORE DATA commands	
IC5	S_LPAd →	MTD_STORE_DATA_SCRIPT(	SW=0x9000 without response data	RQ31_186_1

	eUICC	<BPP_SEG_A0>	for all STORE DATA commands	
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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> </ul>		

			<ul style="list-style-type: none"> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>	
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

##### 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT(#AU THENTICATE_SMDP)	#R_AUTHENTICATE_SMDP SW = 0x9000	
4.2.13.2.2 <sup>5</sup>	S_LPAd → eUICC	MTD_STORE_DATA (#GET_EUICC_INFO1)	#R_EUICC_INFO1 SW = 0x9000	RQ57_051

**TC\_eUICC\_ES10b.GetEUICCInfo2\_RSP\_V2.1**

**Test Sequence #01 Nominal – RSP Version 2.1**

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:	RQ43_001 RQ43_002 RQ43_003

			<ul style="list-style-type: none"> <li>• &lt;EXT_CARD_RESOURCE&gt; contains a “number of installed application” value field set to '00'</li> <li>• #IUT_TS102241_VERSION is equal to 0x090000 or higher</li> <li>• #IUT_GLOBALPLATFORM_VERSION is equal to 0x020300 or higher</li> <li>• #RSP_SVN_H is equal to 0x020100</li> <li>• #IUT_SIMA_VERSION is equal to 0x020000 or to 0x020100</li> <li>• &lt;EUICC_RSP_CAPABILITY&gt; contains                         <ul style="list-style-type: none"> <li>○ criSupport set to '0' if O_E_CRL is not supported (otherwise, it SHALL be set to '1')</li> <li>○ testProfileSupport set to '0' if O_E_TEST_PROF is not supported (otherwise, it SHALL be set to '1')</li> <li>○ rpmSupport set to '0'</li> <li>○ additionalProfile set to '1'</li> </ul> </li> <li>• #IUT_UICC_CAPABILITY contains                         <ul style="list-style-type: none"> <li>○ javacard and akaMilenage set to '1'</li> <li>○ Either akaTuak128 or akaTuak256 set to '1'</li> </ul> </li> </ul> <p>SW = 0x9000</p>	RQ43_004 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
--	--	--	---	--

4.2.13.2.3

**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: <ul style="list-style-type: none"> <li>• &lt;EXT_CARD_RESOURCE&gt; contains a “number of installed application” value field set to '00'</li> <li>• #IUT_TS102241_VERSION is equal to 0x090000 or higher</li> <li>• #IUT_GLOBALPLATFORM_VE</li> </ul>	RQ43_001 RQ43_002 RQ43_003 RQ43_004 RQ43_005 RQ43_006 RQ43_007 RQ43_008 RQ43_009 RQ43_010

			<p>RSSION is equal to 0x020300 or higher</p> <ul style="list-style-type: none"> <li>• #RSP_SVN_H is equal to 0x0202ab 'ab' representing the 'x' in version 2.2.x</li> <li>• #IUT_SIMA_VERSION is equal to 0x020100</li> <li>• &lt;EUICC_RSP_CAPABILITY&gt; contains                     <ul style="list-style-type: none"> <li>○ criSupport set to '0' if O_E_CRL is not supported (otherwise, it SHALL be set to '1')</li> <li>○ testProfileSupport set to '0' if O_E_TEST_PROF is not supported (otherwise, it SHALL be set to '1')</li> <li>○ rpmSupport set to '0'</li> <li>○ additionalProfile set to '1'</li> </ul> </li> <li>• #IUT_UICC_CAPABILITY contains                     <ul style="list-style-type: none"> <li>○ javacard and akaMilenage set to '1'</li> <li>○ Either akaTuak128 or akaTuak256 set to '1'</li> </ul> </li> </ul> <p>SW = 0x9000</p>	<p>RQ43_011                  RQ43_012                  RQ43_013                  RQ57_057_1                  RQ57_060                  RQ57_061                  RQ57_063                  RQ57_064                  RQ57_066</p>
--	--	--	--	---

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)	<p>#R_EUICC_INFO1                      SW = 0x9000                      Extract the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; and &lt;EUICC_CI_PK_ID_LIST_FOR_VERIFICATION&gt; from response data and verify if they contain at least one same GSMA CI Key ID</p>	
IC4	S_LPAd → eUICC	MTD_STORE_DATA (#GET_EUICC_CHALLENGE)	<p>#R_CHALLENGE                      SW = 0x9000                      Extract the &lt;EUICC_CHALLENGE&gt;</p>	
IC5	<p>The following inputs are required for Step IC6 as described in the InitiateAuthentication function:</p> <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> </ul>			

	<ul style="list-style-type: none"> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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)	#R_EUICC_INFO2 same EUICCInfo2 data object as in Step IC6 SW = 0x9000	RQ57_051 RQ57_053 RQ57_054 RQ57_058 RQ57_059 RQ57_062 RQ31_053 RQ31_060 RQ43_001 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.

#### 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_E RROR 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_ENAB LE)	#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_ENAB LE)	#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_DELE TE)	#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_ENAB LE)	#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_ENAB LE_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_ENABL E)	#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_DELET E)	#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_ENABL E)	#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_ENABL E_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_E RROR 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_ENAB LE)	#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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> 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 →	MTD_STORE_DATA( 	#R_LIST_NOTIF_NONE	RQ35_016

	eUICC	#LIST_NOTIF_NONE)	SW = 0x9000 OR #R_LIST_NOTIF_UNDEFINED_ERROR SW = 0x9000	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_LPA → 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_LPA → 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_LPA → eUICC	MTD_STORE_DATA( #LIST_NOTIF_NONE)	#R_LIST_NOTIF_NONE SW = 0x9000 OR #R_LIST_NOTIF_UNDEFINED_E RROR SW = 0x9000	RQ35_016 RQ57_068 RQ57_068_1 RQ57_068_4 RQ57_069
4	S_LPA → 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_LPA → 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_LPA → 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_LPA → 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_LPA →	MTD_STORE_DATA( #LIST_NOTIF_INSTALL)	#R_LIST_NOTIF_IN1_PIR_EN1 SW = 0x9000	RQ35_016



	eUICC	#LIST_NOTIF_INSTALL_ENABLE)	SW = 0x9000	RQ57_068 RQ57_068_1 RQ57_068_2 RQ57_069 RQ57_070 RQ25_020
9	S_LPA → 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_LPA → 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_LPA → 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_2 RQ57_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_LPA → 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_LPA → 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_LPA → 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	RQ35_016 RQ57_068

			SW = 0x9000	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_ENABL E)	#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_DELE TE)	#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_ENAB LE)	#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_ENABL E_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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> 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_LPA → 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_LPA → 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_LPA → 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_LPA → 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_LPA → 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_LPA → eUICC	MTD_STORE_DATA( #LIST_NOTIF_INSTALL_ENABL E)	#R_LIST_NOTIF_NONE SW = 0x9000	RQ35_016 RQ57_068 RQ57_068_1 RQ57_068_4 RQ57_069
9	S_LPA → eUICC	MTD_STORE_DATA( #LIST_NOTIF_DISABLE_DELE TE)	#R_LIST_NOTIF_NONE SW = 0x9000	RQ35_016 RQ57_068 RQ57_068_1 RQ57_068_4 RQ57_069
10	S_LPA → eUICC	MTD_STORE_DATA( #LIST_NOTIF_DISABLE_ENAB LE)	#R_LIST_NOTIF_NONE SW = 0x9000	RQ35_016 RQ57_068 RQ57_068_1 RQ57_068_4 RQ57_069
11	S_LPA → eUICC	MTD_STORE_DATA( #LIST_NOTIF_INSTALL_ENABL E_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	RQ57_071 RQ57_071_1 RQ57_071_2 RQ57_072

		))	euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_ECDSA	RQ57_072_1R Q57_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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> Do not remove the Notification.		
IC5	S_LPAd → eUICC	MTD_STORE_DATA( #LIST_NOTIF_ALL)	#R_LIST_NOTIF_IN1_PIR_IN2_PIR R 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_LPAAd → 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_LPAAd → 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_LPAAd → eUICC	MTD_STORE_DATA( #LIST_NOTIF_ALL)	#R_LIST_NOTIF_IN1_EN1 SW = 0x9000	
1	S_LPAAd → eUICC	MTD_STORE_DATA( MTD_RETRIEVE_NOTIF_SEQ_NU M(<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_LPAAd → eUICC	MTD_STORE_DATA( MTD_RETRIEVE_NOTIF_SEQ_NU M(<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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> 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_NON E 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	MTD_STORE_DATA( #RETRIEVE_NOTIF_ALL)	#R_RETRIEVE_NOTIF_IN1_IN1_PIR	RQ57_071 RQ57_071_1

	→ eUICC	#RETRIEVE_NOTIF_OMITTED)	SW = 0x9000 <ul style="list-style-type: none"> <li>• Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</li> <li>• Verify the euiccSignPIR &lt;EUICC_SIGN_PIR&gt; using the #PK_EUICC_ECDSA</li> </ul>	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_IN 1_PIR SW = 0x9000 <ul style="list-style-type: none"> <li>• Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</li> <li>• Verify the euiccSignPIR &lt;EUICC_SIGN_PIR&gt; using the #PK_EUICC_ECDSA</li> </ul>	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_IN 1_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_DELET E)	#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_ENABL E)	#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_ENABL E_DISABLE)	#R_RETRIEVE_NOTIF_IN1_IN 1_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_EN ABLE)	#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_D ELETE)	#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_E NABLE)	#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_EN ABLE_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	RQ57_071 RQ57_071_1 RQ57_071_2 RQ57_072

			<TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_ECDSA	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_OMMITTED )	#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	RQ57_071 RQ57_071_1 RQ57_071_2

			<ul style="list-style-type: none"> <li>Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</li> </ul>	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 <ul style="list-style-type: none"> <li>Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</li> </ul>	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 <ul style="list-style-type: none"> <li>Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</li> </ul>	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_E NABLE)	#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_E NABLE)	#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_E NABLE_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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> 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 →	MTD_STORE_DATA( #RETRIEVE_NOTIF_DISABLE)	#R_RETRIEVE_NOTIF_NONE	RQ57_071 RQ57_072

	eUICC	#RETRIEVE_NOTIF_DELETE)	SW = 0x9000	RQ57_072_1 RQ57_073 RQ57_074
8	S_LPAd → eUICC	MTD_STORE_DATA( #RETRIEVE_NOTIF_INSTALL_E NABLE)	#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_E NABLE_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	RQ57_071 RQ57_072 RQ57_072_1 RQ57_072_2



			euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_ECDSA <ul style="list-style-type: none"> <li>• Verify the euiccSignPIR                      &lt;EUICC_SIGN_PIR&gt; using the                      #PK_EUICC_ECDSA</li> </ul>	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 <ul style="list-style-type: none"> <li>• Verify the                      euiccNotificationSignature                      &lt;TBS_EUICC_NOTIF_SIG&gt;                      using the #PK_EUICC_ECDSA</li> <li>• Verify the euiccSignPIR                      &lt;EUICC_SIGN_PIR&gt; using the                      #PK_EUICC_ECDSA</li> </ul>	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 <ul style="list-style-type: none"> <li>• Verify the euiccSignPIR                      &lt;EUICC_SIGN_PIR&gt; using the                      #PK_EUICC_ECDSA</li> </ul>	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 <ul style="list-style-type: none"> <li>• Verify the                      euiccNotificationSignature                      &lt;TBS_EUICC_NOTIF_SIG&gt;                      using the #PK_EUICC_ECDSA</li> </ul>	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_E NABLE)	#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_E NABLE_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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> Do not remove the Notification.			
1	S_LPAd → eUICC	MTD_STORE_DATA( #RETRIEVE_NOTIF_ALL)	#R_RETRIEVE_NOTIF_EN1_IN2_PIR SW = 0x9000 <ul style="list-style-type: none"> <li>• Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</li> <li>• Verify the euiccSignPIR &lt;EUICC_SIGN_PIR&gt; using the #PK_EUICC_ECDSA</li> </ul>	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 <ul style="list-style-type: none"> <li>• Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</li> <li>• Verify the euiccSignPIR &lt;EUICC_SIGN_PIR&gt; using the #PK_EUICC_ECDSA</li> </ul>	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 <ul style="list-style-type: none"> <li>• Verify the euiccSignPIR &lt;EUICC_SIGN_PIR&gt; using the #PK_EUICC_ECDSA</li> </ul>	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	RQ57_071 RQ57_072 RQ57_072_1 RQ57_072_2 RQ57_072_3 RQ57_073_1 RQ57_074



			the #PK_EUICC_ECDSA • Verify the euiccSignPIR <EUICC_SIGN_PIR> using the #PK_EUICC_ECDSA	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_LPA → 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_LPA → 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_LPA → 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_LPA → 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_LPA → 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_LPA → 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_LPA → 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_LPA →	MTD_STORE_DATA( #RETRIEVE_NOTIF_INSTALL)	#R_RETRIEVE_NOTIF_NONE	RQ57_071 RQ57_072

	eUICC	_ENABLE)	SW = 0x9000	RQ57_072_1 RQ57_073 RQ57_074
9	S_LPAAd → 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_LPAAd → 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_LPAAd → 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.

#### 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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> </ul>			

	<ul style="list-style-type: none"> <li>#CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> 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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul> 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_LPAAd → eUICC	MTD_STORE_DATA( #LIST_NOTIF_ALL)	#R_LIST_NOTIF_IN1_PIR SW = 0x9000	
1	S_LPAAd → eUICC	MTD_STORE_DATA( MTD_REMOVE_NOTIF( <NOTIF_SEQ_NO_IN1_PIR> - 1))	#R_REMOVE_NOTIF_NOTHIN G_TO_DELETE SW = 0x9000	RQ35_021 RQ57_077 RQ57_078 RQ57_079
2	S_LPAAd → eUICC	MTD_STORE_DATA( #LIST_NOTIF_ALL)	#R_LIST_NOTIF_IN1_PIR SW = 0x9000	
3	S_LPAAd → eUICC	MTD_STORE_DATA( MTD_REMOVE_NOTIF( <NOTIF_SEQ_NO_IN1_PIR> + 1))	#R_REMOVE_NOTIF_NOTHIN G_TO_DELETE SW = 0x9000	RQ35_021 RQ57_077 RQ57_078 RQ57_079
4	S_LPAAd → eUICC	MTD_STORE_DATA( #LIST_NOTIF_ALL)	#R_LIST_NOTIF_IN1_PIR SW = 0x9000	
5	S_LPAAd → 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_LPAAd → eUICC	MTD_STORE_DATA( #LIST_NOTIF_ALL)	#R_LIST_NOTIF_NONE SW = 0x9000	
7	S_LPAAd → eUICC	MTD_STORE_DATA( MTD_REMOVE_NOTIF( <NOTIF_SEQ_NO_IN1_PIR>))	#R_REMOVE_NOTIF_NOTHIN G_TO_DELETE SW = 0x9000	RQ35_021 RQ57_077 RQ57_078 RQ57_079
8	S_LPAAd → eUICC	MTD_STORE_DATA( #LIST_NOTIF_ALL)	#R_LIST_NOTIF_NONE SW = 0x9000	
9	S_LPAAd → eUICC	MTD_STORE_DATA( MTD_REMOVE_NOTIF( <NOTIF_SEQ_NO_IN1_PIR> + 1))	#R_REMOVE_NOTIF_NOTHIN G_TO_DELETE SW = 0x9000	RQ35_021 RQ57_077 RQ57_078 RQ57_079
10	S_LPAAd → 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

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

#### 4.2.18.2.1

#### 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on NIST P-256 curve</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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	RQ26_029 RQ26_005 RQ26_006 RQ26_007 RQ26_008 RQ26_034 RQ26_035

			<p>&lt;S_TRANSACTION_ID&gt; present in the <code>eUICC_Signed1</code> is the same as in <code>#AUTHENTICATE_SMDP</code>.</p> <ul style="list-style-type: none"> <li>Verify that the &lt;S_SMDP_CHALLENGE&gt; present in the <code>eUICC_Signed1</code> is the same as in <code>#AUTHENTICATE_SMDP</code></li> </ul>	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: <ul style="list-style-type: none"> <li>&lt;S_TRANSACTION_ID&gt;</li> <li>&lt;EUICC_CHALLENGE&gt;</li> </ul>			

			<ul style="list-style-type: none"> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on NIST P-256 curve</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>	
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT( #AUTH_SMDP_MATCH_ID)	<p>#R_AUTH_SMDP_MATCH_ID SW = 0x9000</p> <ul style="list-style-type: none"> <li>• Verify the &lt;EUICC_SIGNATURE1&gt; using the #PK_EUICC_ECDSA</li> <li>• Verify that the &lt;S_TRANSACTION_ID&gt; present in the euiCCSigned1 is the same as in #AUTH_SMDP_MATCH_ID</li> <li>• Verify that the &lt;S_SMDP_CHALLENGE&gt; present in the euiCCSigned1 is the same as in #AUTH_SMDP_MATCH_ID</li> </ul>	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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on NIST P-256 curve</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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 <i>euiccSigned1</i> is the same as in #AUTH_SMDP_IMEI • Verify that the <S_SMDP_CHALLENGE> present in the <i>euiccSigned1</i> 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

**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_SIGN ING> and <EUICC_CI_PK_ID_LIST_FOR_VERI FICATION> 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on BrainpoolP256r1 curve</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on BrainpoolP256r1 curve</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>		
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	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

			<S_SMDP_CHALLENGE> present in the euiccSigned1 is the same as in #AUTH_SMDP_MATCH_ID	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_S I GNING> and <EUICC_CI_PK_ID_LIST_FOR_V E RIFICATION> 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on BrainpoolP256r1 curve</li> </ul>			

<ul style="list-style-type: none"> <li>Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>				
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT(#A UTH_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**

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

**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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPauth_INV_SIGN leading to the same Root CI certificate apart from the signature</li> </ul>			
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT (#AUTH_SMDP_INV_CERT)	#R_AUTH_SERVER_INV_CERT SW = 0x9000 <ul style="list-style-type: none"> <li>• Verify that the &lt;S_TRANSACTION_ID&gt; present in the AuthenticateResponseError is the same as in #AUTH_SMDP_INV_CERT.</li> </ul>	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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt; NOT computed with the #SK_S_SM_DPauth_ECDSA but SHALL have the same length as for a valid signature</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT(#AUTHENTICATE_SMDP)	#R_AUTH_SERVER_INV_SIGN SW = 0x9000	RQ31_052 RQ45_006 RQ45_026_1 RQ45_026 RQ45_028

			<ul style="list-style-type: none"> <li>• Verify that the &lt;S_TRANSACTION_ID&gt; present in the AuthenticateResponseError is the same as in #AUTHENTICATE_SMDP</li> </ul>	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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;RANDOM_SM_DP+_SIGN&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• #CERT_S_SM_DPauth_INV_CURVE</li> </ul>			
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT (#AUTH_SMDP_INV_CURV)	#R_AUTH_SERVER_INV_CURV SW = 0x9000  <ul style="list-style-type: none"> <li>• Verify that the &lt;S_TRANSACTION_ID&gt; present in the AuthenticateResponseError is the same as in #AUTH_SMDP_INV_CURV.</li> </ul>	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_LPAAd → 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• #S_EUICC_CHALLENGE considered as different from &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
4	S_LPAAd → 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_LPAAd → eUICC	MTD_STORE_DATA (#GET_EUICC_INFO1)	#R_EUICC_INFO1 SW = 0x9000	
2	S_LPAAd → 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to a CI Key ID not present in the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; (a random SubjectKeyIdentifier can be used)</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_VERIFICATION&gt;</li> </ul>			
4	S_LPAAd → 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPpb_ECDSA (instead of #CERT_S_SM_DPauth_ECDSA) leading to the same Root CI certificate</li> </ul>			
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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• #S_EUICC_CHALLENGE</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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_006 RQ31_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 CtxParams 1**

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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on BrainpoolP256r1 curve</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate</li> </ul>			

4	S_LPA <sub>d</sub> → 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_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

**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_LPA <sub>d</sub> → 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_LPA <sub>d</sub> →	MTD_STORE_DATA	#R_CHALLENGE	

	eUICC	(#GET_EUICC_CHALLENGE )	SW = 0x9000 Extract the <EUICC_CHALLENGE>	
3	The following inputs are required for Step 4 as described in the InitiateAuthentication function: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on BrainpoolP256r1 curve</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate</li> </ul>			
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT( #AUTH_SMDS_IMEI)	#R_AUTH_SMDS_IMEI SW = 0x9000 <ul style="list-style-type: none"> <li>• Verify the &lt;EUICC_SIGNATURE1&gt; using the #PK_EUICC_ECDSA</li> <li>• Verify that the &lt;S_TRANSACTION_ID&gt; present in the euiccSigned1 is the same as in #AUTH_SMDS_IMEI</li> <li>• Verify that the &lt;S_SMDS_CHALLENGE&gt; present in the euiccSigned1 is the same as in #AUTH_SMDS_IMEI</li> </ul>	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_S IGNING> and <EUICC_CI_PK_ID_LIST_FOR_VER IFICATION> 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on NIST P-256 curve</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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_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 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_S IGNING> and <EUICC_CI_PK_ID_LIST_FOR_V ERIFICATION> 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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID based on NIST P-256 curve</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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 RQ36_017
--	--	--	--	--

**TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_FRP**

This test case is defined as FFS and not applicable for this version of test specification.  
 4.2.18.2.7

**TC\_eUICC\_ES10b.AuthenticateServer\_SM-DS\_ErrorCases**

4.2.18.2.8

**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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DSauth_INV_SIGN leading to the same Root CI certificate</li> </ul>			
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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt; NOT computed with the #SK_S_SM_DSauth_ECDSA but SHALL have the same length as for a valid signature</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate</li> </ul>			
4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT( #AUTHENTICATE_SMDS)	#R_AUTH_SERVER_INV_SIGN SW = 0x9000 <ul style="list-style-type: none"> <li>• Verify that the &lt;S_TRANSACTION_ID&gt; present in the AuthenticateResponseError is the same as in #AUTHENTICATE_SMDS</li> </ul>	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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;RANDOM_SM_DS_SIGN&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• #CERT_S_SM_DSauth_INV_CURVE</li> </ul>			

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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• #S_EUICC_CHALLENGE considered as different from &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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 AuthenticateResponseErr or 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 →	MTD_STORE_DATA	#R_CHALLENGE	

	eUICC	(#GET_EUICC_CHALLENGE)	SW = 0x9000 Extract the <EUICC_CHALLENGE>	
3	The following inputs are required for Step 4 as described in the InitiateAuthentication function: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to a CI Key ID not present in the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; (a random SubjectKeyIdentifier can be used)</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_VERIFICATION&gt;</li> </ul>			
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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• #S_EUICC_CHALLENGE</li> <li>• &lt;S_SMDS_CHALLENGE&gt;</li> <li>• &lt;S_SMDS_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate</li> </ul>			
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	RQ57_100 RQ31_052 RQ57_094 RQ57_105 RQ57_107

			(any value SHALL be accepted)	
--	--	--	-------------------------------	--

#### 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

###### 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	<p>The communication between the S_Device and the eUICC has been initialized and the S_LPAd has selected the ISD-R.</p> <p>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)</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI based on NIST P-256 curve has been chosen for signing and for verification</li> </ul>

###### Test Sequence #01 Nominal: End User Rejection

Step	Direction	Sequence / Description	Expected result	REQ
1	S_LPAd → eUICC	MTD_STORE_DATA(#CANCEL_SESSION_REJECT)	<p>#R_CANCEL_SESSION_REJ                      SW = 0x9000</p> <p>The &lt;EUICC_CS_SIGNATURE&gt; SHALL be verified with the #PK_EUICC_ECDSA.</p> <p>Verify that the &lt;S_TRANSACTION_ID&gt; present in the response is the same as in #CANCEL_SESSION_REJECT</p>	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_LPAAd → 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 LPAAd.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_LPAAd → 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 LPAAd detected that PPR(s) set in the Profile Metadata is/are not allowed.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_LPAAd → 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) <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>			
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 LPAAd 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) <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
1	S_LPAAd → 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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>			
5	S_LPAAd → 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 LPAAd 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) <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>		
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>	RQ31_114 RQ31_115 RQ31_116 RQ31_117

	)	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	RQ57_110 RQ57_111 RQ57_114 RQ57_115 RQ57_116 RQ26_034 RQ26_035
2	PROC_VERIFY_SESSION_IS_CANCELLED		RQ57_113

### 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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI based on BrainpoolP256r1 curve has been chosen for signing and for verification</li> </ul>

#### **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+ <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

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+ <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

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+ <ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC has been sent to the eUICC</li> </ul>

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.

**TC\_eUICC\_ES10b.CancelSessionFRP**

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

**4.2.19.2.3 TC\_eUICC\_ES10b.CancelSession\_ErrorCase**

**4.2.19.2.4 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 ANS_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. <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul>

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 ANS_ID SW = 0x9000	RQ57_109 RQ57_114 RQ57_115

2	S_LPA → eUICC	MTD_STORE_DATA_SCRIPT( #PREP_DOWNLOAD_NO_CC)	#R_PREP_DOWNLOAD_NO_C C 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_LPA → eUICC	MTD_STORE_DATA (#GET_PROFILES_INFO_ALL)	response ProfileInfoListResponse::= profileInfoListOk: {  #PROFILE_INFO1, #PROFILE_INFO2, #PROFILE_INFO3  }	RQ32_057 RQ57_117 RQ57_118 RQ57_119 RQ57_123 RQ24_029 RQ31_183

			SW = 0x9000	
--	--	--	-------------	--

**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	RQ57_119 RQ57_120 RQ57_122 RQ57_123

			} SW = 0x9000	
--	--	--	------------------	--

**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 T_PROFILE_NAME)	response ProfileInfoListResponse::= profileInfoListOk : { #PROFILES_INFO_TAGLIST_P ROFILE_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_I CON } 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_LPA → 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_LPA → 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_LPA → 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_LPAAd → eUICC	MTD_STORE_DATA (#GET_PROFILES_INFO_TAGLIST2 )	response ProfileInfoListResponse::= profileInfoListOk : { #PROFILES_INFO_TAGLIST 2 } 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_LPAAd → eUICC	MTD_STORE_DATA (#GET_PROFILES_INFO_TAGLIST3 )	response ProfileInfoListResponse::= profileInfoListOk : { #PROFILES_INFO_TAGLIST 3 } 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_LPAAd → 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_LPAAd →	MTD_STORE_DATA (#GET_PROFILES_INFO_OPTAGLI	response	RQ57_120 RQ57_122

	eUICC	ST1)	ProfileInfoListResponse::= profileInfoListOk : { #PROFILES_INFO_TAGLIST1 } SW = 0x9000	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_OPTAGLI ST2)	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_OPTAGLI ST3)	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_OPTAGLI ST4)	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 →	MTD_STORE_DATA	response	RQ57_120

	eUICC	(#GET_PROFILES_INFO_ICCID_TAGLIST4)	ProfileInfoListResponse::= profileInfoListOk : { #PROFILES_INFO_ICCID_TAGLIST4 } SW = 0x9000	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**

**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,	response ProfileInfoListResponse::= profileInfoListOk : {	RQ24_010 RQ32_011 RQ32_016_1 RQ32_016_2

		<ISD_P_AID1>))	#PROFILE_INFO1 } SW=0x9000	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_LPA → 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_LPA → 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

### TC\_eUICC\_ES10c.EnableProfile\_ErrorCases\_Case3

4 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,	response ProfileInfoListResponse::= profileInfoListOk : { #PROFILE_INFO1_DISABLE	RQ32_011 RQ32_012 RQ32_016_1 RQ57_128 RQ57_129

		<ISD_P_AID1>))	D } SW=0x9000	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_LPAAd → 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_LPAAd → 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**

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_1 RQ57_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**

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_1 RQ57_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 } }	RQ32_011 RQ32_016_1 RQ32_016_2 RQ57_128 RQ57_129 RQ57_133_3

			SW=0x9000	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_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_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_O K 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

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>,	#R_ENABLE_PROFILE_ICCID_IS DP_NOTFOUND SW=0x9000	RQ32_011 RQ32_016_1 RQ57_127 RQ57_128 RQ57_129

		TRUE))		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_IS DP_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_DISALLOW EDbyPOLICY 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_DISALLOW EDbyPOLICY 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**

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( 	SW=0x91XX	

		[GET_MNO_SD])		
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**

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_DISABL ED } 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_DISABLE D } 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_DISAB LED } 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 LED } 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

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_DISABLE D } 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	MTD_STORE_DATA( 	response	RQ32_031

	→eUICC	MTD_GET_PROFILE_INFO( #ICCID_OP_PROF1, NO_PARAM))	ProfileInfoListResponse::= profileInfoListOk : { #PROFILE_INFO1_DISAB LED } SW=0x9000	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**

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**

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	

**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,	response ProfileInfoListResponse::= profileInfoListOk : { #PROFILE_INFO1_DISABLE	RQ32_031 RQ32_033 RQ32_037_1 RQ57_142_2 RQ57_142_3



		<ISD_P_AID1>))	D } SW=0x9000	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_DISABLE D } 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_DISABLE D } 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_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA( MTD_GET_PROFILE_INFO( NO_PARAM, <ISD_P_AID1>))	response ProfileInfoListResponse::= profileInfoListOk : {  #PROFILE_INFO01_DISAB LED  } 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_LPA <sub>d</sub> → 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_DISAB LED } 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

#### TC\_eUICC\_ES10c.DisableProfile\_ErrorCases\_Case4

<b>4 General Initial Conditions</b>	
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_I SDP_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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA( MTD_DISABLE_PROFILE( #ICCID_OP_PROFX, NO_PARAM, TRUE))	#R_DISABLE_PROFILE_ICCID_IS DP_NOTFOUND SW=0x9000	RQ32_031 RQ32_032 RQ57_142_2 RQ57_142_3 RQ57_142_4 RQ57_142_2 RQ57_142_1 5 RQ57_149 RQ57_150
2	S_LPA <sub>d</sub> → 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_1 5 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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA( MTD_DISABLE_PROFILE( #R_DISABLE_PROFILE_NOT_E NABLE_STATE	#R_DISABLE_PROFILE_NOT_E NABLE_STATE	RQ32_031 RQ32_032 RQ57_142_2

		NO_PARAM, <ISD_P_AID1>, TRUE))	SW=0x9000	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_E NABLE_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_DISALL OWEDbyPOLICY 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_DISALL OWEDbyPOLICY 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**

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( #ICCID_OP_PROF4, NO_PARAM, TRUE))	resp DisableProfileResponse ::= { disableResult catBusy	RQ57_142

		NO_PARAM, <ISD_P_AID1>, FALSE))	} SW=0x9000 or 0x91XX	
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**

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

##### TC\_eUICC\_ES10c.DeleteProfile\_Case3

4.2.23.2.1

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_LPA → 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_LPA → eUICC	MTD_STORE_DATA( MTD_GET_PROFILE_INFO( NO_PARAM,	response ProfileInfoListResponse::= profileInfoListOk: { }	RQ57_119 RQ24_010

		<ISD_P_AID1>))	SW=0x9000	
--	--	----------------	-----------	--

**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_LPAAd → 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_LPAAd →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	RQ57_154 RQ57_156 RQ57_160 RQ57_162

			} SW=0x9000	
--	--	--	----------------	--

**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,	response ProfileInfoListResponse::= profileInfoListOk: { } }	RQ24_020 RQ57_119

		NO_PARAM))	SW=0x9000	
--	--	------------	-----------	--

**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_ISD P_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_LPA → eUICC	MTD_STORE_DATA( MTD_DELETE_PROFILE( #ICCID_OP_PROFX, NO_PARAM)	#R_DELETE_PROFILE_ICCID_ISD P_NOTFOUND SW=0x9000	RQ24_016 RQ32_049 RQ57_154 RQ57_157 RQ57_158 RQ57_159
2	S_LPA → 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_LPA → eUICC	MTD_STORE_DATA( MTD_DELETE_PROFILE( #R_DELETE_PROFILE_NOTDIS ABLESTATE SW=0x9000	RQ24_016 RQ32_049 RQ32_050 RQ57_154	

		NO_PARAM, <ISD_P_AID2>)		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_NOTDISAB LESTATE 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_LPA → eUICC	MTD_STORE_DATA( MTD_DELETE_PROFILE( NO_PARAM, <ISD_P_AID3>)	#R_DELETE_PROFILE_DISALLO WEDBYPOLICY SW=0x9000	RQ24_016 RQ29_002 RQ32_049 RQ32_050 RQ57_154 RQ57_156 RQ57_158 RQ57_159 RQ29_002 RQ29_022
2	S_LPA → 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_DISALLOWE DBYPOLICY 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

##### 4.2.24.2.1

##### 4.2.24.2 Test Cases

##### 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_INST ALLED>	
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_PROFI LE> Verify that <FREE_MEM_OP_PROF_INST ALLED> is lower than <FREE_MEMORY_NO_PROFI LE>	RQ31_027 RQ31_028 RQ57_051 RQ57_052 RQ57_054 RQ24_020

6	S_LPAd → eUICC	MTD_STORE_DATA( #GET_EUICC_CONFIGURED_AD DRESSES)	#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_O P_PRO)	#R_EUICC_MEMORY_RESET_O K SW=0x91XX	RQ57_163 RQ57_166 RQ57_169 RQ57_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_1 RQ35_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_SMD ADDRESS)	#R_EUICC_MEMORY_RE SET_OK SW=0x9000	RQ57_163 RQ57_167 RQ57_169 RQ57_170 RQ33_010
2	S_LPAd → eUICC	MTD_STORE_DATA( #GET_PROFILES_INFO_ALL)	response ProfileInfoListResponse::= profileInfoListOk: { #PROFILE_INFO1_DISAB LED, #PROFILE_INFO3 } SW=0x9000	
3	S_LPAd → eUICC	MTD_STORE_DATA( #GET_EUICC_CONFIGURED_ADDRES SES)	#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( #GET_EUICC_CONFIGURED_ADDRES SES)	#R_EUICC_MEMORY_RE SET_OK	RQ57_163 RQ57_166

		#EUICC_MEMORY_RESET)	SW=0x9000	RQ57_167 RQ57_169 RQ57_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_ADDRES SES)	#R_ES10a_GECA_DS SW = 0x9000	RQ33_008

### TC\_eUICC\_ES10c.eUICCMemoryReset\_ErrorCases

#### 4.2.24.2.2 **Test Sequence #01 Error: eUICC Memory Reset while proactive session is ongoing**

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_O P_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,	RQ57_165

			#PROFILE_INFO2 } SW=0x9000	
--	--	--	----------------------------------	--

**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_LPA → 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**

**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_LPA → eUICC	MTD_STORE_DATA( #GET_EID)	resp GetEuiccDataResponse ::= { eidValue #EID1 }	RQ33_002 RQ57_171 RQ57_172

			SW=0x9000	
--	--	--	-----------	--

### 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_LPA → 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_LPA → eUICC	MTD_STORE_DATA( #SET_NICKNAME_OP_PROF1	resp SetNicknameResponse ::= { setNicknameResult ok }	RQ57_177 RQ57_178

		)	SW=0x9000	
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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA( #SET_NICKNAME_EMPTY_O P_PROF1)	resp SetNicknameResponse ::= { setNicknameResult ok } SW=0x9000	RQ57_177 RQ57_178
2	S_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA( #SET_NICKNAME_EMPTY_O P_PROF1)	resp SetNicknameResponse ::= { setNicknameResult ok } SW=0x9000	RQ57_177 RQ57_178
2	S_LPA <sub>d</sub> → 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_LPA → eUICC	MTD_STORE_DATA( #SET_NICKNAME_ICCID_UN KNOWN)	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_LPA → 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**

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

#### **4.3.2 ES2+ (Operator -- SM-DP+): ConfirmOrder**

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

#### **4.3.3 ES2+ (Operator -- SM-DP+): CancelOrder**

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

#### **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.

**TC\_SM-DP+\_ES9+.InitiateAuthenticationNIST**

**4.3.12.2.1**

**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_AUTHENTICAT ION( #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_AUTHENTICATI ON( #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_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #SERVER_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK_2)  Verify that: <ul style="list-style-type: none"> <li>• &lt;TRANSACTION_ID_2&gt;                      received in this step is different                      to the &lt;TRANSACTION_ID_IA&gt;                      in Step 1</li> <li>• &lt;TRANSACTION_ID_SIGNED_                      2&gt; received in this step is                      different to the                      &lt;TRANSACTION_ID_SIGNED_I                      A&gt; in Step 1</li> <li>• &lt;SERVER_CHALLENGE_2&gt;                      received in this step is different                      to the                      &lt;SERVER_CHALLENGE&gt; in                      Step 1.</li> </ul>	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, #EUIICC_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_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → 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
--	--	--	--	---

**TC\_SM-DP+\_ES9+.InitiateAuthenticationFRP**

This test case is defined as FFS and not applicable for this version of test specification.  
 4.3.12.2.2

**TC\_SM-DP+\_ES9+.InitiateAuthenticationBRP**

4.3.12.2.2

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 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**

**TC\_SM-DP+\_ES9+.GetBoundProfilePackageNIST**

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>• SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST</li> <li>• PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1</li> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• There have been no previous attempts to download the pending profile.</li> </ul>

**Test Sequence #01 Nominal: Using S-ENC and S-MAC without Confirmation Code**

Initial Conditions	
Entity	Description of the initial condition
SM-DP+	<ul style="list-style-type: none"> <li>• PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is loaded as an Unprotected Profile Package.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+. Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP( <S_TRANSACTION_ID>,	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_SK)  • Verify that <TRANSACTION_ID_GBPP> matches <S_TRANSACTION_ID>  MTD_TEST_ES8+_GET_BPP_SK	RQ25_001 RQ25_002 RQ25_004 RQ25_006 RQ25_010 RQ25_011 RQ25_012

		#PREP_DOWNLOAD_RESP))	(#R_GET_BPP_RESP_OP1_SK, <S_MAC>, <S_ENC>, #SMDP_METADATA_OP_PROF1)	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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is loaded as an Unprotected Profile Package.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> </ul>

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_RES P_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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_CC		
1	S_LPA → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(	MTD_HTTP_RESP(#R_GET_BP P_RESP_OP1_PPK)	RQ25_001 RQ25_002 RQ25_005 RQ25_006

		<p>&lt;S_TRANSACTION_ID&gt;, #PREP_DOWNLOAD_RESP_CC))</p>	<ul style="list-style-type: none"> <li>Verify that &lt;TRANSACTION_ID_GBPP&gt; matches &lt;S_TRANSACTION_ID&gt;</li> </ul> <p>MTD_TEST_ES8+_GET_BPP_PPK (#R_GET_BPP_RESP_OP1_PPK, &lt;S_MAC&gt;, &lt;S_ENC&gt;, &lt;PPK_MAC&gt;, &lt;PPK_ENC&gt;, #SMDP_METADATA_OP_PROF1)</p>	<p>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</p>
--	--	---	--	---



				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 LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1_2_SEG is loaded as an Unprotected Profile Package.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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 LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with</li> </ul>

	<p>#SMDP_METADATA_OP_PROF1_2_SEG is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</p> <ul style="list-style-type: none"> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>
--	--

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.

**TC\_SM-DP+\_ES9+.GetBoundProfilePackageFRP**

This test case is defined as FFS and not applicable for this version of test specification.  
 4.3.13.2.2

**TC\_SM-DP+\_ES9+.GetBoundProfilePackageBRP**

4 General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for BRP.</li> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1.</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>There have been no previous attempts to download the pending profile.</li> </ul>

**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 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 keys and certificates SHALL be based on BrainpoolP256r1.

**TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_ReuseOTPK\_NIST**

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST.</li> </ul>

	<ul style="list-style-type: none"> <li>• PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>• The EID is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>
--	--

**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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>• PROFILE_OPERATIONAL1 is loaded as an Unprotected Profile Package.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> <li>• There have been no previous attempts to download the pending profile.</li> </ul>

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_LPA <sub>d</sub> → 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is loaded as an Unprotected Profile Package.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempts to download the pending profile.</li> </ul>

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_LPA <sub>d</sub> → 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_BP P_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_PROF 1)  • 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> <li>There has been no previous attempts to download the pending profile.</li> </ul>

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_LPA <sub>d</sub> → 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 LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> <li>There has been no previous attempts to download the pending profile.</li> </ul>

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_LPA → 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 LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is loaded as an Unprotected Profile Package.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempt to download the pending profile.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1			PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CANCEL_SESSION_S K 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))	<p>MTD_HTTP_RESP( #R_ERROR_8_2_3_7)</p> <p>OR</p> <p>MTD_HTTP_RESP( #R_GET_BPP_RESP_OP1_SK)</p> <ul style="list-style-type: none"> <li>Verify that &lt;TRANSACTION_ID_GBPP&gt; matches &lt;S_TRANSACTION_ID&gt;</li> </ul> <p>MTD_TEST_ES8+_GET_BPP_S K (#R_GET_BPP_RESP_OP1_SK, &lt;S_MAC&gt;,&lt;S_ENC&gt;, #SMDP_METADATA_OP_PROF1 )</p> <ul style="list-style-type: none"> <li>Verify that &lt;OTPK_SM_DP+_ECKA&gt; in #INIT_SC_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1</li> </ul>	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 LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is loaded as an Unprotected Profile Package.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> </ul>

	<ul style="list-style-type: none"> <li>• There have been no previous attempt to download the pending profile.</li> </ul>
--	--

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+	<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP_N EW_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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempt to download the pending profile.</li> </ul>

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_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP( <S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP_ NEW_OTPK))	<p>MTD_HTTP_RESP( #R_ERROR_8_2_3_7)</p> <p>OR</p> <p>MTD_HTTP_RESP(#R_GET_BP P_RESP_OP1_PPK)</p> <ul style="list-style-type: none"> <li>Verify that &lt;TRANSACTION_ID_GBPP&gt; matches &lt;S_TRANSACTION_ID&gt;</li> </ul> <p>MTD_TEST_ES8+_GET_BPP_PPK (#R_GET_BPP_RESP_OP1_PPK, &lt;S_MAC&gt;, &lt;S_ENC&gt;, &lt;PPK_MAC&gt;, &lt;PPK_ENC&gt;, #SMDP_METADATA_OP_PROF1)</p> <ul style="list-style-type: none"> <li>Verify that &lt;OTPK_SM_DP+_ECKA&gt; in #INIT_SC_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1</li> </ul>	<p>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 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</p>

				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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempt to download the pending profile.</li> </ul>

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_LPA <sub>d</sub> →	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS,	MTD_HTTP_RESP( #R_ERROR_ 8_2_3_7)	RQ25_001 RQ25_002



	SM-DP+	<pre>#PATH_GET_BPP, MTD_GET_BPP(   &lt;S_TRANSACTION_ID&gt;, #PREP_DOWNLOAD_RESP_ NEW_OTPK_CC))</pre>	<p>OR</p> <pre>MTD_HTTP_RESP(#R_GET_BP P_RESP_OP1_PPK)</pre> <ul style="list-style-type: none"> <li>Verify that &lt;TRANSACTION_ID_GBPP&gt; matches &lt;S_TRANSACTION_ID&gt;</li> </ul> <pre>MTD_TEST_ES8+_GET_BPP_P PK (#R_GET_BPP_RESP_OP1_PPK , &lt;S_MAC&gt;, &lt;S_ENC&gt;, &lt;PPK_MAC&gt;, &lt;PPK_ENC&gt;, #SMDP_METADATA_OP_PROF 1)</pre> <ul style="list-style-type: none"> <li>Verify that &lt;OTPK_SM_DP+_ECKA&gt; in #INIT_SC_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1</li> </ul>	<p>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</p>
--	--------	---	--	---

				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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and PPK_MAC&gt;.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> <li>The SM-DP+ is configured with two retries allowed for the receipt of a valid Confirmation Code.</li> <li>There have been no previous attempts to download the pending profile.</li> </ul>

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_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP( <S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP_C C))	MTD_HTTP_RESP( #R_GET_BPP_RESP_OP1_PPK)  • Verify that <TRANSACTION_ID_GBPP> matches <S_TRANSACTION_ID>  MTD_TEST_ES8+_GET_BPP_PP K (#R_GET_BPP_RESP_OP1_PPK,	RQ25_001 RQ25_002 RQ25_004 RQ25_006 RQ25_010 RQ25_011 RQ25_012 RQ25_013 RQ25_015 RQ26_018

			<S_MAC>, <S_ENC>, <PPK_MAC>, <PPK_ENC>, #SMDP_METADATA_OP_PROF1 )	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_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_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
--	--	--	--	--

**VOID**

4.3.13.2.5 **VOID**

4.3.13.2.6 **TC\_SM-**

4.3.13.2.7 **DP+\_ES9+.GetBoundProfilePackage\_RetryCases\_DifferentOTPK\_NIST**

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1.</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>The EID is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

**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 LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is loaded as an Unprotected Profile Package.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempt to download the pending profile.</li> </ul>

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 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_SK) • Verify that <TRANSACTION_ID_GBPP> matches <S_TRANSACTION_ID> MTD_TEST_ES8+_GET_BPP_S K (#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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is loaded as an Unprotected Profile Package.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempt to download the pending profile.</li> </ul>

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_LPA <sub>d</sub> → 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) <ul style="list-style-type: none"> <li>Verify that &lt;TRANSACTION_ID_GBP&gt; matches &lt;S_TRANSACTION_ID&gt;</li> </ul> MTD_TEST_ES8+_GET_BPP_SK (#R_GET_BPP_RESP_OP1_SK, <S_MAC>, <S_ENC>, #SMDP_METADATA_OP_	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

		<p>PROF1)</p> <ul style="list-style-type: none"> <li>Verify that &lt;OTPK_SM_DP+_ECKA&gt; in #INIT_SC_PROF1 is different from the value previously received in the GetBoundProfilePackage response in step 4 of the procedure in IC1</li> </ul>	<p>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</p>
--	--	---	---

				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 LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempt to download the pending profile.</li> </ul>

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_LPA → 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code #CONFIRMATION_CODE1 associated to PROFILE_OPERATIONAL1 is provided by the Operator to the SM-DP+.</li> <li>There have been no previous attempt to download the pending profile.</li> </ul>

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_LPA <sub>d</sub> → 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
--	--	--	--	--

**VOID**

**VOID**

4.3.13.2.8

**TC\_SM-DP+\_ES9+.GetBoundProfilePackage\_ErrorCasesNIST**

4.3.13.2.9

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST.</li> <li>PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>The EID is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul> <p>There have been no previous attempts to download the pending profile.</p>

**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_LPA → 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_LPA → 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

**4.3.13.2.11 Test Sequence #06 VOID**

4.3.13.2.12 VOID

VOID

**4.3.14 ES9+ (LPA -- SM-DP+): AuthenticateClient**

**4.3.14.1 Conformance Requirements**

**References**

GSM RSP Technical Specification [2]

**Requirements**

- RQ26\_033

- RQ31\_025, RQ31\_058, RQ31\_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**

**TC\_SM-DP+\_ES9+.AuthenticateClientNIST**

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>• SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST and #CERT_SM_DPpb_ECDSA for NIST.</li> <li>• PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> </ul>

**Test Sequence #01 Nominal for Default SM-DP+ Address Use Case without Confirmation Code**

Initial Conditions	
Entity	Description of the initial condition
SM-DP+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	S_LPAAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH,	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)	



		<p>MTD_INITIATE_AUTHENTICATION(                  #S_EUICC_CHALLENGE,                  #S_EUICC_INFO1,                  #IUT_SM_DP_ADDRESS))</p>		
1	S_LPAd → SM-DP+	<p>MTD_HTTP_REQ(                  #IUT_SM_DP_ADDRESS,                  #PATH_AUTH_CLIENT,                  MTD_AUTHENTICATE_CLIENT(                  &lt;S_TRANSACTION_ID&gt;,                  #AUTH_SERVER_RESP_DEF_DP_UC_OK))</p>	<p>MTD_HTTP_RESP(#R_AUTH_CLIENT_OK)</p> <ul style="list-style-type: none"> <li>• Verify that &lt;TRANSACTION_ID_AC&gt; matches &lt;S_TRANSACTION_ID&gt;</li> <li>• Verify the validity of the smdpSignature2 &lt;SMDP_SIGNATURE2&gt; using the #PK_SM_DPpb_ECDSA</li> <li>• Verify that the SM-DP+ Address in the #SMDP_METADATA_OP_PROF1 matches #IUT_SM_DP_ADDRESS.</li> <li>• Verify that &lt;TRANSACTION_ID_SIGNED_AC&gt; matches &lt;S_TRANSACTION_ID&gt;</li> </ul>	<p>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</p>

**Test Sequence #02 Nominal for Default SM-DP+ Address Use Case with Confirmation Code**

Initial Conditions	
Entity	Description of the initial condition
SM-DP+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code #CONFIRMATION_CODE1 is provided by the Operator to the SM-DP+.</li> </ul>

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 the SM-DP+ Address in the #SMDP_METADATA_OP_PROF 1 matches #IUT_SM_DP_ADDRESS. • Verify that <TRANSACTION_ID_SIGNED_A C> 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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → 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>	RQ31_025 RQ31_058 RQ31_059 RQ31_060 RQ31_080 RQ31_081 RQ31_082 RQ31_091 RQ31_092

			using the #PK_SM_DPpb_ECDSA • Verify that the SM-DP+ Address in the #SMDP_METADATA_OP_PR OF1 matches #IUT_SM_DP_ADDRESS. • Verify that <TRANSACTION_ID_SIGNE D_AC> matches <S_TRANSACTION_ID>	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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 in the 'Released' state with a MatchingID equal to &lt;MATCHING_ID_EVENT&gt;.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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_AUTHENTICAT ION( #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_CLIEN T( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_SMD S_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLI ENT_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 the SM-DP+ Address in the #SMDP_METADATA_OP_PROF1 matches #IUT_SM_DP_ADDRESS. • 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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 in the 'Released' state with a MatchingID equal to &lt;MATCHING_ID_EVENT&gt;.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code #CONFIRMATION_CODE1 is provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	S_LPA → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITI ATE_AUTH_OK)	
1	S_LPA → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_SMDS_U C_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 the SM-DP+ Address in the #SMDP_METADATA_OP_P ROF1 matches #IUT_SM_DP_ADDRESS. • Verify that <TRANSACTION_ID_SIGNE D_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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING_ID_1.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1,	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	

		#IUT_SM_DP_ADDRESS))		
1	S_LPAd → SM-DP+	<p>MTD_HTTP_REQ(                  #IUT_SM_DP_ADDRESS,                  #PATH_AUTH_CLIENT,                  MTD_AUTHENTICATE_CLIENT(                  &lt;S_TRANSACTION_ID&gt;,                  #AUTH_SERVER_RESP_ACT_                  CODE_UC_OK))</p>	<p>MTD_HTTP_RESP(#R_AUTH_                  CLIENT_OK)</p> <ul style="list-style-type: none"> <li>• Verify that                      &lt;TRANSACTION_ID_AC&gt;                      matches                      &lt;S_TRANSACTION_ID&gt;</li> <li>• Verify the validity of the                      smdpSignature2                      &lt;SMDP_SIGNATURE2&gt; using                      the #PK_SM_DPpb_ECDSA</li> <li>• Verify that the SM-DP+                      Address in the                      #SMDP_METADATA_OP_PRO                      F1 matches                      #IUT_SM_DP_ADDRESS.</li> <li>• Verify that                      &lt;TRANSACTION_ID_SIGNED_                      AC&gt; matches                      &lt;S_TRANSACTION_ID&gt;</li> </ul>	<p>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</p>



**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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING_ID_1.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code #CONFIRMATION_CODE1 is provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	S_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_ACT_CO DE_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 the SM-DP+ Address in the #SMDP_METADATA_OP_P ROF1 matches #IUT_SM_DP_ADDRESS. • Verify that <TRANSACTION_ID_SIGNE D_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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING_ID_1.</li> <li>• EID #EID1 is not known to the SM-DP+ and is not associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING_ID_1.</li> <li>• EID #EID1 is not known to the SM-DP+ and is not associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code #CONFIRMATION_CODE1 is provided by the</li> </ul>

	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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• EID #EID1 is known to the SM-DP+ and is associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

This test sequence SHALL be the same as the Test Sequence #01 defined in this section except that #AUTH\_SERVER\_RESP\_MATCHING\_ID\_OMITTED shall be used in MTD\_AUTHENTICATE\_CLIENT instead of #AUTH\_SERVER\_RESP\_DEF\_DP\_UC\_OK.

**TC\_SM-DP+\_ES9+.AuthenticateClientNIST\_ErrorCases**

4.3.14.2.2

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>• SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST and #CERT_SM_DPpb_ECDSA for NIST.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+ for the pending profile.</li> </ul>

**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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	S_LPA →	MTD_HTTP_REQ(	MTD_HTTP_RESP(	

	SM-DP+	#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH,  MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	#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_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_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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATIO N( #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_AUTHENTICATIO N( #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_AUTHENTICATIO N( #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_AUTHENTICATIO N( #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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATIO N( #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_SUB_SN))	MTD_HTTP_RESP( #R_ERROR_8_1_3_6_3)	RQ31_061 RQ45_028 RQ56_030 RQ56_038 RQ56_041



	#AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_3))		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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATIO N( #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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATIO N( #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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATI ON( #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_D	MTD_HTTP_RESP( #R_ERROR_8_11_1_3_9)	RQ26_033 RQ31_061 RQ45_028 RQ56_030 RQ56_038 RQ56_041 RQ62_001

	P_UC_8_11_1_3_9))		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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is not in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 igned with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	S_LPA → 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_O K)	
1	S_LPA → 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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 igned with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATIO N( #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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATIO N( #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>, 	MTD_HTTP_RESP( #R_ERROR_8_10_1_3_9)	RQ31_061 RQ56_038 RQ56_041 RQ62_001 RQ62_002

		#AUTH_SERVER_RESP_DEF_DP_UC_8_10_1_3_9))		
--	--	--	--	--

**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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>

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_AUTHENTICATIO N( #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_CO DE_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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_1.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

	<ul style="list-style-type: none"> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>
--	---

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_O K)	
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_6_3_ 8)	RQ31_061 RQ41_006 RQ41_007 RQ41_008 RQ56_033 RQ56_038 RQ56_041 RQ62_001 RQ62_002
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+			
3	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_O K)	
4	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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 in the 'Released' state with a MatchingID equal to &lt;MATCHING_ID_EVENT&gt;.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected</li> </ul>

	Profile Package using <PPK_ENC> and <PPK_MAC>. <ul style="list-style-type: none"> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> </ul>
--	--

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_O K)	
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_6_3_8 )	RQ31_061 RQ41_006 RQ41_007 RQ41_008 RQ56_033 RQ56_038 RQ56_041 RQ62_001 RQ62_002
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+			
3	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_O K)	
4	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 (Subject Code 8.1.1 Reason Code 3.8)**

Initial Conditions	
Entity	Description of the initial condition
SM-DP+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

	<ul style="list-style-type: none"> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• EID #EID2 is not known to the SM-DP+ and is not associated to PROFILE_OPERATIONAL1.</li> </ul>
--	--

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_AUTHENTICATIO N( #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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL3 configured with #SMDP_METADATA_OP_PROF3 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• Pending Profile PROFILE_OPERATIONAL3 is in the 'Released' state, with an empty MatchingID.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL3.</li> </ul>

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_AUTHENTICATIO N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_ OK)	



1	S_LPA → 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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There have been no previous attempts to download the pending profile.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• The SM-DP+ has expired Profile download order.</li> </ul> <p>NOTE: This is expected to be done through proprietary means.</p>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+			
IC2	S_LPA → 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_LPA → 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_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+	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC TC_SM-DP+_ES9+.AuthenticateClientBRP.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• All previous attempts to download the pending Profile have been unsuccessful.</li> <li>• 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.</li> </ul>

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_O K)	
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_O K))	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

4.3.14.2.3  
**Test Sequence #18 VOID**

**TC\_SM-DP+\_ES9+.AuthenticateClientFRP**

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

VOID

**TC\_SM-DP+\_ES9+.AuthenticateClientBRP**

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for BRP.</li> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>There have been no previous attempts to download the pending profile.</li> </ul>

**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 keys and certificates SHALL be based on BrainpoolP256r1.

**TC\_SM-DP+\_ES9+.AuthenticateClient\_RetryCases\_Reuse\_OTPK**

4.3.14.2.6

**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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with #MATCHING_ID_EMPTY.</li> <li>EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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_AUTHENTICATIO N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	

1	S_LPAd → SM-DP+	<p>MTD_HTTP_REQ(                  #IUT_SM_DP_ADDRESS,                  #PATH_AUTH_CLIENT,                  MTD_AUTHENTICATE_CLIENT(                  &lt;S_TRANSACTION_ID&gt;,                  #AUTH_SERVER_RESP_DEF_DP_UC_OK))</p>	<p>MTD_HTTP_RESP(                  #R_AUTH_CLIENT_RETRY_OK)                  • Verify that                  &lt;TRANSACTION_ID_AC&gt;                  matches                  &lt;S_TRANSACTION_ID&gt;                  • Verify the validity of the                  smdpSignature2                  &lt;SMDP_SIGNATURE2&gt;                  using the                  #PK_SM_DPauth_ECDSA                  • Verify that the SM-DP+                  Address in the                  #SMDP_METADATA_OP_P                  ROF1 matches                  #IUT_SM_DP_ADDRESS.                  • Verify that                  &lt;TRANSACTION_ID_SIGNE                  D_AC&gt; matches                  &lt;S_TRANSACTION_ID&gt;</p>	<p>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</p>
---	--------------------	--	--	---

**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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

	<ul style="list-style-type: none"> <li>• Pending Profile PROFILE_OPERATIONAL1 in the 'Released' state with a MatchingID equal to &lt;MATCHING_ID_EVENT&gt;.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>
--	--

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_AUTHENTICATIO N( #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_U C_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 the SM-DP+ Address in the #SMDP_METADATA_OP_P ROF1 matches #IUT_SM_DP_ADDRESS. • Verify that <TRANSACTION_ID_SIGNE D_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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING_ID_1.</li> <li>EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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_LPA → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPA → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_ACT_CO DE_UC_OK))	MTD_HTTP_RESP(#R_AUT H_CLIENT_RETRY_OK) • Verify that <TRANSACTION_ID_AC> matches <S_TRANSACTION_ID> • Verify the validity of the smdpSignature2 <SMDP_SIGNATURE2> using the	RQ31_058 RQ31_059 RQ31_060 RQ31_080 RQ31_081 RQ31_082 RQ31_091 RQ31_092 RQ31_093 RQ31_094 RQ31_095

			#PK_SM_DPauth_ECDSA • Verify that the SM-DP+ Address in the #SMDP_METADATA_OP_P ROF1 matches #IUT_SM_DP_ADDRESS. • Verify that <TRANSACTION_ID_SIGNED_AC> matches <S_TRANSACTION_ID>	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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with the MatchingID set as an Activation Code Token with the value #MATCHING_ID_1.</li> <li>EID #EID1 is not known to the SM-DP+ and not associated to PROFILE_OPERATIONAL1.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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+	<ul style="list-style-type: none"> <li>• SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>• The EID is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• There have been no previous attempts to download pending Profile PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

#### **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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>
--------	---

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_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_OTHER_Inst1))	#R_HTTP_204_OK	RQ25_016 RQ25_018 RQ25_023 RQ25_024 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
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_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
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_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
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_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
--	--	--	--	--

**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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1_EN is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC_EN		
1	S_LPA <sub>d</sub> → 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_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
--	--	--	--	--

**Test Sequence #03 Error: Invalid Transaction ID**

Initial Conditions	
Entity	Description of the initial condition
SM-DP+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA → 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_1 RQ31_178 RQ35_017 RQ35_019 RQ35_022 RQ56_042 RQ56_042_1 RQ56_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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF( #S_PN_PIR_INCORRECT_INPU T_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_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 #05 Error: PIR Error Reason – invalid signature**

Initial Conditions	
Entity	Description of the initial condition
SM-DP+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPAAd → 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;..</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF( #S_PN_PIR_UNSUPPORTED_C RT))	#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 #07 Error: PIR Error Reason – unsupported Remote Operation Type**

Initial Conditions	
Entity	Description of the initial condition
SM-DP+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF( #S_PN_PIR_UNSUP_REMOTE_O P_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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF( #S_PN_PIR_UNSUP_PROFILE_	#R_HTTP_204_OK	RQ25_016 RQ25_018 RQ25_023 RQ25_024 RQ25_025 RQ25_026

		CLASS))		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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA → 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

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_EXI	#R_HTTP_204_OK	RQ25_016 RQ25_018 RQ25_023 RQ25_024 RQ25_025

		STS))		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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA → 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA → 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF( #S_PN_PIR_PE_PROCESSING_	#R_HTTP_204_OK	RQ25_016 RQ25_018 RQ25_023 RQ25_024 RQ25_025

		ERROR))		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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA → 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF( #S_PN_PIR_TEST_PROFILE_IN VALID_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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF( #S_PN_PIR_TEST_PROFILE_IN VALID_NAA_KEY))	#R_HTTP_204_OK	RQ25_016 RQ25_018 RQ25_023 RQ25_024

		#S_PN_PIR_PPR_NOT_ALLOW ED))		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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> </ul>

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_ERRO R))	#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



**TC\_SM-DP+\_ES9+\_HandleNotificationFRP**

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

**TC\_SM-DP+\_ES9+\_HandleNotificationBRP**

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for BRP.</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>The EID is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>There have been no previous attempts to download pending Profile PROFILE_OPERATIONAL1.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

**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 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**

**TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientNIST**

General Initial Conditions	
Entity	Description of the general initial condition
4.3.16.2.1  SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST.</li> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1.</li> <li>Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>The EID is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>There have been no previous attempts to download the pending profile.</li> </ul>

**Test Sequence #01 Nominal: End User Rejection after Authenticate Client**

The purpose of this test is to verify that the LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		ROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is configured with #SMDP_METADATA_OP_PROF1_PPR2.</li> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1_PPR2 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA <sub>d</sub> → 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → 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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA → 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

**TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageNIST**

4.3.16.2.2

General Initial Conditions	
Entity	Description of the general initial condition
SM-DP+	<ul style="list-style-type: none"> <li>SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST.</li> </ul>

**Test Sequence #01 Nominal: End User Rejection after GetBoundProfilePackage**

The purpose of this test is to verify that the LPA 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC			
1	S_LPA → 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_SUCCESS)	RQ31_118 RQ31_119 RQ31_121 RQ31_122 RQ31_125

		#CS_RESP_OK_EU_REJ))		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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(	MTD_HTTP_RESP( #R_SUCCESS)	RQ31_118 RQ31_119 RQ31_121 RQ31_122

		<S_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))		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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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 LPA<sub>d</sub> 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<sub>+</sub>.

Initial Conditions	
Entity	Description of the initial condition
SM-DP <sub>+</sub>	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 is configured with #SMDP_METADATA_OP_PROF1_PPR2.</li> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1_PPR2 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP<sub>+</sub>.</li> </ul>

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_LPA <sub>d</sub> → SM-DP <sub>+</sub>	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION( <S_TRANSACTION_ID>, #CS_RESP_OK_PPR_NOT_ALLOWE	MTD_HTTP_RESP( #R_SUCCESS)	RQ31_118 RQ31_119 RQ31_121 RQ31_122 RQ31_125 RQ31_126

		D))		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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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_MISMAT	MTD_HTTP_RESP( #R_SUCCESS)	RQ31_118 RQ31_119 RQ31_121 RQ31_122 RQ31_125 RQ31_126

		CH))		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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION( <S_TRANSACTION_ID>, #CS_RESP_OK_L_BPP_EXE_ER	MTD_HTTP_RESP( #R_SUCCESS)	RQ31_118 RQ31_119 RQ31_121 RQ31_122 RQ31_125 RQ31_126

		ROR))		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 LPA<sub>d</sub> 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<sub>+</sub>.

Initial Conditions	
Entity	Description of the initial condition
SM-DP <sub>+</sub>	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP<sub>+</sub>.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → SM-DP <sub>+</sub>	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION( <S_TRANSACTION_ID>,	MTD_HTTP_RESP( #R_SUCCESS)	RQ31_118 RQ31_119 RQ31_121 RQ31_122 RQ31_125



		#CS_RESP_OK_UNDEFINED))		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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> →	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS,	MTD_HTTP_RESP( 	RQ31_118 RQ31_119

	SM-DP+	#PATH_CANCEL_SESSION, MTD_CANCEL_SESSION( <INVALID_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))	#R_ERROR_8_10_1_3_9)	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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

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 LPA<sub>d</sub> 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+	<ul style="list-style-type: none"> <li>PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul>

Step	Direction	Sequence / Description	Expected result	REQ
IC1		PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC		
1	S_LPA <sub>d</sub> → 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
4.3.16.2.3		PROC_ES9+_VERIFY_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC		RQ57_114_1

**4.3.16.2.4 TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientFRP**

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

**TC\_SM-DP+\_ES9+.CancelSession\_After\_GetBoundProfilePackageFRP**

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

**TC\_SM-DP+\_ES9+.CancelSession\_After\_AuthenticateClientBRP**

General Initial Conditions	
Entity	Description of the general initial condition
4.3.16.2.5  SM-DP+	<ul style="list-style-type: none"> <li>• SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for BRP.</li> <li>• PROFILE_OPERATIONAL1 configured with #SMDP_METADATA_OP_PROF1.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>• The EID is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• There have been no previous attempts to download the pending profile.</li> </ul>

**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 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 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+	<ul style="list-style-type: none"> <li>• SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for BRP.</li> </ul>

**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 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 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
  - 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
  - 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
  - CERT\_CLIENT\_TLS = #CERT\_SM\_DP\_TLS for NIST
- SERVER = S\_SM-DS
  - 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
  - CERT\_CLIENT\_TLS = #CERT\_SM\_DP\_TLS for BRP

- SERVER = S\_SM-DS
  - CERT\_S\_SERVER\_TLS = #CERT\_S\_SM\_DS\_TLS for BRP

## 4.4 LPA 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.
LPAAd	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	LPAAd → 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+ → LPAAd	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\_LPAAd\_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.
LPAAd	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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_8_1_3_8)	LPAAd aborts AddProfile procedure	RQ31_034 RQ56_008 RQ56_011
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_8_2_3_1)	LPAAd aborts AddProfile procedure	RQ31_035 RQ56_008 RQ56_011
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_8_3_3_1)	LPAAd aborts AddProfile procedure	RQ56_008 RQ56_011

2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIM EOUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_8_4_3_7)	LPAAd aborts AddProfile procedure	RQ31_036 RQ56_008 RQ56_011
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIM EOUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #INITIATE_AUTH_INV_CERT)	LPAAd aborts AddProfile procedure	RQ31_052
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_SIGN)	LPAAd aborts AddProfile procedure	RQ31_052
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_SMDP+_ADDRESS)	LPAAd informs the S_EndUser and aborts the AddProfile procedure	RQ31_045
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_CI)	LPAAd aborts AddProfile procedure	RQ31_052
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIMEOUT in Annex F.	RQ31_052

**Test Sequence #09 Error: Invalid SM-DP+ OID**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	Add Profile operation is initiated, #ACTIVATION_CODE_2 is provided to the LPAAd 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #INITIATE_AUTH_INV_OID)	LPAAd informs the S_EndUser and aborts the AddProfile procedure	RQ31_075
2	LPAAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_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**

**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. Request for Authenticated Confirmation, if not requested before and not aborted.				

**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	LPA → S_EndUser	LPA requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE1 is provided by manual entry.	
1	LPA → 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+ → LPA	MTD_HTTP_RESP(#GET_BPP_OK)	No error, see NOTE.	RQ56_027
NOTE: The LPA MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. Request for Authenticated Confirmation, if not requested before and not aborted.				

**Test Sequence #03 Nominal: Get BPP using PPK-ENC and PPK-MAC without Confirmation Code**

Initial Conditions	
Entity	Description of the initial condition
LPA	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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#GET_BPP_OK_PPK)	No error, see NOTE.	RQ56_027
NOTE: The LPAAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. Request for Authenticated Confirmation, if not requested before and not aborted.				

**Test Sequence #04 Nominal: Get BPP using PPK-ENC and PPK-MAC with Confirmation Code**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	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	LPAAd → S_EndUser	LPAAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE1 is provided by manual entry.	
1	LPAAd → 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_C C)) Verify if: • <S_TRANSACTION_ID> is the same as in #R_PREP_DOWNLOAD_WITH_C C	RQ31_112 RQ31_113 RQ31_141 RQ31_148_2 RQ31_146 RQ31_147 RQ56_015 RQ56_024 RQ56_026



			<ul style="list-style-type: none"> <li>• &lt;EUICC_SIGNATURE2&gt; using the #PK_EUICC_ECDSA</li> <li>• &lt;S_HASHED_CC&gt; = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE1, &lt;S_TRANSACTION_ID&gt;)</li> </ul>	
2	S_SM-DP+ → LPAAd	MTD_HTTP_RESP(#GET_BPP_OK_PPK)	No error, see NOTE.	RQ56_027
<p>NOTE: The LPAAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. Request for Authenticated Confirmation, if not requested before and not aborted.</p>				

### TC\_LPAAd\_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
LPAAd	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	LPAAd → S_EndUser	LPAAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.	
IC5	LPAAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ( <ul style="list-style-type: none"> <li>#TEST_DP_ADDRESS1,</li> <li>#PATH_GET_BPP,</li> <li>MTD_GET_BPP(&lt;S_TRANSACTION_ID&gt;,</li> <li>#R_PREP_DOWNLOAD_WITH_CC))</li> </ul> Verify if: <ul style="list-style-type: none"> <li>&lt;S_HASHED_CC&gt; = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE2, &lt;S_TRANSACTION_ID&gt;)</li> </ul>	
1	S_SM-DP+	MTD_HTTP_RESP(#R_ERROR)	Continue to step 2	RQ31_148_2

	→ LPAAd	_8_2_7_3_8)		RQ56_022
2	S_SM-DP+ closes TLS session (unless ,LPAAd 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	LPAAd → S_EndUser	LPAAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE1 is provided by manual entry.	RQ31_148_3 RQ56_022
7	LPAAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	<p>MTD_HTTP_REQ(                  #TEST_DP_ADDRESS1,                  #PATH_GET_BPP,                  MTD_GET_BPP(&lt;S_TRANSACTION_ID&gt;,                  #R_PREP_DOWNLOAD_WITH_C C))</p> <p>Verify if:</p> <ul style="list-style-type: none"> <li>• If &lt;S_TRANSACTION_ID&gt; is the same as in #R_PREP_DOWNLOAD_WITH_C C</li> <li>• &lt;EUICC_SIGNATURE2&gt; using the #PK_EUICC_ECDSA • &lt;S_HASHED_CC&gt; = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE1, &lt;S_TRANSACTION_ID&gt;)</li> </ul>	RQ31_148_3 RQ56_022 RQ56_026
8	S_SM-DP+ → LPAAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error, see NOTE 1.	RQ56_024 RQ56_027
<p><b>NOTE:</b> The LPAAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. Request for Authenticated Confirmation, if not requested before and not aborted.</p>				

**4.4.22.2.3 TC\_LPAAd\_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).
LPAAd	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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_6_1)	LPAAd aborts AddProfile procedure NOTE: The LPAAd 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_2_3_7)	LPAAd aborts AddProfile procedure NOTE: the LPAAd 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_10_1_3_9)	LPAAd aborts AddProfile procedure NOTE: the LPAAd 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	LPA <sub>d</sub> → 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+ → LPA <sub>d</sub>	MTD_HTTP_RESP( #R_ERROR_8_2_7_2_2)	LPA <sub>d</sub> aborts AddProfile procedure NOTE: the LPA <sub>d</sub> 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	LPA <sub>d</sub> → 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+ → LPA <sub>d</sub>	MTD_HTTP_RESP( #R_ERROR_8_8_5_4_10)	LPA <sub>d</sub> aborts AddProfile procedure NOTE: The LPA <sub>d</sub> 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
LPA <sub>d</sub>	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	LPAAd → S_EndUser	LPAAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.	
IC5	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_2_7_3_8)	LPAAd aborts AddProfile procedure NOTE: The LPAAd 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
LPAAd	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	LPAAd → S_EndUser	LPAAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.	
IC5	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_2_7_6_4)	LPAAd aborts AddProfile procedure The LPAAd 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\_LPAAd\_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
LPAAd	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	LPAAd → S_SM-DP+	Send ES9+.InitiateAuthentication	MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH,	

		method	<p>MTD_INITIATE_AUTHENTICATIO                  N(                  &lt;EUICC_CHALLENGE&gt;,                  #R_EUICC_INFO1,                  #TEST_DP_ADDRESS1))                  • Extract &lt;EUICC_CHALLENGE&gt;</p>	
1	S_SM-DP+ → LPAAd	MTD_HTTP_RESP(#INITIATE_ AUTH_OK)	<p>MTD_HTTP_REQ(                  #TEST_DP_ADDRESS1,                  #PATH_AUTH_CLIENT,                  MTD_AUTHENTICATE_CLIENT(&lt;                  S_TRANSACTION_ID&gt;,                  #R_AUTH_SERVER_MATCH_ID_                  DEV_INFO))</p> <p>Verify:</p> <ul style="list-style-type: none"> <li>• if                      #R_AUTH_SERVER_MATCH_ID_                      DEV_INFO used with the                      #MATCHING_ID_1</li> <li>• If &lt;S_TRANSACTION_ID&gt; is the                      same as in #INITIATE_AUTH_OK</li> <li>• &lt;EUICC_SIGNATURE1&gt; using                      the #PK_EUICC_ECDSA</li> <li>• if &lt;S_SMDP_CHALLENGE&gt;                      present in the                      #R_AUTH_SERVER_MATCH_ID_                      DEV_INFO is the same as in                      &lt;S_SMDP_SIGNED1&gt; present in                      #INITIATE_AUTH_OK</li> <li>• 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]</li> <li>- 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)</li> <li>- if O_D_GSM_GERAN then                      gsmSupportedRelease is set to the                      highest release as defined in                      #IUT_GSM_GERAN_REL.</li> <li>- if O_D_UMTS_UTRAN then                      utranSupportedRelease is set to                      the highest release as defined in                      #IUT_UMTS_UTRAN_REL.</li> <li>- if O_D_CDMA2000_1X then</li> </ul>	<p>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</p>

			<p>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 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.</p>	<p>RQ65_009                  RQ65_019                  RQ65_022</p>
2	S_SM-DP+ → LPAAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK)	No Error	<p>RQ31_073                  RQ31_095                  RQ56_037                  RQ56_040                  RQ56_041_1                  RQ56_041_2</p>

**Test Sequence #02 Nominal: Authenticate Client with Confirmation Code**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	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	LPAAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1))  • Extract <EUICC_CHALLENGE>	
1	S_SM-DP+ → LPAAd	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)	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

			<p>- if O_D_GSM_GERAN then  gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL.</p> <p>- if O_D_UMTS_UTRAN then  utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL.</p> <p>- if O_D_CDMA2000_1X then  cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL.</p> <p>- if O_D_CDMA2000_HRPD then  cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL.</p> <p>The value R is either 1, 2 or 3 for Rev 0, A or B respectively.</p> <p>- if O_D_CDMA2000_EHRPD then  cdma2000ehrpdsupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL.</p> <p>- if O_D_LTE then  eutranSupportedRelease is set to the highest release as defined in #IUT_LTE_EUTRAN_REL.</p> <p>- if O_D_NFC_TS26 then  contactlessSupportedRelease is set to the highest release as defined in #IUT_NFC_REL.</p> <p>- if O_D_CRL then  rspCrlSupportedVersion is set to the highest release as defined in #IUT_RSP_VERSION.</p> <p>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.</p>	<p>RQ65_001  RQ65_002  RQ65_003  RQ65_004  RQ65_005  RQ65_007  RQ65_008  RQ65_022</p>
2	S_SM-DP+ → LPAAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK_CC)	No Error	<p>RQ31_073  RQ31_095  RQ56_037  RQ56_040  RQ56_041_1  RQ56_041_2</p>

**Test Sequence #03 Nominal: Authenticate Client with Confirmation Code Retry**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	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	LPAAd → S_EndUser	LPAAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.	
IC5	LPAAd → 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+ → LPAAd	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+ → LPAAd	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+ → LPAAd	MTD_HTTP_RESP(#AUTH_CLIENT_OK_CC)	No Error	RQ31_091

**4.4.23.2.2 TC\_LPAAd\_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.
LPAAd	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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_2_6_1)	LPAAd aborts AddProfile procedure	RQ31_061 RQ56_030 RQ56_038
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_2_6_3)	LPAAd aborts AddProfile procedure	RQ31_061 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_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	LPAAd → 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_DE V_INFO))	
2	S_SM-DP+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_3_6_1)	LPAAd aborts AddProfile procedure	RQ31_061 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIM EOUT 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	LPAAd → 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_DE V_INFO))	
2	S_SM-DP+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_3_6_3)	LPAAd aborts AddProfile procedure	RQ31_061 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIM EOUT 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	LPAAd → 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_DE V_INFO))	
2	S_SM-DP+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_6_1)	LPAAd aborts AddProfile procedure	RQ31_061 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIM EOUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_4_8)	LPAAd aborts AddProfile procedure	RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_11_1_3_9)	LPAAd aborts AddProfile procedure	RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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))	

			#R_AUTH_SERVER_MATCH_ID_DEV_INFO))	
2	S_SM-DP+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_2_1_2)	LPAAd aborts AddProfile procedure	RQ31_083 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_10_1_3_9)	LPAAd aborts AddProfile procedure	RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_2_6_3_8)	LPAAd aborts AddProfile procedure	RQ31_083 RQ31_090 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_1_1_3_8)	LPAAd aborts AddProfile procedure	RQ31_083 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_ERROR_8_2_5_4_3)	LPAAd aborts AddProfile procedure	RQ31_090 RQ31_083 RQ56_030
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#R_ERROR_8_8_5_4_10)	LPAAd aborts AddProfile procedure	RQ31_090 RQ56_030 RQ56_031
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#R_ERROR_8_8_5_6_4)	LPAAd aborts AddProfile procedure	RQ31_085 RQ56_030 RQ56_031_1
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#AUTH_CLIENT_INV_PB_CERT)	LPAAd aborts AddProfile procedure (See NOTE)	RQ31_136 RQ57_031
3	LPAAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIM	RQ31_136 RQ57_031

			EOUT in Annex F.	
NOTE: Before the AddProfile procedure is aborted, the LPAad may request for Authenticated Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Authenticated 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	LPAad → S_SM-DP+	AuthenticateClient	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEVICE_INFO))	
2	S_SM-DP+ → LPAad	MTD_HTTP_RESP(#AUTH_CLIENT_INV_CI)	LPAad aborts AddProfile procedure (See NOTE)	RQ31_136 RQ57_031
3	LPAad → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAad_SESSION_CLOSE_TIMEOUT in Annex F.	RQ31_136 RQ57_031
NOTE: Before the AddProfile procedure is aborted, the LPAad may request for Authenticated Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Authenticated 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	LPAad → 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_DEVICE_INFO))	
2	S_SM-DP+ → LPAad	MTD_HTTP_RESP(#AUTH_CLIENT_INV_SIGN)	LPAad aborts AddProfile procedure (See NOTE)	RQ31_136 RQ57_031
3	LPAad → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAad_SESSION_CLOSE_TIMEOUT in Annex F.	RQ31_136 RQ57_031
NOTE: Before the AddProfile procedure is aborted, the LPAad may request for Authenticated Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Authenticated				

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	LPA <sub>d</sub> → 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+ → LPA <sub>d</sub>	MTD_HTTP_RESP( #AUTH_CLIENT_INV_TRANSACTION_ID)	LPA <sub>d</sub> aborts AddProfile procedure (See NOTE)	RQ31_136 RQ57_031
3	LPA <sub>d</sub> → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPA <sub>d</sub> _SESSION_CLOSE_TIMEOUT in Annex F.	RQ31_136 RQ57_031
NOTE: Before the AddProfile procedure is aborted, the LPA <sub>d</sub> may request for Authenticated Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Authenticated 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, RQ32\_001
- 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

4.4.24.2.1

**4.4.24.2 Test Cases**

**TC\_LPA<sub>d</sub>\_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_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ32_001
2	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
3	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
4	LPAd →	Establish an HTTPs connection if previously closed		

	S_SM-DP+(1)			
5	LPA <sub>d</sub> → S_SM-DP+(1)	Send ES9+.HandleNotification method	<p>MTD_HTTP_REQ(                  #TEST_DP_ADDRESS1,                  #PATH_HANDLE_NOTIF,                  MTD_HANDLE_NOTIF(                  #PENDING_NOTIF_INST1))                  sent within the timeout                  #IUT_LPA<sub>d</sub>_NOTIFICATION_TIMEOUT</p> <p>Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</p>	RQ35_008 RQ35_013 RQ35_014 RQ35_022 RQ35_018 RQ62_003 RQ65_024 RQC3_003
6	S_SM-DP+(1) → LPA <sub>d</sub>	#R_HTTP_204_OK	No error exhibited by the LPA <sub>d</sub>	RQ35_008 RQ35_022 RQ56_042 RQ62_003 RQ62_009 RQ63_005
NOTE 1: The LPA <sub>d</sub> 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 session. NOTE 2: The timeout SHALL start after the PIR is received.				

**Test Sequence #02 Nominal: Successful PIR and Enable Notifications to the Same SM-DP+ Address**

Initial Conditions	
Entity	Description of the initial condition
LPA <sub>d</sub>	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	LPA <sub>d</sub> → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ32_001
2	LPA <sub>d</sub> → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_OK))	RQ31_171 RQ31_176 RQ35_008 RQ35_013

			<ul style="list-style-type: none"> <li>Verify the euiccSignPIR &lt;EUICC_SIGN_PIR&gt; using the #PK_EUICC_ECDSA</li> </ul>	RQ35_017 RQ35_018
3	S_SM-DP+(1) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd. The LPAAd MAY inform the End User of the success status indicated by the Profile Installation Result.	RQ35_008 RQ35_014 RQ35_017
4	S_EndUser → LPAAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL1	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation PROFILE_OPERATIONAL1 is enabled	
5	LPAAd → S_SM-DP+(1)	Establish an HTTPs connection if previously closed		
6	LPAAd → 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_LPAAd_NOTIFICATION_TIMEOUT UT 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) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd	RQ35_008 RQ35_022
NOTE 1: The LPAAd 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 session. NOTE 2: The timeout SHALL start after the End User Intent verification.				

**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 → LPAAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL1	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation PROFILE_OPERATIONAL1 is	RQ32_001

			disabled	
2	LPAAd → S_SM- DP+(1)	Establish an HTTPs connection if previously closed		
3	LPAAd → S_SM- DP+(1)	Send ES9+.HandleNotification method	<p>MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_ NOTIF_DIS1)) sent within the timeout #IUT_LPAAd_NOTIFICATION_TIMEO UT</p> <p>Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</p>	RQ35_008 RQ35_013 RQ35_017 RQ35_018
4	S_SM- DP+(1) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd	RQ35_008 RQ35_014 RQ35_017
5	LPAAd → S_SM- DP+(1)	Establish an HTTPs connection if previously closed		
6	S_EndUser → LPAAd	Initiate the Delete Profile operation for PROFILE_OPERATIONAL1	<p>Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation</p> <p>End User acknowledges the consequences of deleting the Profile (it MAY be done in one single step combined with the End User Intent verification)</p> <p>PROFILE_OPERATIONAL1 is deleted</p>	
7	LPAAd → S_SM- DP+(1)	Send ES9+.HandleNotification method	<p>MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_ NOTIF_DEL1)) sent within the timeout #IUT_LPAAd_NOTIFICATION_TIMEO UT</p> <p>Verify the euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt; using the #PK_EUICC_ECDSA</p>	RQ35_008 RQ35_013 RQ35_014 RQ35_022 RQ35_018
8	S_SM- DP+(1) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd	RQ35_008 RQ35_022
NOTE: 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.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL2	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation PROFILE_OPERATIONAL2 is enabled	RQ32_001
2	LPAAd → S_SM-DP+(1)	Establish an HTTPs connection if previously closed		
3	LPAAd → 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_LPAAd_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) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd	RQ35_008 RQ35_014 RQ35_017
5	LPAAd → S_SM-DP+(2)	Establish an HTTPs connection		
6	LPAAd → 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_LPAAd_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) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd	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 End User Intent verification.				

**Test Sequence #05 Nominal: Different SM-DP+ Addresses in PIR and Install Notifications**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	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)		
IC5	LPAAd → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	
1	LPAAd → 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) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd. The LPAAd MAY inform the End User of the success status indicated by the Profile Installation Result.	RQ35_008 RQ35_014 RQ35_017
3	LPAAd → S_SM-DP+(2)	Establish an HTTPs connection		
4	LPAAd → 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_LPAAd_NOTIFICATION_TIMEOUT UT 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) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd	RQ35_008 RQ35_022
<p>NOTE 1: The LPAAd 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 session.</p> <p>NOTE 2: Steps 1 and 2 can be executed in parallel to the steps 3,4 and 5.</p> <p>NOTE 3: The timeout SHALL start after the End User Intent verification.</p>				

**Test Sequence #06 Nominal: Profile Download with PIR Failed**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	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	LPAAd → 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) → LPAAd	MTD_HTTP_RESP(#GET_BPP_INV)	No error exhibited by the LPAAd, s. note 1.	
IC6	LPAAd → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	
1	LPAAd → 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) → LPAAd	#R_HTTP_204_OK	No error exhibited by the LPAAd. The LPAAd MAY inform the End User of the error status indicated by the Profile	RQ35_008

			Installation Result.	
NOTE 1: The LPA <sub>d</sub> 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 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
LPA <sub>d</sub>	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	LPA <sub>d</sub> → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	
2	LPA <sub>d</sub> → S_SM-DP+(1)	Send ES9+.HandleNotification method initiated	MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_OK))	
3	LPA <sub>d</sub> → S_SM-DP+(1)	No ES9+.HandleNotification method sent	No ES9+.HandleNotification requests are sent within the timeout #IUT_LPA <sub>d</sub> _NOTIFICATION_TIMEOUT EOUT OR TLS Session closed independent of timeout.	RQ35_014
NOTE 1: The LPA <sub>d</sub> 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 session.				

NOTE 2: The timeout in Step 3 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\_071, 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).

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	LPA <sub>d</sub> → 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_SIG NATURE> 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+ → LPA <sub>d</sub>	MTD_HTTP_RESP(#R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPA <sub>d</sub> _SESSION_CLOSE_TIMEOUT. OR If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPA <sub>d</sub> _SESSION_CLOSE_TIMEOUT.	RQ31_099

**Test Sequence #02 Nominal: End User rejection due to PPR1 set**

Initial Conditions	
Entity	Description of the initial condition
eUICC	The PROFILE_OPERATIONAL4 with PPR1 is installed and enabled on the eUICC.
LPA <sub>d</sub>	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	LPA <sub>d</sub> → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of	RQ31_071 RQ31_096

			<p>the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation                  OR                  Simple End User Confirmation/Rejection if Authenticated Confirmation was requested before.                  End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.</p>	
2	S_EndUser → LPAAd	End User Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT		
3	LPAAd → S_SM-DP+	Send ES9+.CancelSession method	<p>MTD_HTTP_REQ(                  #TEST_DP_ADDRESS1,                  #PATH_CANCEL_SESSION,                  MTD_CANCEL_SESSION(                  &lt;S_TRANSACTION_ID&gt;,                  #CS_OK_EU_REJ))</p> <p>Verify:                  •&lt;EUICC_CANCEL_SESSION_SIG NATURE&gt; with the #PK_EUICC_ECDSA                  •&lt;S_TRANSACTION_ID&gt; is the same as in IC3</p>	<p>RQ56_044                  RQ56_047                  RQ65_025                  RQ31_114                  RQ31_117                  RQ31_118                  RQ31_120</p>
4	S_SM-DP+ → LPAAd	MTD_HTTP_RESP( #R_SUCCESS)	<p>If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIMEOUT.                  OR                  If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIMEOUT.</p>	RQ31_114

**Test Sequence #03 Nominal: Load BPP Error**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	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	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #GET_BPP_LOAD_ERROR)	Continue to step 2 (End User Confirmation) if requested, otherwise continue with Step 3	
2	LPAAd → S_EndUser	Request for Confirmation if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before OR Simple End User Confirmation if Authenticated Confirmation was requested before.	
3	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP( #R_SUCCESS)	No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIMEOUT.	RQ31_129

**Test Sequence #04 Nominal: End User Timeout due to PPR1 set**

Initial Conditions	
Entity	Description of the initial condition
eUICC	The PROFILE_OPERATIONAL4 with PPR1 set is installed and enabled on the eUICC.
LPAAd	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	LPA <sub>d</sub> → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation OR Simple End User Confirmation if Authenticated Confirmation was requested before.  End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.	RQ31_071 RQ31_096 RQ31_159
2	S_EndUser → LPA <sub>d</sub>	No End User Rejection or Confirmation is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT		
3	LPA <sub>d</sub> → 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_SIG NATURE> with the #PK_EUICC_ECDSA •<S_TRANSACTION_ID> is the same as in IC3	RQ56_044 RQ56_047 RQ65_025 RQ31_114 RQ31_124 RQ56_044 RQ56_047 RQ65_025 RQ31_117 RQ31_118 RQ31_120 RQ31_111
4	S_SM-DP+ → LPA <sub>d</sub>	MTD_HTTP_RESP( #R_SUCCESS)	If Step 1 was performed directly after IC3:  No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPA <sub>d</sub> _SESSION_CLOSE_TIMEOUT.  OR  If Step 1 was performed after IC4:  No ES9+.HandleNotification requests are sent within the timeout #IUT_LPA <sub>d</sub> _SESSION_CLOSE_TIMEOUT.	RQ31_114



**Test Sequence #05 Nominal: Load BPP Error due to unknown TAG**

Initial Conditions	
Entity	Description of the initial condition
LPAAd	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	LPAAd → 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_C C))	
1	S_SM-DP+ → LPAAd	MTD_HTTP_RESP(#GET_BPP_LOAD_ERROR_UNKNOWN_TAG)	Continue to step 2 (End User Confirmation) if requested, otherwise continue with Step 3	
2	LPAAd → S_EndUser	Request for Confirmation if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before OR Simple End User Confirmation if Authenticated Confirmation was requested before.	
3	LPAAd → S_SM-DP+	Send method ES9+.CancelSession	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+ → LPAAd	MTD_HTTP_RESP(#R_SUCCESS)	No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TIMEOUT.	RQ31_129

**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 due to PPR1 set**

Initial Conditions	
Entity	Description of the initial condition
eUICC	The PROFILE_OPERATIONAL4 with PPR1 set 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 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 as defined in SGP.21 [3] for Authenticated Confirmation OR Simple End User Confirmation/Rejection if Authenticated Confirmation was requested before.  End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.	RQ31_071 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,	RQ56_044 RQ56_047 RQ65_025

			MTD_CANCEL_SESSION( <S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED))  Verify: •<EUICC_CANCEL_SESSION_SIG NATURE> with the #PK_EUICC_ECDSA •<S_TRANSACTION_ID> is the same as in IC3	RQ31_114
4	S_SM-DP+ → LPAAd	MTD_HTTP_RESP( #R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TI MEOU.  OR  If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_TI MEOU.	RQ31_114

**TC\_LPAAd\_ES9+\_CancelSession\_Error**

4.4.25.2.2

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
eUICC	The PROFILE_OPERATIONAL4 with PPR1 set is installed and enabled on the eUICC.
LPAAd	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	LPAAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation OR Simple End User Confirmation/Rejection if Authenticated Confirmation was requested before.  End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.	
IC6	S_EndUser → LPAAd	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TI MEOUT		
1	LPAAd → 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+ → LPAAd	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
eUICC	The PROFILE_OPERATIONAL4 with PPR1 set is installed and enabled on the eUICC.
LPAAd	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	LPA <sub>d</sub> → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation OR Simple End User Confirmation/Rejection if Authenticated Confirmation was requested before.  End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.	
IC6	S_EndUser → LPA <sub>d</sub>	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT		
1	LPA <sub>d</sub> → 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+ → LPA <sub>d</sub>	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_LPA <sub>d</sub> _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
eUICC	The PROFILE_OPERATIONAL4 with PPR1 set is installed and enabled on the eUICC.
LPA <sub>d</sub>	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	LPA <sub>d</sub> → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation OR Simple End User Confirmation/Rejection if Authenticated Confirmation was requested before. End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.	
IC6	S_EndUser → LPA <sub>d</sub>	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT		
1	LPA <sub>d</sub> → 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_REJ))	

			#CS_OK_EU_POSTPONED))	
2	S_SM-DP+ → LPAad	MTD_HTTP_RESP(#R_ER ROR_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_LPAad_SESSION_CLOSE_TIME OUT..	RQ56_044 RQ56_047 RQ56_049 RQ31_123_1

**TC\_LPAad\_ES9+\_CancelSession\_PPRs**

General Initial Conditions	
4	<b>Entity</b>
	<b>Description of the general initial condition</b>
	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	
	<b>Entity</b>
	<b>Description of the initial condition</b>
	eUICC
	The eUICC's RAT is configured as detailed SGP.21 Annex H: <ul style="list-style-type: none"> <li>one PPAR authorizing PPR1 and PPR2 for all MNOs with End User consent required (i.e. #PPRS_ALLOWED)</li> <li>no additional rules</li> </ul>
	LPAad
	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	LPAad → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation OR Simple End User Confirmation/Rejection if Authenticated Confirmation was	RQ31_102 RQ31_103 RQ29_007_1 RQ29_008 RQ29_009 RQ29_015 RQ29_011 RQ29_013

			requested before. Relevant information about PPRs is shown, including consequences for the End User, and the End User consent is requested if not requested before.	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	<p>MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION( &lt;S_TRANSACTION_ID&gt;, #CS_OK_EU_REJ))</p> <p>OR</p> <p>MTD_HTTP_REQ( #TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION( &lt;S_TRANSACTION_ID&gt;, #CS_OK_EU_POSTPONED))</p> <p>Verify:                  •&lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; with the #PK_EUICC_ECDSA                  •&lt;S_TRANSACTION_ID&gt; is the same as in IC3</p>	RQ31_100 RQ31_105 RQ31_160
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	<p>If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.</p> <p>OR</p> <p>If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.</p>	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 eUICC's RAT is configured as detailed SGP.21 Annex H: <ul style="list-style-type: none"> <li>one PPAR authorizing PPR1 and PPR2 for all MNOs with End User</li> </ul>



	consent required (i.e. #PPRS_ALLOWED) <ul style="list-style-type: none"> <li>no additional rules</li> </ul>
LPAAd	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	LPAAd → S_EndUser		The LPA provides means for the End User Confirmation/Rejection of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation OR Simple End User Confirmation/Rejection if Authenticated Confirmation was requested before.  Relevant information about PPRs is shown, including consequences for the End User, and the End User consent is requested if not requested before.	RQ31_102 RQ31_103 RQ29_007_1 RQ29_008 RQ29_009 RQ29_015 RQ29_011 RQ29_013 RQ29_018
2	S_EndUser → LPAAd	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT		
3	LPAAd → S_SM-DP+	Send method ES9+.CancelSession	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>, 	RQ31_100 RQ31_105 RQ31_160

			#CS_OK_EU_POSTPONED))  Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_ECDSA •<S_TRANSACTION_ID> is the same as in IC3	
4	S_SM-DP+ LPAAd →	MTD_HTTP_RESP(#R_SUCCESS)	<p>If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout              #IUT_LPAAd_SESSION_CLOSE_TIMEOUT.</p> <p>OR</p> <p>If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout              #IUT_LPAAd_SESSION_CLOSE_TIMEOUT.</p>	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

##### TC\_LPAAd\_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.
LPAAd	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	LPA <sub>d</sub> → 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+ → LPA <sub>d</sub>	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+ → LPA <sub>d</sub>	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 LPA<sub>d</sub> 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	LPA <sub>d</sub> → 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	RQ31_032

			at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 or TLS_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).	
2	S_SM-DP+ → LPAAd	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 LPAAd in IC1	RQ31_032
3	S_SM-DP+ → LPAAd	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\_LPAAd\_HTTPS\_ErrorCases

General Initial Conditions	
Entity	Description of the general initial condition
LPAAd	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	LPAAd → 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+ → LPAAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_SIG)	LPAAd aborts AddProfile procedure	RQ31_032 RQ45_026
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_EXPIRED)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_026
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ56_003

**Test Sequence #03 Error: Invalid TLS Certificate with critical extension not set**

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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_CRITICA L_EXT)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_026
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ56_003

**Test Sequence #04 Error: Invalid TLS Certificate with invalid 'key usage' extension**

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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_KEY_US AGE)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_031
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be	RQ26_023 RQ56_003

			sent.	
--	--	--	-------	--

**Test Sequence #05 Error: Invalid TLS Certificate with invalid 'extended key usage' extension**

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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_EXT_KEY _USAGE)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_031
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ56_003

**Test Sequence #06 Error: Invalid TLS Certificate with invalid 'Certificate Policies' extensions**

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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_CERT_P OL)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_031
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ56_003

**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+	MTD_TLS_SERVER_HELLO_ETC(#TL	LPAd aborts AddProfile	RQ31_032

	→ LPA <sub>d</sub>	S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_CURVE)	procedure	RQ45_031
3	LPD <sub>d</sub> → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	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\_LPA<sub>d</sub>\_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 LPA <sub>d</sub> 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	LPA <sub>d</sub> → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ( #TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(	RQ31_033

			<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS))  • Extract <EUICC_CHALLENGE>	
2	S_SM-DS → LPAAd	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\_LPAAd\_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 LPAAd for profile download during test sequence execution, as the LPAAd 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	LPAAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ( #TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS))	
1	S_SM-DS → LPAAd	MTD_HTTP_RESP(#R_ERRO R_8_9_1_3_8)	LPAAd aborts AddProfile procedure	RQ31_034 RQ58_020
2	LPAAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_T IMEOUT 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	LPAAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ( #TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH,	



			MTD_INITIATE_AUTHENTICATIO N( <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_T IMEOUT 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_AUTHENTICATIO N( <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_T IMEOUT 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_AUTHENTICATIO N( <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_TI MEOUT 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	LPAAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ( #TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS))	
1	S_SM-DS → LPAAd	MTD_HTTP_RESP( #INITIATE_AUTH_INV_CERT_D S)	LPAAd aborts AddProfile procedure	RQ31_052 RQ58_013
2	LPAAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_T IMEOUT 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	LPAAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ( #TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS))	
1	S_SM-DS → LPAAd	MTD_HTTP_RESP(#INITIATE _AUTH_INV_SIGN_DS)	LPAAd aborts AddProfile procedure	RQ31_052 RQ58_013
2	LPAAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_SESSION_CLOSE_T IMEOUT 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	LPAAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ( #TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS))	

1	S_SM-DS → LPAAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_SMDS_ADDRESS)	LPAAd informs the S_EndUser and aborts the AddProfile procedure	RQ31_045 RQ31_052 RQ58_013
2	LPAAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_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	LPAAd → 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 → LPAAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_CI_DS)	LPAAd aborts AddProfile procedure	RQ31_048 RQ31_052 RQ58_013
2	LPAAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT_LPAAd_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).
<p>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.</p>	

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_AUTHENTICATI ON( <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS))  • Extract <EUICC_CHALLENGE>	
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_A UTH_DS_OK)	MTD_HTTP_REQ(#TEST_ROOT _DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATC H_ID_DEV_INFO))  Verify: • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_DS_OK • <EUICC_SIGNATURE1> using	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

			<p>the #PK_EUICC_ECDSA</p> <ul style="list-style-type: none"> <li>• if matchingId field in #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO is missing OR matchingId field is present and &lt;MATCHING_ID&gt; is empty</li> <li>• if &lt;S_SMDS_CHALLENGE&gt; present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO is the same as in &lt;S_SMDS_SIGNED1&gt; present in #INITIATE_AUTH_DS_OK</li> <li>• for #DEVICE_INFO:                         <ul style="list-style-type: none"> <li>- 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]</li> <li>- 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)</li> <li>- if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL.</li> <li>- if O_D_UMTS_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL.</li> <li>- if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL.</li> <li>- 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.</li> <li>- if O_D_CDMA2000_EHRPD then cdma2000ehrpdsupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL.</li> </ul> </li> </ul>	<p>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</p>
--	--	--	--	---

			<p>– if O_D_LTE then eutranSupportedRelease is set to the highest release as defined in #IUT_LTE_EUTRAN_REL.</p> <p>– if O_D_NFC_TS26 then contactlessSupportedRelease is set to the highest release as defined in #IUT_NFC_REL.</p> <p>– if O_D_CRL then rspCrISupportedVersion is set to the highest release as defined in #IUT_RSP_VERSION .</p> <p>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.</p>	
2	S_SM-DS → LPAAd	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).
<p>NOTE: In order to avoid potentially misleading errors on LUI, the S_SM-DP+ SHALL be available to the LPAAd for profile download during test sequence execution. The test tool SHALL NOT check the ES9+ communication.</p>	

Step	Direction	Sequence/ Description	Expected result	REQ
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAAd → 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_I D_DEV_INFO))  Verify: • if matchingId field in #R_AUTH_SERVER_DS_MATCH_I D_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_DSA DDR1)	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_ADD RESS1 , #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S _TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_I D_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	LPAAd → 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 → LPAAd	MTD_HTTP_RESP(#R_ERROR_8_1_3_6_1)	LPAAd aborts AddProfile procedure	RQ31_061
3	LPAAd → S_SM-DS	No Profile download action	No requests are sent on ES11 within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME OUT 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	LPAAd → 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 → LPAAd	MTD_HTTP_RESP(#R_ERROR_8_1_3_6_3)	LPAAd aborts AddProfile procedure	RQ31_061
3	LPAAd → S_SM-DS	No Profile download action	No requests are sent on ES11 within the timeout #IUT_LPAAd_SESSION_CLOSE_TIME EOUT 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	LPAAd → 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 → LPAAd	MTD_HTTP_RESP(#R_ERROR_8_1_6_1)	LPAAd aborts AddProfile procedure	RQ58_030 RQ58_039

3	LPAAd → S_SM-DS	No Profile download action	No requests are sent on ES11 within the timeout #IUT_LPAAd_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	LPAAd → 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 → LPAAd	MTD_HTTP_RESP(#R_ERROR_8_10_1_3_9)	LPAAd aborts AddProfile procedure	RQ56_030
3	LPAAd → S_SM-DS	No Profile download action	No requests are sent on ES11 within the timeout #IUT_LPAAd_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	LPAAd → 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 → LPAAd	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 → LPAAd	MTD_HTTP_RESP (#AUTH_CLIENT_DS_OK_DSA DDR1)	No Error	
IC5	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11 with #TEST_DS_ADDRESS1 and #CERT_S_SM_DS2_TLS			
IC6	LPAAd → 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 → LPAAd	MTD_HTTP_RESP(#INITIATE_ AUTH_DS_OK_1)	MTD_HTTP_REQ(#TEST_DS_ADDR ESS1 , #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_ TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID _DEV_INFO_1))	
1	S_SM-DS → LPAAd	MTD_HTTP_RESP(#R_ERROR _8_9_5_3_9)	LPAAd aborts AddProfile procedure	RQ31_090 RQ31_083
2	LPAAd → S_SM-DS	No Profile download action	No requests are sent on ES11 within the timeout #IUT_LPAAd_SESSION_CLOSE_TIM EOUT 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

##### 4.4.29.2.1

##### 4.4.29.2 Test Cases

##### TC\_LPAAd\_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	LPA <sub>d</sub> → S_SM-DS	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TL S_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 RQ58_001
2	S_SM-DS → LPA <sub>d</sub>	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 → LPA <sub>d</sub>	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 LPA<sub>d</sub> 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	LPA <sub>d</sub> →	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(	RQ31_032

	S_SM-DS		<p>#IUT_TLS_VERSION,                  &lt;TLS_CIPHER_SUITES&gt;,                  &lt;SESSION_ID_CLIENT&gt;,                  &lt;EXT_SHA256_ECDSA&gt;)</p> <p>Verify the following:</p> <ul style="list-style-type: none"> <li>• #IUT_TLS_VERSION SHALL be 1.2 or higher</li> <li>• &lt;TLS_CIPHER_SUITES&gt; SHALL be at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 or TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</li> <li>• if &lt;SESSION_ID_CLIENT&gt; is non-empty then it SHALL be different from &lt;SESSION_ID_CLIENT&gt; and &lt;S_SESSION_ID_SERVER&gt; extracted in IC1.</li> <li>• &lt;EXT_SHA256_ECDSA&gt; SHALL have at least the 'supported_signature_algorithms' extension set with HashAlgorithm sha256 (04) and SignatureAlgorithm ecDSA (03).</li> </ul>	
2	S_SM-DS → LPA d	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 <ul style="list-style-type: none"> <li>• &lt;CLIENT_TLS_EPHEM_KEY&gt; is different from the one used by LPA d in IC1</li> </ul>	RQ31_032
3	S_SM-DS → LPA d	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established	RQ31_032 RQ58_001 RQ60_001 RQ60_002 RQ61_001

4.4.29.2.2

### TC\_LPA d\_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	LPAAd → 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 → LPAAd	MTD_TLS_SERVER_HELLO_ETC(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_SIG)	LPAAd aborts AddProfile procedure	RQ31_032 RQ45_026 RQ45_028
3	LPDd → S_SM-DS	TLS 1.2 close	A TLS alert is sent with Fatal-level	RQ26_023 RQ58_002

**Test Sequence #02 Error: Expired TLS Certificate**

Step	Direction	Sequence / Description	Expected result	REQ
1	LPAAd → 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 → LPAAd	MTD_TLS_SERVER_HELLO_ETC(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_EXPIRED)	LPAAd aborts AddProfile procedure	RQ31_032 RQ45_026
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ58_002

**Test Sequence #03 Error: Invalid TLS Certificate with critical extension not set**

Step	Direction	Sequence / Description	Expected result	REQ
1	LPAAd → 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 → LPAAd	MTD_TLS_SERVER_HELLO_ETC(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_CRITICA L_EXT)	LPAAd aborts AddProfile procedure	RQ31_032 RQ45_026
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ58_002

**Test Sequence #04 Error: Invalid TLS Certificate with invalid 'key usage' extension**

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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_KEY_US AGE)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_033
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ58_002

**Test Sequence #05 Error: Invalid TLS Certificate with invalid 'extended key usage' extension**

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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_EXT_KEY _USAGE)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_033
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ58_002

**Test Sequence #06 Error: Invalid TLS Certificate with invalid 'Certificate Policies' extensions**

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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_CERT_P OL)	LPAd aborts AddProfile procedure	RQ31_032 RQ45_033

3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	RQ26_023 RQ58_002
---	-------------------	---------------	--	----------------------

**Test Sequence #07 Error: Invalid TLS Certificate based on Invalid CI (Invalid Curve)**

Step	Direction	Sequence / Description	Expected result	REQ
1	LPAAd → 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 → LPAAd	MTD_TLS_SERVER_HELLO_ETC(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_CURVE)	LPAAd aborts AddProfile procedure	RQ26_029 RQ31_032 RQ45_033
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected. A TLS alert MAY be sent.	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

**4.5.1.2.1**

**4.5.1.2 Test Cases**

**TC\_ROOT\_SM\_DS\_ES12.RegisterEvent**

General Initial Conditions	
Entity	Description of the general initial condition
Root SM-DS	<ul style="list-style-type: none"> <li>• No TLS connections are established between the Root SM-DS and any of the simulator test tools.</li> </ul>



**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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not already used by the Root SM-DS.</li> </ul>

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, #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		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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not already used by the Root SM-DS.</li> </ul>

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, #PATH_REGISTER_EVENT, #PATH_REGISTER_EVENT,	MTD_HTTP_RESP(#R_SUCCESS)	RQ59_003 RQ59_012 RQ62_001

		MTD_REGISTER_EVENT( #S_SM_DP+_F_REQ_ID, #FUNCTION_CALL_ID_1, #EID1, #TEST_DP_ADDRESS1, #EVENT_ID_1, TRUE))		RQ62_002
2	S_LPAd → Root SM-DS	PROC_ES11_VERIFY_EVENT_RETRIEVAL		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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is already used by the Root SM-DS.</li> </ul>

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, #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_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	<ul style="list-style-type: none"> <li>No TLS connections are established between the Alternative SM-DS and any of the simulator test tools.</li> </ul>

**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, #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, <EVENT_ID>, 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 RQ62_004 RQ62_006 RQ62_007 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_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_013

				RQ59_009 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
6	S_SM-DP+ → Alt. SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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)	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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not already used by the Alternative SM-DS.</li> </ul>

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, #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, <EVENT_ID>, FALSE))  Extract the value of <EVENT_ID>	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, #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, <EVENT_ID>, FALSE))  Verify that <EVENT_ID> in step 3 is not equal to <EVENT_ID>	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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not already used by the Alternative SM-DS.</li> </ul>

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, #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	TLS communication is initiated with S_SM-DS	MTD_TLS_CLIENT_HELLO( #TLS_VERSION_1_2, #MIN_TLS_CIPHER_SUITES, #S_SESSION_ID_EMPTY, <EXT_SHA256_ECDSA>)  No TLS response from S_SM-DS	RQ36_010
3	Alt. SM-DS → S_SM-DP+	Wait for #IUT_SM_DS_TLS_TIMEOUT to expire.	MTD_HTTP_RESP( #R_ERROR_8_9_5_1)	RQ59_005 RQ59_007 RQ59_010 RQ59_015 RQ62_001 RQ62_002
4	S_SM-DP+ → Alt. SM-DS	PROC_TLS_INITIALIZATION_MUTUAL_AUTH on ES12		
5	S_SM-DP+ → Alt. SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_REGISTER_EVENT, MTD_REGISTER_EVENT( #S_SM_DP+_F_REQ_ID, #FUNCTION_CALL_ID_1, #EID1, #TEST_DP_ADDRESS1, #EVENT_ID_1, TRUE))		
6	Alt. SM-DS → S_SM-DS	PROC_TLS_INITIALIZATION_MUTUAL_AUTH on ES15		
7	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>,	RQ59_007 RQ62_001 RQ62_002

			#EID1, #IUT_SM_DS_ADDRESS, <EVENT_ID>, FALSE))	
8	S_SM-DS → Alt. SM-DS	MTD_HTTP_RESP(#R_SUCCE SS) on ES15	No Error	
9	Alt. SM-DS → S_SM-DP+	Successful result is sent to the S_SM-DP+	MTD_HTTP_RESP(#R_SUCCESS) on ES12	RQ59_007 RQ62_001 RQ62_002

**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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not already used by the Alternative SM-DS for #EID1.</li> </ul>

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, #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, <EVENT_ID>, 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_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, #PATH_REGISTER_EVENT, MTD_REGISTER_EVENT( #S_SM_DP+_F_REQ_ID, #FUNCTION_CALL_ID_1, #EID1, #TEST_DP_ADDRESS1, #EVENT_ID_1, 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, <EVENT_ID>, FALSE))	RQ36_009 RQ62_001 RQ62_002
11	S_SM-DS → Alt. SM-DS	MTD_HTTP_RESP(#R_SUCCE S) on ES15	No Error	
12	Alt. SM-DS → S_SM- DP+	Successful result is sent to the S_SM-DP+	MTD_HTTP_RESP(#R_SUCCE S) on ES12	RQ36_009 RQ62_001 RQ62_002

**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
SM-DS	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is already used by the Alternative SM-DS.</li> </ul>



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, #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

## 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	<ul style="list-style-type: none"> <li>• #EVENT_ID_1 was registered for #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_DELETE_EVENT,	MTD_HTTP_RESP(#R_SUCCE SS)	RQ36_024 RQ36_025 RQ36_025_1

		MTD_DELETE_EVENT( #S_SM_DP+_F_REQ_ID, #FUNCTION_CALL_ID_1, #EID1, #EVENT_ID_1))		RQ36_029 RQ36_030 RQ59_016 RQ59_021 RQ59_023 RQ59_024 RQ510_019 RQ62_001 RQ62_002 RQ62_005 RQ62_006 RQ65_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_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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not registered.</li> </ul>

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, #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
Alt. SM-DS	<ul style="list-style-type: none"> <li>#EVENT_ID_1 was registered for #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #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		RQ59_016_1

### TC\_ALT\_SM\_DS\_ES12.DeleteEvent

4.5.2.2.2 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	<ul style="list-style-type: none"> <li>No TLS connections are established between the Alternative SM-DS and any of the simulator test tools.</li> </ul>

### 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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 registration for #EID1 and #TEST_DP_ADDRESS1 was cascaded using &lt;EVENT_ID_R&gt; to the Root SM-DS.</li> </ul>

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, #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( #IUT_SM_DS_ADDRESS,	RQ36_027 RQ36_028

			<pre>#PATH_DELETE_EVENT, MTD_DELETE_EVENT( &lt;FUNCTION_REQ_ID&gt;, &lt;FUNCTION_CALL_ID&gt;, #EID1, &lt;EVENT_ID&gt;))</pre> <p>Verify that &lt;EVENT_ID&gt; is equal to &lt;EVENT_ID_R&gt;</p>	RQ36_031 RQ36_032 RQ59_016 RQ59_017 RQ59_017_2 RQ59_023 RQ62_001 RQ62_002 RQ62_004 RQ62_006 RQ62_007 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_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_024 RQ510_019 RQ62_001 RQ62_002 RQ62_005 RQ62_006 RQ65_001 RQ65_002 RQ65_003 RQ65_005 RQ65_007 RQ65_008 RQ65_009 RQ65_031
6	S_SM-DP+ → Alt. SM-DS	<pre>MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_DELETE_EVENT, MTD_DELETE_EVENT( #S_SM_DP+_F_REQ_ID, #FUNCTION_CALL_ID_1, #EID1, #EVENT_ID_1))</pre>	MTD_HTTP_RESP( #R_ERROR_8_9_5_3_9)	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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 registration for #EID1 and #TEST_DP_ADDRESS1 was cascaded using &lt;EVENT_ID_R&gt; to the Root SM-DS.</li> </ul>

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, #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( #IUT_SM_DS_ADDRESS, #PATH_DELETE_EVENT, MTD_DELETE_EVENT( <FUNCTION_REQ_ID>, <FUNCTION_CALL_ID>, #EID1, <EVENT_ID>))  Verify that <EVENT_ID> 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_SM-DP+ → Alt. SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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)	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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 registration for #EID1 and #TEST_DP_ADDRESS1 was cascaded using &lt;EVENT_ID_R&gt; to the Root SM-DS.</li> </ul>

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, #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	TLS communication is initiated with S_SM-DS	MTD_TLS_CLIENT_HELLO( #TLS_VERSION_1_2, #MIN_TLS_CIPHER_SUITES, #S_SESSION_ID_EMPTY, <EXT_SHA256_ECDSA>)  No TLS response from S_SM- DS	RQ36_028
3	Alt. SM-DS → S_SM-DP+	Wait for #IUT_SM_DS_TLS_TIMEOUT to expire.	MTD_HTTP_RESP( #R_ERROR_8_9_5_1)	RQ59_016_1 RQ59_018 RQ59_022 RQ59_025 RQ62_001 RQ62_002
4	S_SM-DP+ → Alt. SM-DS	PROC_TLS_INITIALIZATION_MUTUAL_AUTH on ES12		
5	S_SM-DP+ → Alt. SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_DELETE_EVENT, MTD_DELETE_EVENT( #S_SM_DP+_F_REQ_ID, #FUNCTION_CALL_ID_1, #EID1, #EVENT_ID_1))		
6	Alt. SM-DS → S_SM-DS	PROC_TLS_INITIALIZATION_MUTUAL_AUTH on ES15		
7	Alt. SM-DS → S_SM-DS	Call ES15.DeleteEvent	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_DELETE_EVENT, MTD_DELETE_EVENT( <FUNCTION_REQ_ID>, <FUNCTION_CALL_ID>,	RQ59_018 RQ62_001 RQ62_002

			#EID1, <EVENT_ID>))  Verify that <EVENT_ID> is equal to <EVENT_ID_R>	
8	S_SM-DS → Alt. SM-DS	MTD_HTTP_RESP( #R_SUCCESS) on ES15	No Error	RQ510_019 RQ62_001 RQ62_002
9	Alt. SM-DS → S_SM-DP+	SM-DS sends response back to S_SM-DP+	MTD_HTTP_RESP( #R_SUCCESS) on ES12	RQ59_018 RQ510_019 RQ62_001 RQ62_002

**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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 registration for #EID1 and #TEST_DP_ADDRESS1 was cascaded using &lt;EVENT_ID_R&gt; to the Root SM-DS.</li> </ul>

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, #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( #IUT_SM_DS_ADDRESS, #PATH_DELETE_EVENT, MTD_DELETE_EVENT( <FUNCTION_REQ_ID>, <FUNCTION_CALL_ID>, #EID1, <EVENT_ID>))	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

*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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 registration for #EID1 and #TEST_DP_ADDRESS1 was cascaded using &lt;EVENT_ID_R&gt; to the Root SM-DS.</li> </ul>

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, #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_SM-DP+ → Alt. SM-DS	PROC_TLS_INITIALIZATION_MUTUAL_AUTH on ES12		
3	Alt. SM-DS → S_SM-DS	Call ES15.DeleteEvent	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_DELETE_EVENT, MTD_DELETE_EVENT( <FUNCTION_REQ_ID>, <FUNCTION_CALL_ID>, #EID1, <EVENT_ID>))	RQ59_016_1 RQ62_001 RQ62_002
4	S_SM-DS → Alt. SM-DS	MTD_HTTP_RESP( #R_SUCCESS) on ES15	No Error	RQ510_019 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_016_1 RQ510_019 RQ62_001 RQ62_002



### TC\_ALT\_SM\_DS\_ES12.DeleteEvent\_Error\_Nonexistent\_EventID

General Initial Conditions	
Entity	Description of the general initial condition
Alt. SM-DS	<ul style="list-style-type: none"> <li>No TLS connections are established between the Alternative SM-DS and any of the simulator test tools.</li> </ul>

#### 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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not registered.</li> </ul>

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, #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

#### TC\_ROOT\_SM\_DS\_ES15.RegisterEvent

General Initial Conditions	
Entity	Description of the general initial condition
4.5.3.2.1 Root SM-DS	<ul style="list-style-type: none"> <li>No TLS connections are established between the Root SM-DS and any of the simulator test tools.</li> </ul>

#### 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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not already used by the Root SM-DS.</li> </ul>

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, #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		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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not already used by the Root SM-DS.</li> </ul>

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, #PATH_REGISTER_EVENT,	MTD_HTTP_RESP( #R_SUCCESS)	RQ36_010 RQ36_011 RQ36_012

		MTD_REGISTER_EVENT( #S_SM_DS_F_REQ_ID, #FUNCTION_CALL_ID_1, #EID1, #TEST_ALT_DS_ADDRESS, #EVENT_ID_1, FALSE))		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		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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is already used by the Root SM-DS.</li> </ul>

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, #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_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**

**TC\_ROOT\_SM\_DS\_ES15.DeleteEvent**

General Initial Conditions	
Entity	Description of the general initial condition
Root SM-DS	<ul style="list-style-type: none"> <li>• No TLS connections are established between the Alternative SM-DS and any of the simulator test tools.</li> </ul>

**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	<ul style="list-style-type: none"> <li>• #EVENT_ID_1 was registered for #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #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_006 RQ65_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_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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 is not registered.</li> </ul>

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, #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
Alt. SM-DS	#EVENT_ID_1 was registered for #EID1 and #TEST_DP_ADDRESS1.

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, #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		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**

**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
  - CERT\_SM\_XXauth\_ECDSA = CERT\_SM\_DSauth\_ECDSA
  - 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**

**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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATC HING_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	<ul style="list-style-type: none"> <li>• #EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> <li>• #EVENT_ID_2 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS2.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>No Events have been registered in the SM-DS for #EID1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATC HING_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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHI NG_ID_OMITTED))	MTD_HTTP_RESP( #R_AUTH_CLIENT_DS_EVEN T_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	<ul style="list-style-type: none"> <li>• #EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> <li>• #EVENT_ID_2 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS2.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATC HING_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	<ul style="list-style-type: none"> <li>No Events have been registered in the SM-DS for #EID1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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 Nominal Matching ID containing EventID with 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 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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATC HING_ID_EVENT_ID))	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 Nominal Matching ID containing EventID with 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 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	<ul style="list-style-type: none"> <li>• #EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> <li>• #EVENT_ID_2 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS2.</li> </ul>

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	<ul style="list-style-type: none"> <li>• #EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
5	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1,	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	

		#IUT_SM_DS_ADDRESS))		
9	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
13	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_ OK)	
17	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1,	MTD_HTTP_RESP( #R_INITIATE_AUTH_O K)	

		#IUT_SM_DS_ADDRESS))		
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_ OK)	
5	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1,	MTD_HTTP_RESP( #R_INITIATE_AUTH_ OK)	

		#IUT_SM_DS_ADDRESS))		
9	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
13	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
17	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1,	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	

		#IUT_SM_DS_ADDRESS))		
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <INVALID_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHING_I D_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	<ul style="list-style-type: none"> <li>#EVENT_ID_1 has been registered in the SM-DS with #EID1 and #TEST_DP_ADDRESS1.</li> </ul>



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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #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	<ul style="list-style-type: none"> <li>No Events have been registered in the SM-DS for #EID1.</li> </ul>

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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
1	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_SMDS_8_9_5_3_9))	MTD_HTTP_RESP( #R_ERROR_8_9_5_3_9)	RQ26_005 RQ26_006 RQ26_012 RQ26_014

		<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHING_I D_EVENT_ID))		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
--	--	---	--	--

**TC\_SM-DS\_ES11.AuthenticateClientBRP**

<b>4 General Initial Conditions</b>	
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 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 #01 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all 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 #01 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all 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 #01 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all 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 #01 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all 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 #01 defined in section 4.5.6.2.1 TC\_SM-DS\_ES11.AuthenticateClientNIST except that all keys and certificates SHALL be based on BrainpoolP256r1.

#### ***Test Sequence #07 Nominal Matching ID containing EventID with 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 keys and certificates SHALL be based on BrainpoolP256r1.

#### ***Test Sequence #08 Nominal Matching ID containing EventID with two pending Events***

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 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
  - CERT\_CLIENT\_TLS = #CERT\_SM\_DS\_TLS for NIST
- SERVER = Root S\_SM-DS
  - 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
  - CERT\_CLIENT\_TLS = #CERT\_SM\_DS\_TLS for BRP
- SERVER = Root S\_SM-DS

- 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+
  - CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DP\_TLS for NIST
- SERVER = Alternative or Root SM-DS under test.
  - 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+
  - CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DP\_TLS for BRP
- SERVER = Alternative or Root SM-DS under test.
  - 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
  - CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DS\_TLS for NIST
- SERVER = Root SM-DS under test.
  - 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
  - CERT\_S\_CLIENT\_TLS = CERT\_S\_SM\_DS\_TLS for BRP
- SERVER = Root SM-DS under test.
  - 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\_Mutual\_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 = Root SM-DS under test.
  - CERT\_SERVER\_TLS = #CERT\_SM\_DS\_TLS for NIST

##### **4.5.10.2 TC\_SM- DS\_ES11\_Server\_Mutual\_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 = Root SM-DS under test.
  - 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**

**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+	<ul style="list-style-type: none"> <li>• PROFILE_OPERATIONAL1 is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• There is currently no TLS connection established to the S_SM-DS.</li> </ul>

General Initial Conditions for SM-DS as Client under test	
Entity	Description of the initial condition
SM-DS	<ul style="list-style-type: none"> <li>• EventID to be used by the S_SM-DP+ is not already used in the SM-DS.</li> <li>• There is currently no TLS connection established to the S_SM-DS.</li> </ul>

**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, #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>)	RQ26_023 RQ26_025 RQ26_026 RQ26_027 RQ26_028 RQ45_026

			Verify that: • <TLS_CIPHER_SUITES> SHALL contain at least one of TLS_ECDHE_ECDSA_WITH_A ES_128_GCM_SHA256 or TLS_ECDHE_ECDSA_WITH_A ES_128_CBC_SHA256  • <EXT_SHA256_ECDSA> SHALL have at least the 'supported_signature_algorith m_s' extension set with HashAlgorithm sha256 (04) and SignatureAlgorithm ecDSA (03).	RQ45_026_1 RQ510_001 RQ56_002 RQ59_001 RQ60_003 RQ61_001
2	S_SERVER → CLIENT	MTD_TLS_MUTUAL_AUTH_SER VER_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_CL IENT_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

**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.

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 smsdAddress #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, #PATH_REGISTER_EVENT, MTD_REGISTER_EVENT(		

	#EID1, #TEST_DP_ADDRESS1, <EVENT_ID>, TRUE)			
IC2	PROC_TLS_INITIALIZATION_MUTUAL_AUTH Extract <CLIENT_TLS_EPHEM_KEY> from the ClientKeyExchange message Extract <SESSION_ID_CLIENT> and <SESSION_ID_RANDOM>			
IC3	S_SERVER → CLIENT	Close TLS session (unless CLIENT has already closed TLS session)		
IC4	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.	
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

**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, #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>)	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)	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

**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, #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>)	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)	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

**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, #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>)	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 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

		#CLIENT_CERT_TYPE, #S_SAH_SHA256_ECDSA, #DIST_NAME_CI)		RQ45_026_1 RQ510_001 RQ56_002 RQ59_001 RQ60_003 RQ61_001
--	--	--	--	---

**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, #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>)	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_CRI TICAL_EXT, <SERVER_TLS_EPHEM_KEY>, #CLIENT_CERT_TYPE, #S_SAH_SHA256_ECDSA, #DIST_NAME_CI)	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

**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])		

		<p>section 3.1.1) for the SM-DS use case with smdsAddress #TEST_ROOT_DS_ADDRESS to be used for Event Registration.</p> <p>When the Client under test is the SM-DS, the S_SM-DP+ calls ES12.RegisterEvent configured as follows:</p> <pre>MTD_HTTP_REQ(   #IUT_SM_DS_ADDRESS,   #PATH_REGISTER_EVENT,   MTD_REGISTER_EVENT(     #EID1,     #TEST_DP_ADDRESS1,     &lt;EVENT_ID&gt;,     TRUE)</pre>		
1	CLIENT → S_SERVER	Send TLS Client Hello	<pre>MTD_TLS_CLIENT_HELLO(   #IUT_CLIENT_TLS_VER,   &lt;TLS_CIPHER_SUITES&gt;,   &lt;SESSION_ID_CLIENT&gt;,   &lt;EXT_SHA256_ECDSA&gt;)</pre>	<p>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</p>
2	S_SERVER → CLIENT	<pre>MTD_TLS_MUTUAL_AUTH_SERVER_HELLO_ETC(   #TLS_VERSION_1_2,   &lt;S_SEL_TLS_CIPHER_SUITE&gt;,   &lt;SESSION_ID_RANDOM&gt;,   #CERT_S_SERVER_TLS_INV_KEY_USAGE,   &lt;SERVER_TLS_EPHEM_KEY&gt;,   #CLIENT_CERT_TYPE,   #S_SAH_SHA256_ECDSA,   #DIST_NAME_CI)</pre>	<p>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</p>	<p>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</p>

**Test Sequence #08 Error: Invalid TLS Certificate with invalid 'extended key usage' extension**

Step	Direction	Sequence / Description	Expected result	REQ
IC1		<p>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.</p> <p>When the Client under test is the SM-DS, the S_SM-DP+ calls ES12.RegisterEvent configured as follows:</p> <pre>MTD_HTTP_REQ(   #IUT_SM_DS_ADDRESS,   #PATH_REGISTER_EVENT,   MTD_REGISTER_EVENT(     #EID1,     #TEST_DP_ADDRESS1,     &lt;EVENT_ID&gt;,     TRUE)</pre>		

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>)	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_SERV ER_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)	Client sends a TLS Fatal-alert during or after any of the messages sent by the S_SERVER in MTD_TLS_MUTUAL_AUTH_S ERVER_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

**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, #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>)	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_SHA_SHA256_ECDSA, #DIST_NAME_CI)	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
---	----------------------	--	---	---

**TC\_Client\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP**

**4.6.1.2.2 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

**4.6.2.2.1**

**4.6.2.2 Test Cases**

**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	RQ26_025

			<SERVER_TLS_EPHEM_KEY> value used in IC1.	
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, #S_EXT_SHA256_ECDSA)	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> )	Server sends a TLS Fatal-alert	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>)	Server sends a TLS Fatal-alert	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	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

			TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	
2	S_CLIENT → SERVER	MTD_TLS_MUTUAL_AUTH_CLIENT_EXCH( #CERT_S_CLIENT_TLS_INV_CRITICAL_EXT, <CLIENT_TLS_EPHEM_KEY>)	Server sends a TLS Fatal-alert	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>)	Server sends a TLS Fatal-alert	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>)	Server sends a TLS Fatal-alert	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	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

			_128_CBC_SHA256	
2	S_CLIENT → SERVER	MTD_TLS_MUTUAL_AUTH_CLIENT_EXCH( #CERT_S_CLIENT_TLS_INV_ CERT_POL, <CLIENT_TLS_EPHEM_KEY>)	Server sends a TLS Fatal-alert	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>)	Server sends a TLS Fatal-alert	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

## **TC\_Server\_Mutual\_Authentication\_for\_HTTPS\_EstablishmentBRP**

### ***Test Sequence #01 Nominal: HTTPS Session Establishment***

4.6.2.2.1 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

4.6.3.2.1

### **4.6.3.2 Test Cases**

## **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_LPA <sub>d</sub> → 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_LPA <sub>d</sub> → SERVER	MTD_TLS_CLIENT_KEY_EXC H_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_LPA <sub>d</sub> → 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_ET C( #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_LPA <sub>d</sub> → SERVER	MTD_TLS_CLIENT_KEY_EXCH _ETC(<CLIENT_TLS_EPHEM_K EY>)	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_LPA → 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

**TC\_Server\_Authentication\_for\_HTTPS\_EstablishmentBRP**

4.6.3.2.2

**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	<p>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+</p> <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul>

#### **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 )	<p>#R_PREP_DOWNLOAD_WITH_CC SW=0x9000</p> <p>The &lt;EUICC_SIGNATURE2&gt; SHALL be verified with the #PK_EUICC_ECDSA.</p> <p>Verify that the &lt;S_TRANSACTION_ID&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_WITH_CC.</p> <p>Verify that the &lt;S_HASHED_CC&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_WITH_CC.</p> <p>Extract the &lt;OTPK_EUICC_ECKA&gt; and reuse the same value in step 4</p>	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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP are sent to the eUICC</li> <li>• the same GSMA CI as for the first attempt has been chosen for signing and for verification</li> </ul>		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)	<p>#R_PREP_DOWNLOAD_WITH_CC SW=0x9000</p> <p>The &lt;EUICC_SIGNATURE2&gt; SHALL be verified with the #PK_EUICC_ECDSA.</p> <p>Verify that the &lt;S_TRANSACTION_ID&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_RETRY_CC.</p> <p>Verify that the &lt;S_HASHED_CC&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_RETRY_CC.</p> <p>Verify that the &lt;OTPK_EUICC_ECKA&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_RETRY_CC.</p>	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 RQ57_033 RQ57_039 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+ <ul style="list-style-type: none"> <li>#GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP are sent to the eUICC</li> <li>the same GSMA CI as for the first attempt has been chosen for signing and for verification</li> </ul>		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	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

			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.	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+ <ul style="list-style-type: none"> <li>#GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP are sent to the eUICC</li> <li>the same GSMA CI as for the first attempt has been chosen for signing and for verification</li> </ul>		RQ57_047

4	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT( #PREP_DOWNLOAD_RETRY_C C)	#R_PREP_DOWNLOAD_WITH_C C  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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP have been sent to the eUICC</li> <li>• the same GSMA CI has been chosen for signing and for verification</li> </ul>

**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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA_SCRIPT( #PREP_DOWNLOAD_WITH_C C)	<p>#R_PREP_DOWNLOAD_WITH_CC SW=0x9000</p> <p>The &lt;EUICC_SIGNATURE2&gt; SHALL be verified with the #PK_EUICC_ECDSA.</p> <p>Verify that the &lt;S_TRANSACTION_ID&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_WITH_CC.</p> <p>Verify that the &lt;S_HASHED_CC&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_WITH_CC.</p> <p>Extract the &lt;OTPK_EUICC_ECKA&gt; and reuse the same value in step 4</p>	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+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP are sent to the eUICC</li> <li>• the same GSMA CI as for the first attempt has been chosen for signing and for verification</li> </ul>		RQ57_047	
3	<S_HASHED_CC> = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE2, <S_TRANSACTION_ID>)			
4	S_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA_SCRIPT( #PREP_DOWNLOAD_RETRY_C CC)	<p>#R_PREP_DOWNLOAD_WITH_CC SW=0x9000</p> <p>The &lt;EUICC_SIGNATURE2&gt; SHALL be verified with the #PK_EUICC_ECDSA.</p> <p>Verify that the &lt;S_TRANSACTION_ID&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_RETRY_CC.</p> <p>Verify that the &lt;S_HASHED_CC&gt; present in the euiccSigned2 is the same as in #PREP_DOWNLOAD_RETRY_CC.</p> <p>Verify that the &lt;OTPK_EUICC_ECKA&gt; present in the euiccSigned2 is NOT the same as in #PREP_DOWNLOAD_RETRY_CC.</p>	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

#### TC\_eUICC\_ForbiddenPPRs

5 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_LPA → eUICC	MTD_STORE_DATA( #GET_EUICC_INFO2)	forbiddenProfilePolicyRules in EUICCInfo2 does not contain ppr1	RQ57_053 RQ57_054 RQ57_056 RQ57_057
2	Install PROFILE_OPERATIONAL1			RQ57_057
3	S_LPA → eUICC	MTD_STORE_DATA( #GET_EUICC_INFO2)	forbiddenProfilePolicyRules in EUICCInfo2 contains ppr1(1)	RQ57_053 RQ57_054 RQ57_056
4	Execute the Common Mutual Authentication procedure between the eUICC and the S_SM-DP+ <ul style="list-style-type: none"> <li>• #GET_EUICC_INFO1, #GET_EUICC_CHALLENGE and #AUTHENTICATE_SMDP are sent to the eUICC</li> <li>• the same GSMA CI is chosen for signing and for verification</li> </ul>			
5	Execute the Sub-procedure Profile Download and Installation – End User Confirmation between the eUICC and the S_SM-DP+			



	<ul style="list-style-type: none"> <li>• #PREP_DOWNLOAD_NO_CC is sent to the eUICC</li> </ul>			
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: <ul style="list-style-type: none"> <li>• &lt;BPP_SEG_INIT&gt;</li> <li>• &lt;BPP_SEG_A0&gt;</li> <li>• &lt;BPP_SEG_A1&gt;</li> <li>• &lt;BPP_SEG_A3&gt;</li> </ul>			
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_025 RQ25_023 RQ57_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_053 RQ57_054 RQ57_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**

**TC\_eUICC\_GetProfilesInfo\_GetRAT\_RSPSession**

**Test Sequence #01 Nominal: GetProfilesInfo and GetRAT during RSP session**

5.2.3.2.1 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: <ul style="list-style-type: none"> <li>• one PPAR authorizing PPR1 and PPR2 for all MNOs with End User consent required (i.e. #PPRS_ALLOWED)</li> <li>• no additional rules</li> </ul>
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: <ul style="list-style-type: none"> <li>• &lt;S_TRANSACTION_ID&gt;</li> <li>• &lt;EUICC_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_CHALLENGE&gt;</li> <li>• &lt;S_SMDP_SIGNATURE1&gt;</li> <li>• Set the &lt;EUICC_CI_PK_ID_TO_BE_USED&gt; to the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;</li> <li>• Choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate</li> </ul>		
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

#### TC\_eUICC\_DeleteProfile\_ISDP\_And\_Components

##### 5.2.5.2.1

#### **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_LPA → 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_LPA → 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_INSTALLE D> Verify that <FREE_MEM_OP_PROF1_INSTALLE D> 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_LPA → 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_INSTALLE D>	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: <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP2auth_ECDSA SHALL be set in #AUTH_SMDP_MATCH_ID rather than #CERT_S_SM_DPauth_ECDSA</li> <li>• #TEST_DP_ADDRESS2 SHALL be set in #AUTH_SMDP_MATCH_ID rather than #TEST_DP_ADDRESS1</li> <li>• #CERT_S_SM_DP2pb_ECDSA SHALL be set in #PREP_DOWNLOAD_NO_CC rather than #CERT_S_SM_DPpb_ECDSA</li> </ul>			
1	S_LPAAd → 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_LPAAd → 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

#### TC\_eUICC\_DisableProfile\_ApplicationManagement

General Initial Conditions	
Entity	Description of the general initial condition
5.2.7.2.1 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

#### 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_LPA → 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_LPAAd → 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_LPAAd → 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+	<ul style="list-style-type: none"> <li>• SM-DP+ is configured with the #CERT_SM_DPauth_ECDSA for NIST.</li> <li>• PROFILE_OPERATIONAL1 (configured with metadata as specified in each sequence) is securely loaded as a Protected Profile Package using &lt;PPK_ENC&gt; and &lt;PPK_MAC&gt;.</li> <li>• Pending Profile PROFILE_OPERATIONAL1 is in the 'Released' state with an empty MatchingID.</li> <li>• EID #EID1 is known to the SM-DP+ and associated to PROFILE_OPERATIONAL1.</li> <li>• Confirmation Code is not provided by the Operator to the SM-DP+.</li> </ul> <p>NOTE: the Profile Metadata for PROFILE_OPERATIONAL1 SHALL be specified in the Initial Conditions for each individual sequence.</p>

**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_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH,  MTD_INITIATE_AUTHENTICAT ION( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AU TH_OK)	
1	S_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT,  MTD_AUTHENTICATE_CLIEN T( <S_TRANSACTION_ID>,  #AUTH_SERVER_RESP_DEF_ DP_UC_OK))	MTD_HTTP_RESP(PARAM_R_AUT H_CLIENT)	RQ44_002
2	S_LPA <sub>d</sub> → 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_R ESP_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 notification Configuration Info 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_LPA → 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_LPA → 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_LPA → 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\_072, RQ31\_077, RQ31\_079, RQ31\_096, RQ31\_100, RQ31\_102, RQ31\_106, RQ31\_108, RQ31\_112, RQ31\_161
- RQ32\_001, RQ32\_002, RQ32\_003, RQ32\_004, RQ32\_062, RQ32\_065, 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**

**TC\_LPAd\_AddProfile\_Manual\_Entry**

5 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 Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in	RQ31_062 RQ31_106 RQ32_001 RQ32_002

			SGP.21 [3], if not verified before.	RQ32_065 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	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_062 RQ31_106 RQ32_065 RQC1_008 RQC3_014
9	PROC_ES9+_HANDLE_NOTIF			
10	S_EndUser →	Initiate List Profile operation	PROFILE_OPERATIONAL1 is	RQ32_069

	LPAAd		displayed in Disabled state	RQ32_070 RQ32_071 RQ44_001 RQC1_006
NOTE 1: The LPAAd 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.				

### TC\_LPAAd\_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→ LPAAd	Initiate Add Profile operation	LPAAd requests the Activation Code from the End User	RQ32_062 RQ32_066 RQC1_009
2	S_EndUser→ LPAAd	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	LPAAd → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ32_065 RQC1_008 RQC3_014
8	PROC_ES9+_HANDLE_NOTIF			
9	S_EndUser→ LPAAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state	RQ32_069 RQ32_070 RQ32_071 RQ44_001 RQC1_006

NOTE 1: The LPA<sub>d</sub> 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.

**TC\_LPA<sub>d</sub>\_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 → LPA <sub>d</sub>	Initiate Add Profile operation	Activation Code is requested from the End User by LPA <sub>d</sub>	RQ32_062 RQ32_066
2	S_EndUser → LPA <sub>d</sub>	Provide #ACTIVATION_CODE_INVALID_FORMAT by scanning the QR code	LPA <sub>d</sub> provides an error message to the EndUser	RQ31_072
3	S_EndUser → LPA <sub>d</sub>	Initiate List Profile operation	PROFILE_OPERATIONAL1 is not displayed	RQ31_072

5.4.1.2.4

**TC\_LPA<sub>d</sub>\_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 → LPA <sub>d</sub>	Initiate Add Profile operation		RQ32_062 RQ32_066
2	S_EndUser	Provide	LPA <sub>d</sub> provides an error	RQ31_072



	→ LPAAd	#ACTIVATION_CODE_INVALID_FORMA T by manual entry	message to the EndUser	
3	S_EndUser → LPAAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is not displayed	RQ31_072

**TC\_LPAAd\_AddProfile\_ConfirmationCode\_smdpSigned2\_QR**

General Initial Conditions	
Entity	Description of the general initial condition
5 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→ LPAAd	Initiate Add Profile operation	Activation Code is requested from the End User by LPAAd	RQ32_062 RQ32_066
2	S_EndUser→ LPAAd	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	LPAAd → 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	LPAAd → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ32_065
9	PROC_ES9+_HANDLE_NOTIF			
10	S_EndUser→ LPAAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state	RQ32_069 RQ32_070 RQ32_071
NOTE 1: The LPAAd 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.				

### 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	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ32_065
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.				

**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 Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ32_065
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
<p>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.</p> <p>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.</p>				

### 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_06 2 RQ32_06 6
2	S_EndUser→ LPAd	Provide#ACTIVATION_CODE_3 by scanning the QR code		RQ41_00 1 RQ41_00 5
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_10 8
7		PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ32_065
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.				

### TC\_LPAd\_AddProfile\_PPRs

#### Test Sequence #01 Nominal: End User Confirmation after PPR1 consent requested

5 Initial Conditions	
Entity	Description of the initial condition
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.	<p>The LPA provides means for the End User Confirmation/Rejection of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation either at this point or at a previous point of the procedure</p> <p>If Authenticated Confirmation was requested at a previous point, simple End User Confirmation/Rejection is requested.</p> <p>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)</p>	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 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
LPAAd	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	LPAAd → S_EndUser	Request for Confirmation if not requested before.	<p>The LPA provides means for the End User Confirmation/Rejection of the Profile Download as defined in SGP.21 [3] for Authenticated Confirmation either at this point or at a previous point of the procedure</p> <p>If Authenticated Confirmation was requested at a previous point, simple End User Confirmation/Rejection is requested.</p> <p>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.</p> <p>(See NOTE)</p>	RQ29_007_1 RQ29_008 RQ29_009 RQ29_015 RQ31_096 RQ31_100 RQ31_102 RQ29_011 RQ29_013
2	S_EndUser → LPAAd	End User Confirmation is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT		
3	PROC_ES9+_HANDLE_NOTIF			
4	S_EndUser → LPAAd	List Profile operation is initiated	PROFILE_OPERATIONAL3 is displayed in Disabled state	RQ31_161
NOTE: The request for this End User consent for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt.				

### TC\_LPAd\_LUI\_access\_protected

General Initial Conditions	
Entity	Description of the general initial condition
Device	The protection of access to the LUI is enabled.

#### Test Sequence #01 Nominal: Add a new Operational Profile

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	Enter the LUI	End User Intent verification for Authenticated Confirmation is requested.	RQ32_003
2	S_EndUser → LPAd	Initiate Add Profile operation	LPAd requests the Activation Code from the End User	RQ32_062 RQ32_066
3	S_EndUser → LPAd	Provide #ACTIVATION_CODE_1 by manual entry		RQ31_064 RQ31_077 RQ41_001
4	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+			
5	PROC_ES9+_INIT_AUTH			
6	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>			
7	PROC_ES9+_GET_BPP (see NOTE 1)			
8	LPAd → S_EndUser	Request for Authenticated or Simple Confirmation, if not requested before.	End User Intent successfully verified for Authenticated or Simple Confirmation as defined in SGP.21 [3], if not verified before.	RQ32_001 RQ32_003
9	PROC_ES9+_HANDLE_NOTIF			
10	S_EndUser → LPAd	List Profile operation is initiated	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.				

### 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 Authenticated Confirmation, if not requested before. The End User SHALL not provide Authenticated Confirmation.	The LPAd stops the Add Profile procedure	RQ32_001 RQ32_002 RQ32_004 RQ32_065
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. NOTE 2: Step IC6 is conditional – occurs only if Step 1 (Request for Confirmation) was not executed before.				

### 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 → LPAAd	Initiate Add Profile operation		RQ32_062 RQ32_066 RQC1_009
2	S_EndUser → LPAAd	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	LPAAd → S_EndUser		End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ32_065
8	PROC_ES9+_HANDLE_NOTIF			
9	S_EndUser → LPAAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state	RQ32_069 RQ32_070 RQ32_071
NOTE 1: The LPAAd 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	Initiate Add Profile operation		RQ32_062

	→ LPAAd			RQ32_066 RQC1_009
2	S_EndUser → LPAAd	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	LPAAd → S_EndUser		End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ32_065
8	PROC_ES9+_HANDLE_NOTIF			
9	S_EndUser → LPAAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state	RQ32_069 RQ32_070 RQ32_071
NOTE 1: The LPAAd 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

#### TC\_LPAAd\_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 → LPAAd	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\_LPAAd\_SetNickname**

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: Add a Nickname on 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 not defined .

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser→ LPAAd	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	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before.  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 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	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before.  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 of PROFILE_OPERATIONAL2 equals to #NICKNAME3	RQ32_078
5	Power off then power on the Device			
5.4.3.2.2	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

**TC\_LPAd\_EditNickname**

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: Edit the Nickname on 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
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	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before. 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 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	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before. LPAd sets the Profile Nickname	RQ32_073 RQ32_076

			(No Error)	
3	Exit the UI menu			
4	S_EndUser→ LPAAd	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→ LPAAd	Perform an LUI dependent action to display the NickName of PROFILE_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\_043, RQ32\_044, RQ32\_047, RQ32\_050
- RQ35\_008

### 5.4.4.2 Test Cases

#### 5.4.4.2.1 TC\_LPAAd\_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.
eUICC	The PROFILE_OPERATIONAL2 is in Disabled state.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAAd	Delete Profile procedure is initiated for PROFILE_OPERATIONAL1	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation.	RQ32_001 RQ32_002 RQ32_043 RQ32_044

2	LPAAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF1 is sent by the LPAAd	The delete Notification as defined below is received by the S_SM-DP+ within the timeout #IUT_LPAAd_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 → LPAAd	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.				

### TC\_LPAAd\_DeleteProfile\_Enabled\_without\_PPR

<b>5 General Initial Conditions</b>	
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

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 → LPAAd	Initiate Delete Profile procedure for PROFILE_OPERATIONAL5	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation.	RQ32_043 RQ32_044 RQ32_047
2	LPAAd → 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_LPAAd_NOTIFICATION_TIMEOUT MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS5) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the	RQ35_008 RQ35_015

			#PK_EUICC_ECDSA See NOTE	
3	LPAAd → 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_TIMEO UT 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 → LPAAd	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.				

#### 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 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→ LPAAd	Delete Profile procedure is initiated for PROFILE_OPERATIONAL4	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation. See NOTE 1 and NOTE 2	RQ32_043 RQ32_044 RQ32_047 RQ32_050
2	S_EndUser→ LPAAd	Request for List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL4 is	RQ32_058



			shown in Enabled state.	
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.				

**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 PROFILE_OPERATIONAL7 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL7 is in Disabled state.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser→ LPAd	Delete Profile procedure is initiated for PROFILE_OPERATIONAL7	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation. See NOTE 1 and NOTE 2	RQ32_043 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.				

**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 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→ LPAAd	Initiate Delete Profile procedure for PROFILE_OPERATIONAL8	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation. See NOTE 2 and NOTE 3	RQ32_043 RQ32_044 RQ32_047 RQ32_050
2	LPAAd → 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_LPAAd_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→ LPAAd	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 LPAAd 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 LPAAd MAY display an error indicating that the deletion of the Profile is failed.

**5.4.4.2.6 TC\_LPAAd\_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.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPA	Delete Profile procedure is initiated for PROFILE_OPERATIONAL1.  The End User SHALL not provide Authenticated Confirmation.	The LPA stops the Delete Profile procedure.	RQ32_001 RQ32_002 RQ32_004 RQ32_043
2	S_EndUser → LPA	Request for List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL1 is shown.	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**

**TC\_LPA\_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**

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.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL5	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation. 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 End User Intent verification.				

54.5.2.2

**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**

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 → LPAAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL6	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation.	RQ32_006 RQ32_007
2	LPAAd → S_SM-DP+(1)	Disable Notification containing #ICCID_OP_PROF5 is sent by the LPAAd	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_LPAAd_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	LPAAd → 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_LPAAd_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 → LPAAd	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
<p>NOTE 1: The Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel.</p> <p>NOTE 2: The timeout (steps 2 and 3) SHALL start after the End User Intent verification.</p>				

### 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	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation.	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 PROFILE_OPERATIONAL4 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL1	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation. See NOTE 1 and NOTE 2	RQ32_006 RQ32_007 RQ32_008 RQ32_014

2	S_EndUser → LPAAd	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
<p>NOTE 1: The LPAAd 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.</p> <p>NOTE 2: The LPAAd MAY display an error indicating that the enabling of the Profile is failed.</p>				

#### 5.4.5.2.5 TC\_LPAAd\_EnableProfile\_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 Enable Profile Operation if No Confirmation Provided*

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 → LPAAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL1 The End User SHALL not provide Simple Confirmation.	The LPAAd stops the Enable Profile procedure.	RQ32_001 RQ32_002 RQ32_004 RQ32_006
2	S_EndUser → LPAAd		PROFILE_OPERATIONAL1 is shown in Disabled state.	RQ32_004

### 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	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation. 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

#### 5.4.6.2.2

NOTE: The timeout SHALL start after the End User Intent verification.

#### 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→ LPAAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL1	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation	RQ32_025 RQ32_026
2	LPAAd → S_EndUser	Result of the Profile disabling	The Disable Profile procedure terminates indicating a failure	RQ32_034

**TC\_LPAAd\_DisableProfile\_Error\_PPR1Set**

5

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 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 → LPAAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL4	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation. See NOTE 1 and NOTE 2	RQ32_025 RQ32_026 RQ32_028 RQ32_034
2	S_EndUser → LPAAd	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
<p>NOTE 1: The LPAAd 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.</p> <p>NOTE 2: The LPAAd MAY display an error indicating that the disabling of the Profile is failed.</p>				

General Initial Conditions	
Entity	Description of the general initial condition
Device	The protection of access to the LUI is disabled.

**Test Sequence #01 Nominal: Stop Disable Profile Operation if No Confirmation Provided**

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 → LPAAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL1. The End User SHALL not provide Simple Confirmation.	The LPAAd stops the Disable Profile procedure.	RQ32_001 RQ32_002 RQ32_004 RQ32_025
2	S_EndUser → LPAAd		Installed Operational Profile(s) with their current states are displayed in a human readable format. PROFILE_OPERATIONAL1 is shown in Enabled state.	RQ32_004

**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

**5.4.7.2.1**

**5.4.7.2 Test Cases**

**TC\_LPAAd\_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	<p>The LPA presents the #EID1 to the End User as a text string and/or as a QR code.</p> <p>If the EID is represented as text string, the text SHALL be identical to #EID1</p> <p>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.</p>	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\_006, RQ33\_012, RQ33\_021\_1, RQ33\_021\_2
- RQ35\_008, RQ35\_018, RQ35\_019

### 5.4.8.2 Test Cases

#### 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.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAAd	Initiate the eUICC Memory Reset for operational profiles	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation.	RQ33_006
2	LPAAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF1 is sent by the LPAAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1) is received by the S_SM-DP+ within the timeout #IUT_LPAAd_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 → LPAAd	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 PROFILE_OPERATIONAL1 is installed on the eUICC with #METADATA_OP_PROF1_MEMRES1.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAAd	Initiate the eUICC Memory Reset for operational profiles	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation.	RQ33_006
2	LPAAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF1 is sent by the LPAAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1) is received by the S_SM-DP+ within the timeout #IUT_LPAAd_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 → LPAAd		Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available	RQ32_053 RQ33_012

#### 5.4.8.2.2 TC\_LPAAd\_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 PPR1 and PPR2**

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') and PPR2 ('Deletion of this Profile is not allowed').

Initial Conditions	
Entity	Description of the initial condition
eUICC	The PROFILE_OPERATIONAL5 is installed on the eUICC with #METADATA_OP_PROF5_MEMRES2.
eUICC	The PROFILE_OPERATIONAL5 is in Enabled state.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAAd	Initiate the eUICC Memory Reset for operational profiles	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation.	RQ33_006
2	LPAAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF5 is sent by the LPAAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL5) is received by the S_SM-DP+ within the timeout (#IUT_LPAAd_NOTIFICATION_TIMEOUT, + #IUT_LPAAd_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.		RQ33_011 RQ33_012
4	S_EndUser → LPAAd		Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available	RQ32_053 RQ33_012
<p>NOTE 1: The Delete Notification (step 2) can be sent at any step after having successfully initiated the eUICC Memory Reset.</p> <p>NOTE 2: The timeout (step 2) SHALL start after the End User Intent verification.</p> <p>NOTE 3: A Disable Notification for PROFILE_OPERATIONAL5 MAY be sent before the Delete Notification. This notification SHALL NOT be checked.</p>				

**Test Sequence #02 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_OPERATIONAL5 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL5 is in Enabled state.
eUICC	The PROFILE_OPERATIONAL6 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL6 is in Disabled state.

Step	Direction	Sequence / Description	Expected result	REQ
1	S_EndUser → LPAAd	Initiate the eUICC Memory Reset for operational profiles	Successful End User Intent verified as defined in SGP.21 [3] for Authenticated Confirmation.	RQ33_006
2	LPAAd → S_SM-DP+(1)	Delete Notifications containing #ICCID_OP_PROF5 is sent by the LPAAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL5) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS1) within the timeout #IUT_LPAAd_NOTIFICATION_TIMEOUT The S_SM-DP+ SHALL return #R_HTTP_204_OK	RQ35_008 RQ35_018 RQ35_019
3	LPAAd → S_SM-DP+(2)	Delete Notification containing #ICCID_OP_PROF6 is sent by the LPAAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL6) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS2) within the timeout (#IUT_LPAAd_NOTIFICATION_TIMEOUT, + #IUT_LPAAd_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	RQ35_008 RQ35_018 RQ35_019
4	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.		RQ33_011 RQ33_012
5	S_EndUser → LPAAd		Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available	RQ32_053 RQ33_012
<p>NOTE 1: The Delete Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel and can be sent at any step after having successfully initiated the eUICC Memory Reset.</p> <p>NOTE 2: The timeout (steps 2 and 3) SHALL start after the End User Intent verification.</p> <p>NOTE 3: A Disable Notification for PROFILE_OPERATIONAL5 MAY be sent before the Delete Notification. This notification SHALL NOT be checked.</p>				

#### 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

#### 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		Successful End User Intent verified as defined in SGP.21 [4] for Simple Confirmation, if not verified before.	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	RQ33_021_5



			shown	
--	--	--	-------	--

**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 → LPA	Initiate the function to retrieve the configured address	The LPA 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 → LPA		Successful End User Intent verified as defined in SGP.21 [4] for Simple Confirmation, if not verified before.	RQ33_021_3
3	S_EndUser → LPA	Initiate the function to retrieve the configured address	The LPA 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

**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 → LPA	Initiate the function to retrieve the configured address	The LPA retrieves the Default SM-DP+ Address and presents it to the EndUser	RQ33_021_2

			The current Default SM-DP+ Address is #TEST_DEFAULT_DP_ADDRESS_1	
2	S_EndUser → LPAAd	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+.	Successful End User Intent verified as defined in SGP.21 [4] for Simple Confirmation, if not verified before.	RQ33_021_3
3	S_EndUser → LPAAd	Initiate the function to retrieve the configured address	The LPAAd 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

### 5.4.11 Device Power On – Profile Discovery

#### 5.4.11.1 Conformance Requirements

##### References

GSMA RSP Technical Specification [2]

##### Requirements

- RQ31\_106
- RQ34\_18, RQ34\_020, RQ34\_021, RQ34\_023, RQ34\_024

#### 5.4.11.2 Test Cases

##### 5.4.11.2.1

##### TC\_LPAAd\_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	LPA → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ34_023 RQ34_024
6	PROC_ES9+_HANDLE_NOTIF			
7	LPA → S_EndUser	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state	RQ34_018 RQ34_020
NOTE 1: The LPA 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\_LPA → 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	LPA → S_EndUser	Request for Authenticated Confirmation, if not requested before.	End User Intent successfully verified for Authenticated Confirmation as defined in SGP.21 [3], if not verified before.	RQ31_106 RQ34_023 RQ34_024
9	PROC_ES9+_HANDLE_NOTIF			
10	LPA → S_EndUser	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state	RQ34_018 RQ34_021
NOTE 1: The LPA 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.				

## End-to-End Testing

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

### 6 7 External Test Specifications

Some test specifications related to the RSP ecosystem have been developed by external organisations (e.g. 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 SIMAlliance eUICC Profile Package Test Specification

The SIMAlliance eUICC Profile Package: Interoperable Format Test Specification [23] SHALL be executed on the eUICC in order to check its compliance with the SIMAlliance eUICC Profile Package: Interoperable Format Technical Specification [4].

Test cases are applicable according to the applicability table of the referred Test Specification [23].

The table below describes the restrictions on the SIMAlliance tests applicability depending on the SGP.22 version supported by the eUICC:

SGP.22 version	SIMAlliance [4] version indicating which test cases are applicable for the given SGP.22 version
2.1	2.0 or 2.1
2.2.x	2.1

Moreover, eUICC Manufacturers SHALL declare that the following SIMAlliance options are supported by the eUICC:

- O\_MILENAGE
- O\_TUAK\_128
- O\_JAVACARD

The successful execution of SIMAlliance 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_EVENT_ID (CtxParams1)	ctxParamsForCommonAuthentication : { matchingId #EVENT_ID_1, #S_DEVICE_INFO

Name	Content
	}
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	GSMA Test CI
EF_UST1	0x0A 2E 14 8C E7 32 04 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 -- 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 29 90 00 11 23 41 23 40 12 34 56 78 90 13 53
EUICC_CI_PK_ID_LIST_FOR_SIGNING_ 1	#CI_PKI_ID1,

Name	Content
	#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	<pre> 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   } }                     </pre>
EUICC_INFO1_8_8_3_3_1_HIGHER	<pre> 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   } }                     </pre>
EUICC_INFO1_8_8_3_3_1_LOWER	<pre> 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   } }                     </pre>
EUICC_INFO1_8_8_4_3_7	<pre> 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   } }                     </pre>
EUICC_SIGNED1	<pre> {   transactionId &lt;S_TRANSACTION_ID&gt;,   serverAddress #TEST_DP_ADDRESS1,   serverChallenge &lt;S_SMDP_CHALLENGE&gt;,   euiccInfo2 #R_EUICC_INFO2, -- check only that   the field is present and has a valid TLV asn.1 }                     </pre>



Name	Content
	<pre> structure   ctxParams1 #CTX_PARAMS1 }                     </pre>
EVENT_ID_1	07399-BGH7E-T8779
EVENT_ID_2	07399-BGH7E-T8778
EXT_SHA256_RSA	TLS extension data for "supported_signature_algorithms" set as: <ul style="list-style-type: none"> <li>o HashAlgorithm sha256 (04) and</li> <li>o SignatureAlgorithm rsa (01).</li> </ul>
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_PROF2	0x98 92 09 01 32 54 76 98 10 F9
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

Name	Content
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 88 43 65 87 09 21
IMSI_OP_PROF8	0x08 29 99 88 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_MNC8	0x92 F9 88
MCC_MNC9	0x92 F9 98

Name	Content
MIN_TLS_CIPHER_SUITES	The minimum TLS cipher suites proposed by the Client: <ul style="list-style-type: none"> <li>○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</li> <li>○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</li> </ul>
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_SESSION	/gsma/rsp2/es9plus/cancelSession
PATH_DELETE_EVENT	/gsma/rsp2/es12/deleteEvent
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

Name	Content
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
PROP_TLS_CIPHER_SUITES	The TLS cipher suites proposed by the Client: <ul style="list-style-type: none"> <li>o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</li> <li>o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</li> <li>o TLS_RSA_WITH_AES_128_CBC_SHA</li> <li>o TLS_RSA_WITH_AES_256_CBC_SHA256</li> <li>o TLS_DHE_RSA_WITH_AES_256_CBC_SHA256</li> </ul>
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
S_DEVICE_INFO	<pre>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   } }</pre>
S_DEVICE_INFO_IMEI	<pre>deviceInfo {   tac #S_TAC,   deviceCapabilities {     gsmSupportedRelease '050000'H,     utranSupportedRelease '080000'H,     cdma2000onexSupportedRelease '010000'H,     eutranSupportedRelease '020000'H   },   imei #S_IMEI }</pre>
S_EUICC_CHALLENGE	0x01 02 03 04 05 06 07 08 01 02 03 04 05 06 07 08

Name	Content
S_EUICC_CHALLENGE_2	0x21 22 23 24 25 26 27 28 21 22 23 24 25 26 27 28
S_EUICC_INFO1	<pre> euiccInfo1 EUICCInfo1 ::= {   svn #RSP_SVN,   euiccCiPKIdListForVerification {     #EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1   },   euiccCiPKIdListForSigning {     #EUICC_CI_PK_ID_LIST_FOR_SIGNING_1   } }                     </pre>
S_EUICC_INFO2	<pre> 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 }                     </pre>
S_EXT_SHA256_ECDSA	TLS extension data for "supported_signature_algorithms" set as: <ul style="list-style-type: none"> <li>o HashAlgorithm sha256 (04) and</li> <li>o SignatureAlgorithm ecdsa (03).</li> </ul>
S_IMEI	0x00 00 00 00 11 11 11 11
S_SAH_SHA256_ECDSA	Signature And Hash Algorithm extension sent in the CertificateRequest message set as: <ul style="list-style-type: none"> <li>o HashAlgorithm sha256 (04) and</li> <li>o SignatureAlgorithm rsa (01).</li> </ul>
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

Name	Content
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: <ul style="list-style-type: none"> <li>• #IUT_SM_DP_ADDRESS</li> <li>• #IUT_SM_DS_ADDRESS</li> </ul>
SIMA_RESULT_OK	<pre> simaresp EUICCRresponse ::= {   peStatus {     {status ok}   } }                     </pre>
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.gsma.com
TEST_DEFAULT_DP_ADDRESS_1	testdefaultsdmplus1.gsma.com
TEST_DP_ADDRESS1	testsmdplus1.gsma.com
TEST_DP_ADDRESS2	testsmdplus2.gsma.com
TEST_DP_ADDRESS3	testsmdplus3.gsma.com
TEST_DP_ADDRESS4	testsmdplus4.gsma.com
TEST_DP_ADDRESS8	testsmdplus8.gsma.com
TEST_DS_ADDRESS1	testsmds1.gsma.com
TEST_ROOT_DS_ADDRESS	testrootsmds.gsma.com
TLS_VERSION_1_1	1.1
TLS_VERSION_1_2	1.2 The minimum TLS Version supported by the Server.

Name	Content
UNKNOWN_BPP_SEGMENT	0xC9 05 01 02 03 04 05
UNKNOWN_SERVER_ADDRESS	unknownserver.gsma.com
UNSUP_TLS_CIPHER_SUITES	The TLS cipher suites proposed by the Client: <ul style="list-style-type: none"> <li>○ TLS_RSA_WITH_AES_128_CBC_SHA</li> <li>○ TLS_RSA_WITH_AES_256_CBC_SHA256</li> </ul>
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

## 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. All test keys and test certificates used in the present document are contained in SGP.26 v1.2 Files, which accompanies the present document.

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: <ul style="list-style-type: none"> <li>• #CERT_SM_DP_TLS</li> <li>• #CERT_SM_DS_TLS</li> <li>• #CERT_S_SM_DP_TLS</li> <li>• #CERT_S_SM_DS_TLS</li> </ul>
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

<p>CERT_EUICC_ECDSA_EID2</p>	<p>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.</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUICC_ECDSA_EXPIRED</p>	<p>RSP Certificate of the eUICC (CERT.EUICC_ECDSA) set as a fixed test CERT with 13th January 2016 set in the validity field.</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUICC_ECDSA_INVALID_EX_CP</p>	<p>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".</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUICC_ECDSA_INVALID_EX_KU</p>	<p>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".</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUICC_ECDSA_INVALID_SIG</p>	<p>RSP Certificate of the eUICC (CERT.EUICC_ECDSA) set as a fixed test CERT with an invalid signature in the signatureValue field.</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUICC_ECDSA_INVALID_SUB_ORG</p>	<p>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".</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUICC_ECDSA_INVALID_SUB_SN</p>	<p>RSP Certificate of the eUICC (CERT.EUICC_ECDSA) set as a fixed test CERT with an invalid 'serialNumber' attribute value in the subject field set as "89000000000000000000000000000000".</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUM_ECDSA</p>	<p>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.</p>
<p>CERT_EUM_ECDSA_EXPIRED</p>	<p>RSP Certificate of the eUICC (CERT.EUM_ECDSA) set as a fixed test CERT with 13<sup>th</sup> January 2016 set in the validity field.</p> <p>Depending on the eUICC configuration, this certificate is based on NIST P-256, brainpoolP256r1 or FRP256V1.</p>
<p>CERT_EUM_ECDSA_INVALID_EX_BC_CA</p>	<p>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".</p>



	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:  <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS</li> <li>• #CERT_S_SM_DS_TLS</li> </ul>
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:  <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_EXPIRED</li> <li>• #CERT_S_SM_DS_TLS_EXPIRED</li> </ul>
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:  <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_CERT_POL</li> <li>• #CERT_S_SM_DS_TLS_INV_CERT_POL</li> </ul>
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:  <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_CRITICAL_EXT</li> <li>• #CERT_S_SM_DS_TLS_INV_CRITICAL_EXT</li> </ul>

CERT_S_CLIENT_TLS_INV_EXT_KEY_USAGE	<p>CERT.CLIENT.TLS certificate of the S_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_EXT_KEY_USAGE</li> <li>• #CERT_S_SM_DS_TLS_INV_EXT_KEY_USAGE</li> </ul>
CERT_S_CLIENT_TLS_INV_KEY_USAGE	<p>CERT.CLIENT.TLS certificate of the S_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_KEY_USAGE</li> <li>• #CERT_S_SM_DS_TLS_INV_KEY_USAGE</li> </ul>
CERT_S_CLIENT_TLS_INV_OID	<p>CERT.CLIENT.TLS certificate of the S_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_OID</li> <li>• #CERT_S_SM_DS_TLS_INV_OID</li> </ul>
CERT_S_CLIENT_TLS_INV_SIG	<p>CERT.CLIENT.TLS certificate of the S_CLIENT, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_SIG</li> <li>• #CERT_S_SM_DS_TLS_INV_SIG</li> </ul>
CERT_S_SERVER_TLS	<p>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:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS on ES9+</li> <li>• #CERT_S_SM_DS_TLS on ES11 or ES12</li> </ul>
CERT_S_SERVER_TLS_EXPIRED	<p>CERT.SERVER.TLS certificate of the S_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_EXPIRED</li> <li>• #CERT_S_SM_DS_TLS_EXPIRED</li> </ul>
CERT_S_SERVER_TLS_INV_CERT_POL	<p>CERT.SERVER.TLS certificate of the S_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_CERT_POL</li> <li>• #CERT_S_SM_DS_TLS_INV_CERT_POL</li> </ul>
CERT_S_SERVER_TLS_INV_CRITICAL_EXT	<p>CERT.SERVER.TLS certificate of the S_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_CRITICAL_EXT</li> <li>• #CERT_S_SM_DS_TLS_INV_CRITICAL_EXT</li> </ul>
CERT_S_SERVER_TLS_INV_EXT_KEY_USAGE	<p>CERT.SERVER.TLS certificate of the S_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_EXT_KEY_USAGE</li> <li>• #CERT_S_SM_DS_TLS_INV_EXT_KEY_USAGE</li> </ul>
CERT_S_SERVER_TLS_INV_KEY_USAGE	<p>CERT.SERVER.TLS certificate of the S_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_KEY_USAGE</li> </ul>

	<ul style="list-style-type: none"> <li>• #CERT_S_SM_DS_TLS_INV_KEY_USAGE</li> </ul>
CERT_S_SERVER_TLS_INV_SIG	<p>CERT.SERVER.TLS certificate of the S_SERVER, where the Certificate MAY be one of the following depending on the role of the simulator:</p> <ul style="list-style-type: none"> <li>• #CERT_S_SM_DP_TLS_INV_SIG</li> <li>• #CERT_S_SM_DS_TLS_INV_SIG</li> </ul>
CERT_S_SM_DP_TLS	<p>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</p>
CERT_S_SM_DP2_TLS	<p>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.</p>
CERT_S_SM_DP4_TLS	<p>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.</p>
CERT_S_SM_DP8_TLS	<p>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.</p>
CERT_S_SM_DP_TLS_EXPIRED	<p>Expired CERT.DP.TLS certificate of the S_SM-DP+ with a valid signature, correctly formatted as X.509 certificate.</p>
CERT_S_SM_DP_TLS_INV_CERT_POL	<p>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.</p>
CERT_S_SM_DP_TLS_INV_CRITICAL_EXT	<p>CERT.DP.TLS certificate of the S_SM-DP+ with one of the critical extensions not present, formatted as X.509 certificate.</p>
CERT_S_SM_DP_TLS_INV_CURVE	<p>CERT.DP.TLS certificate of the S_SM-DP+, based on the different CI as defined in #IUT_LPAd_CI, not based on</p> <ul style="list-style-type: none"> <li>• NIST P-256 curve, defined in Digital Signature Standard [11]</li> <li>• brainpoolP256r1 curve, defined in RFC 5639 [8]</li> <li>• FRP256V1 curve, defined in ANSSI ECC [9]</li> </ul>
CERT_S_SM_DP_TLS_INV_EXT_KEY_USAGE	<p>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.</p>
CERT_S_SM_DP_TLS_INV_KEY_USAGE	<p>CERT.DP.TLS certificate of the S_SM-DP+ with invalid 'key usage' extension (not set to 'digitalSignature'), formatted as X.509 certificate.</p>
CERT_S_SM_DP_TLS_INV_OID	<p>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.</p>
CERT_S_SM_DP_TLS_INV_SIG	<p>Invalid CERT.DP.TLS certificate of the S_SM-DP+ with an invalid signature with the same tag and length as a valid</p>

	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: <ul style="list-style-type: none"> <li>• NIST P-256 curve, defined in Digital Signature Standard [11]</li> <li>• brainpoolP256r1 curve, defined in RFC 5639 [8]</li> <li>• FRP256V1 curve, defined in ANSSI ECC [9]</li> </ul>
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: <ul style="list-style-type: none"> <li>• NIST P-256 curve, defined in Digital Signature Standard [11]</li> <li>• brainpoolP256r1 curve, defined in RFC 5639 [8]</li> <li>• FRP256V1 curve, defined in ANSSI ECC [9]</li> </ul>
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: <ul style="list-style-type: none"> <li>• NIST P-256 curve, defined in Digital Signature Standard [11]</li> <li>• brainpoolP256r1 curve, defined in RFC 5639 [8]</li> <li>• FRP256V1 curve, defined in ANSSI ECC [9]</li> </ul>
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 <ul style="list-style-type: none"> <li>• NIST P-256 curve, defined in Digital Signature Standard [11]</li> <li>• brainpoolP256r1 curve, defined in RFC 5639 [8]</li> <li>• FRP256V1 curve, defined in ANSSI ECC [9]</li> </ul>
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: <ul style="list-style-type: none"> <li>• #CERT_SM_DP_TLS</li> <li>• #CERT_SM_DS_TLS</li> <li>• #CERT_S_SM_DP_TLS</li> <li>• #CERT_S_SM_DS_TLS</li> </ul>

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 S_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. <ul style="list-style-type: none"> <li>The Authority Key Identifier is set as #CI_PKI_ID1</li> </ul>
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. <ul style="list-style-type: none"> <li>The Authority Key Identifier is set as #CI_PKI_ID1</li> </ul>
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: <ul style="list-style-type: none"> <li>#CERT_SM_DPauth_ECDSA</li> <li>#CERT_SM_DSauth_ECDSA</li> </ul>
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: <ul style="list-style-type: none"> <li>#PK_S_SM_DP_TLS</li> </ul>

	<ul style="list-style-type: none"> <li>• #PK_S_SM_DS_TLS</li> </ul>
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: <ul style="list-style-type: none"> <li>• #PK_S_SM_DP_TLS on ES9+</li> <li>• #PK_S_SM_DS_TLS on ES11</li> </ul>
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: <ul style="list-style-type: none"> <li>• #PK_SM_DPauth_ECDSA</li> <li>• #PK_SM_DSauth_ECDSA</li> </ul>
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_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: <ul style="list-style-type: none"> <li>array first element containing the Tag and length fields of the sequenceOf88 TLV</li> <li>array following elements containing each of the '88' TLVs containing the StoreMetadata command</li> </ul>
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: <ul style="list-style-type: none"> <li>array first element containing the tag and length fields of the sequenceOf86 TLV</li> <li>array following elements containing each of the '86' TLVs containing the Protected Profile Package (PPP)</li> </ul>
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



Variable	Description
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 (4 bits) to be used for indication of additionalProfile, crlSupport, rpmSupport , testProfileSupport
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: <ul style="list-style-type: none"> <li>• euiccSigned2</li> <li>• smdpSignature2 present in the PrepareDownloadRequest structure</li> </ul>
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-DS during Event Record registration.
EVENT_ID_R	The EventID value in String format generated by the SM-DS during Event Record registration.
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

Variable	Description
	(03).
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.

Variable	Description
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-

Variable	Description
	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

Variable	Description
	1020 bytes including the tag and length field.
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
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: <ul style="list-style-type: none"> <li>o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 if present in &lt;TLS_CIPHER_SUITES&gt;, otherwise</li> <li>o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256.</li> </ul>
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: <ul style="list-style-type: none"> <li>• remoteOpId</li> <li>• transactionId</li> <li>• controlRefTemplate</li> <li>• smdpOtpk</li> <li>• euiccOtpk, as provided earlier in the prepareDownloadResponse data object</li> </ul>
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: <ul style="list-style-type: none"> <li>• smdpSignature2</li> <li>• euiccSignature1 present in the AuthenticateServerResponse structure</li> </ul>
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

Variable	Description
	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: <ul style="list-style-type: none"> <li>• HashAlgorithm sha256 (04) and</li> <li>• SignatureAlgorithm ecdsa (03).</li> </ul>

Variable	Description
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: <ul style="list-style-type: none"> <li>• &lt;SMDP_CHALLENGE&gt;</li> <li>• &lt;SMDS_CHALLENGE&gt;</li> </ul>
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: <ul style="list-style-type: none"> <li>• &lt;SMDP_CHALLENGE_2&gt;</li> <li>• &lt;SMDS_CHALLENGE_2&gt;</li> </ul>
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: <ul style="list-style-type: none"> <li>• SM-DP+ signature (serverSignature1) generated over #SERVER_SIGNED1 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING</li> <li>• SM-DS signature (serverSignature1) generated over #SERVER_SIGNED1 using SK.DSauth.ECDSA, coded as ASN.1 OCTET STRING</li> </ul>
SERVER_SIGNATURE1_2	SERVER signature (serverSignature1) which can be one of the following depending on the entity under test: <ul style="list-style-type: none"> <li>• SM-DP signature (serverSignature1) generated over #SERVER_SIGNED1_2 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING</li> <li>• SM-DS signature (serverSignature1) generated over #SERVER_SIGNED1_2 using SK.DSauth.ECDSA, coded as ASN.1 OCTET STRING</li> </ul>
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.

Variable	Description
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.
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 LPA 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



Variable	Description
	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.
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)	<ul style="list-style-type: none"> <li>paramTransactionId: random 16 byte identifier encoded as String Hexadecimal.</li> <li>paramAuthenticateServerResponse: server authentication response structured as ASN.1 encoded as base 64.</li> </ul>
Details	JSON body <pre>{   "transactionId" : paramTransactionId,   "authenticateServerResponse" :   paramAuthenticateServerResponse }</pre>

Method	MTD_CANCEL_SESSION
Description	Sends or verifies the JSON formatted CancelSession request
Parameter(s)	<ul style="list-style-type: none"> <li>paramTransactionId: random 16 byte identifier.</li> <li>paramCancelSessionResponse: eUICC information structured as ASN.1 encoded as base 64.</li> </ul>
Details	JSON body <pre>{   "transactionId" : paramTransactionId,   "cancelSessionResponse" : paramCancelSessionResponse }</pre>

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

	<pre>23 &lt;L&gt; 00 90 00 -- R-APDU of the INSTALL FOR PERSONALIZATION command 23 &lt;L&gt; paramExpectedSW -- SW of the last STORE DATA command executed</pre> <p>&lt;NB_EXECUTED_C_APDUS&gt; SHALL be equal to the number of executed C-APDUs          (i.e. one INSTALL FOR PERSONALIZATION + n STORE DATA command(s))</p>
--	--

<b>Method</b>	MTD_DELETE_EVENT
<b>Description</b>	Sends and checks the JSON formatted DeleteEvent request
<b>Parameter(s)</b>	<ul style="list-style-type: none"> <li>paramFunctionRequesterId: identification of the function requester.</li> <li>paramFunctionCallId: identification of the function call.</li> <li>paramEID: EID of the targeted eUICC</li> <li>paramEventId: unique Identification of the Event to be registered</li> </ul>
<b>Details</b>	<pre>JSON requestHeader {   "header" : {     "functionRequesterIdentifier" : "paramFunctionRequesterId",     "functionCallIdentifier" : "paramFunctionCallId"   } } JSON body {   "eid" : paramEID,   "eventId" : paramEventId } }</pre>

<b>Method</b>	MTD_DISABLE_PROFILE
<b>Description</b>	Generate the ASN.1 DisableProfileRequest structure according to the input parameters.
<b>Parameter(s)</b>	<ul style="list-style-type: none"> <li>paramIccidValue: The ICCID of the Profile to Disable (optional)</li> <li>paramIsdpAidValue: The ISD-P AID of the Profile to Disable (optional)</li> <li>paramRefreshFlag: Boolean, TRUE if refreshFlag SHALL be set, FALSE otherwise</li> </ul> <p>Either paramIccidValue or paramIsdpAidValue is passed as a parameter.</p>
<b>Details</b>	<pre>IF paramIccidValue is provided Then   req DisableProfileRequest ::= {     profileIdentifier iccid : paramIccidValue,     refreshFlag paramRefreshFlag   } Else   req DisableProfileRequest ::= {     profileIdentifier isdpAid : paramIsdpAidValue,     refreshFlag paramRefreshFlag   } End if</pre>

Method	MTD_ENABLE_PROFILE
Description	Generate the ASN.1 EnableProfileRequest structure according to the input parameters.
Parameter(s)	<ul style="list-style-type: none"> <li>paramIccidValue: The ICCID of the Profile to Disable (optional)</li> <li>paramIsdpAidValue: The ISD-P AID of the Profile to Disable (optional)</li> <li>paramRefreshFlag: Boolean, TRUE if refreshFlag SHALL be set, FALSE otherwise</li> </ul> Either paramIccidValue or paramIsdpAidValue is passed as a parameter.
Details	IF paramIccidValue is provided Then <pre> req EnableProfileRequest ::= {   profileIdentifier iccid : paramIccidValue,   refreshFlag paramRefreshFlag }                     </pre> Else <pre> req EnableProfileRequest ::= {   profileIdentifier isdpAid : paramIsdpAidValue,   refreshFlag paramRefreshFlag }                     </pre> End if

Method	MTD_DELETE_PROFILE
Description	Generate the ASN.1 DeleteProfileRequest structure according to the input parameters.
Parameter(s)	<ul style="list-style-type: none"> <li>paramIccidValue: The ICCID of the Profile to Delete (optional)</li> <li>paramIsdpAidValue: The ISD-P AID of the Profile to Delete (optional)</li> </ul> Either paramIccidValue or paramIsdpAidValue is passed as a parameter.
Details	IF paramIccidValue is provided Then <pre> req DeleteProfileRequest ::= iccid : paramIccidValue                     </pre> Else <pre> req DeleteProfileRequest ::= isdpAid : paramIsdpAidValue                     </pre> End if

Method	MTD_GET_PROFILE_INFO
Description	Generate the ASN.1 ProfileInfoListRequest according to the input parameters.
Parameter(s)	<ul style="list-style-type: none"> <li>paramIccidValue: The ICCID of the Profile</li> <li>paramIsdpAidValue: The ISD-P AID of the Profile</li> </ul> Either paramIccidValue or paramIsdpAidValue is passed as a parameter.
Details	IF paramIccidValue is provided Then <pre> req ProfileInfoListRequest ::= {   searchCriteria iccid: paramIccidValue }                     </pre> Else <pre> req ProfileInfoListRequest ::= {   searchCriteria isdpAid: paramIsdpAidValue }                     </pre>

	<pre>         }     End If     </pre>
--	---------------------------------------

Method	MTD_GENERATE_BPP
Description	Generate a BPP according to the input parameters.
Parameter(s)	<ul style="list-style-type: none"> <li>• paramInitSC: The InitialiseSecureChannel request</li> <li>• paramConfISDP: The ConfigureISDP request (plain)</li> <li>• paramStoreMetadata: The StoreMetadata request (plain)</li> <li>• paramReplaceSessionKeys: The ReplaceSessionKeys request (plain) – Optional parameter</li> <li>• paramUPP: The Unprotected Profile Package to download</li> </ul>
Details	<p>Split the paramStoreMetadata in several segments of maximum 1008 bytes. Each Metadata segment is named &lt;METADATA_SEG&gt; here after.</p> <p>Split the paramUPP in several segments of maximum 1007 bytes. Each UPP segment named &lt;UPP_SEG&gt; here after.</p> <p>Create the following structure of data:</p> <pre> req BoundProfilePackage ::= {     paramInitSC,     firstSequenceOf87 {         0x87 &lt;L&gt; paramConfISDP     },     sequenceOf88 {         0x88 &lt;L&gt; &lt;METADATA_SEG&gt;,         ...         0x88 &lt;L&gt; &lt;METADATA_SEG&gt;     },     -- secondSequenceOf87 SHALL be set only if paramReplaceSessionKeys is     -- provided     secondSequenceOf87 {         0x87 &lt;L&gt; paramReplaceSessionKeys     },     sequenceOf86 {         0x86 &lt;L&gt; &lt;UPP_SEG&gt;,         ...         0x86 &lt;L&gt; &lt;UPP_SEG&gt;     } }     </pre> <p>Use &lt;OT_SK_S_SM_DP+_ECKA&gt; and &lt;OTPK_EUICC_ECKA&gt; in order to generate the &lt;SHS&gt;.</p> <p>Concatenate #KEY_TYPE, #KEY_LENGTH, &lt;L&gt; #HOST_ID and &lt;L&gt; #EID1 as SharedInfo.</p> <p>Retrieve &lt;S_ENC&gt;, &lt;S_MAC&gt; and &lt;S_INIT_MAC&gt; across SHA-256 calculated from &lt;SHS&gt; and SharedInfo.</p> <p>Encrypt paramConfISDP with &lt;S_ENC&gt;.</p> <p>Calculate and add a MAC to the tag 0x87 of firstSequenceOf87 by using &lt;S_MAC&gt;.</p> <p>Calculate and add a MAC to all tags 0x88 of sequenceOf88 by using &lt;S_MAC&gt;.</p>

	<p>If paramReplaceSessionKeys is provided Then</p> <p style="padding-left: 40px;">Encrypt paramReplaceSessionKeys with &lt;S_ENC&gt;</p> <p style="padding-left: 40px;">Calculate and add a MAC to the tag 0x87 of secondSequenceOf87 by using &lt;S_MAC&gt;.</p> <p>End If</p> <p>Encrypt all &lt;UPP_SEG&gt; with &lt;S_ENC&gt;, or &lt;PPK_ENC&gt; if paramReplaceSessionKeys is provided.</p> <p>Calculate and add a MAC to all tags 0x86 of sequenceOf86 by using &lt;S_MAC&gt;, or &lt;PPK_MAC&gt; (and &lt;PPK_INIT_MAC&gt; for the first tag) if paramReplaceSessionKeys is provided.</p>
--	---

<b>Method</b>	MTD_GENERATE_HASHED_CC
Description	Generate an Hashed Confirmation Code based on the Confirmation Code and the Transaction ID given in parameter.
Parameter(s)	<ul style="list-style-type: none"> <li>• paramConfirmationCode: The Confirmation Code (plain)</li> <li>• paramTransactionId: The Transaction ID (plain)</li> </ul>
Details	<p>Generate a SHA-256 of the paramConfirmationCode.</p> <p>Concatenate the obtained hash value with the paramTransactionId.</p> <p>Generate and return a SHA-256 of these two concatenated elements.</p>

<b>Method</b>	MTD_GET_BPP
Description	Generates or verifies the JSON formatted GetBoundProfilePackage request
Parameter(s)	<ul style="list-style-type: none"> <li>• paramTransactionId: random 16 byte identifier.</li> <li>• paramPrepareDownloadResponse structured as ASN.1 encoded as base 64.</li> </ul>
Details	<p>JSON body</p> <pre style="margin-left: 40px;">{     "transactionId" : paramTransactionId,     "prepareDownloadResponse" :     paramPrepareDownloadResponse }</pre>

<b>Method</b>	MTD_HANDLE_NOTIF
Description	Generates or verifies the JSON formatted HandleNotification request
Parameter(s)	paramPendingNotification: PendingNotification data object
Details	<p>JSON body</p> <pre style="margin-left: 40px;">{     "pendingNotification" : paramPendingNotification }</pre>

<b>Method</b>	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)	<ul style="list-style-type: none"> <li>paramServerAddress: Target Server address</li> <li>paramFunctionPath: Function path</li> <li>paramRequestMessage: JSON Request message</li> </ul>
Details	<p>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: &lt;L&gt;</p> <p>paramRequestMessage</p> <p>NOTE 1: If the request is sent by the LPA, 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.</p> <p>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). 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.</p>

<b>Method</b>	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)	<ul style="list-style-type: none"> <li>paramResponseMessage: JSON Response message</li> </ul>
Details	<p>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: &lt;L&gt;</p> <p>paramResponseMessage</p> <p>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). If the response is sent by a simulator, application/json shall be used.</p> <p>The HTTP response may contain additional header fields. These shall not be checked.</p>

<b>Method</b>	MTD_INITIATE_AUTHENTICATION
Description	Generates or verifies the JSON formatted Initiate Authentication request on ES9+ or ES11 as applicable.

Parameter(s)	<ul style="list-style-type: none"> <li>paramEUICCChallenge: random 16 byte challenge coded as base 64</li> <li>paramEUICCInfo1: eUICC information structured coded as base 64</li> <li>paramServerAddress: FQDN of the Server.</li> </ul>
Details	JSON body <pre>                 {                   "euiccChallenge" : paramEUICCChallenge,                   "euiccInfo1" : paramEUICCInfo1,                   "smdpAddress" : paramServerAddress                 }                 </pre>

<b>Method</b>	MTD_REGISTER_EVENT
Description	Send or checks the JSON formatted RegisterEvent request
Parameter(s)	<ul style="list-style-type: none"> <li>paramFunctionRequesterId: identification of the function requester.</li> <li>paramFunctionCallId: identification of the function call.</li> <li>paramEID: EID of the targeted eUICC</li> <li>paramRspServerAddress: Address of the Server sending the RegisterEvent formatted as FQDN</li> <li>paramEventId: unique Identification of the Event to be registered</li> <li>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</li> </ul>
Details	JSON requestHeader <pre>                 {                   "header" : {                     "functionRequesterIdentifier" :                     "paramFunctionRequesterId",                     "functionCallIdentifier" : "paramFunctionCallId"                   }                 }                 </pre> JSON body <pre>                 {                   "eid" : paramEID,                   "rspServerAddress" : paramRspServerAddress,                   "eventId" : paramEventId,                   "forwardingIndicator" : paramForwardingIndicator                 }                 </pre>

<b>Method</b>	MTD_REMOVE_NOTIF
Description	Constructs the command data for RemoveNotificationFromList
Parameter(s)	<ul style="list-style-type: none"> <li>paramSeqNumber: the sequence number to be removed</li> </ul>
Details	<pre>                 request NotificationSentRequest ::= {                   seqNumber paramSeqNumber                 }                 </pre>

<b>Method</b>	MTD_RETRIEVE_NOTIF_SEQ_NUM
Description	Constructs the command data for RetrieveNotificationsList filtered by sequence number



Parameter(s)	<ul style="list-style-type: none"> <li>paramSeqNumber: the sequence number to be retrieved</li> </ul>
Details	<pre>request RetrieveNotificationsListRequest ::= {     searchCriteria seqNumber paramSeqNumber }</pre>

<b>Method</b>	MTD_SELECT
Description	Generates the SELECT command as defined in GlobalPlatform Card Specification [6].
Parameter(s)	<ul style="list-style-type: none"> <li>paramAID: the AID to select</li> </ul>
Details	<ul style="list-style-type: none"> <li>- CLA = 0x or 4x (x = &lt;CHANNEL_NUMBER&gt;)</li> <li>- INS = A4</li> <li>- P1 = 04</li> <li>- P2 = 00</li> <li>- LC = &lt;L&gt;</li> <li>- paramAID</li> <li>- LE = 00</li> </ul>

<b>Method</b>	MTD_SEND_SMS_PP
Description	Generate and send an envelope SMS-PP download to the MNO-SD
Parameter(s)	<ul style="list-style-type: none"> <li>paramApdusList: the list of APDUs (plain) to send</li> </ul>
Details	<p>Generate and send the following envelope:</p> <pre>80 C2 00 00 &lt;L&gt;   D1 &lt;L&gt;     02 02 83 81 -- Device identity Tag     06 07 91 33 86 09 40 00 F0 -- Address Tag (TON/NPI/..)     0B &lt;L&gt; -- 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       &lt;L&gt; -- TP-User-Data-Length         02 -- User-Data-Header-Length         70 -- IEIa         00 -- IEIDLa       &lt;L&gt; -- Command Packet Length (2 bytes)         &lt;L&gt; -- Command Header Length (1 byte)           12 21 -- SPI           00 -- KIC           15 -- KID (SCP80 Keypset version 0x01 in Triple DES)            B2 01 00 -- MNO-SD TAR           &lt;MNO_SCP80_COUNTER&gt;           00 -- Padding Counter           &lt;CC&gt; -- Cryptographic checksum           &lt;C_APDUS_SCRIPT&gt; -- Command APDUs script</pre>

	<p>&lt;C_APDUS_SCRIPT&gt; SHALL contain the paramApdusList (i.e. each APDU is named &lt;APDU1&gt;; &lt;APDU2&gt;; ...; &lt;APDU<sub>n</sub>&gt; here after) formatted as an expanded structure with definite length as defined in ETSI TS 102 226 [14]:</p> <pre>AA &lt;L&gt;   22 &lt;L&gt; &lt;APDU1&gt;   22 &lt;L&gt; &lt;APDU2&gt;   ...   22 &lt;L&gt; &lt;APDU<sub>n</sub>&gt;</pre> <p>The Cryptographic checksum &lt;CC&gt; 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].</p> <p>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.</p>
--	--

<b>Method</b>	MTD_STORE_DATA
Description	Generates the STORE DATA command (Case 4) as defined in GlobalPlatform Card Specification [6].
Parameter(s)	<ul style="list-style-type: none"> <li>paramCommandData: the command data</li> </ul>
Details	<ul style="list-style-type: none"> <li>- CLA = 8x or Cx (x = &lt;CHANNEL_NUMBER&gt;)</li> <li>- INS = E2</li> <li>- P1 = 91</li> <li>- P2 = 00</li> <li>- LC = &lt;L&gt;</li> <li>- paramCommandData</li> <li>- LE = 00</li> </ul>

<b>Method</b>	MTD_STORE_DATA_Case3
Description	Generates the STORE DATA command (Case3) as defined in GlobalPlatform Card Specification [6].
Parameter(s)	<ul style="list-style-type: none"> <li>paramCommandData: the command data</li> </ul>
Details	<ul style="list-style-type: none"> <li>- CLA = 8x or Cx (x = &lt;CHANNEL_NUMBER&gt;)</li> <li>- INS = E2</li> <li>- P1 = 90</li> <li>- P2 = 00</li> <li>- LC = &lt;L&gt;</li> <li>- paramCommandData</li> </ul>

<b>Method</b>	MTD_STORE_DATA_SCRIPT
Description	Generate (multiple) STORE DATA command(s) by breaking the data into smaller components (if needed) for transmission.

Parameter(s)	<ul style="list-style-type: none"> <li>paramTLVDataToTransmit: TLVs array or single TLV to transfer to the eUICC</li> <li>paramCase4Command (optional parameter, default value = TRUE): TRUE if the APDU is a Case 4 command, FALSE if the APDU is a Case 3 command</li> </ul>
Details	<p>For each element of paramTLVDataToTransmit</p> <p>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 &lt;DATA_SUB_PART&gt; here after.</p> <p>If the element is up to 255 bytes, &lt;DATA_SUB_PART&gt; contains the value of the element.</p> <p>The bit b1 of P1 in the STORE DATA commands is named &lt;B1_P1&gt; here after and is defined as below:</p> <pre>         If paramCase4Command = TRUE Then             &lt;B1_P1&gt; = 1         Else             &lt;B1_P1&gt; = 0         End If     </pre> <p>Set &lt;STORE_DATA_BLOCK_NUM&gt; to 0</p> <p>For each &lt;DATA_SUB_PART&gt;</p> <p>If &lt;DATA_SUB_PART&gt; is an intermediate part, generate the following STORE DATA:</p> <pre>         - CLA = 8x or Cx (x = &lt;CHANNEL_NUMBER&gt;)         - INS = E2         - P1 = 1x (x = &lt;B1_P1&gt;)         - P2 = &lt;STORE_DATA_BLOCK_NUM&gt;         - LC = &lt;L&gt;         - &lt;DATA_SUB_PART&gt;         - LE = 00 -- present only if paramCase4Command = TRUE     </pre> <p>If &lt;DATA_SUB_PART&gt; is the last part, generate the following STORE DATA:</p> <pre>         - CLA = 8x or Cx (x = &lt;CHANNEL_NUMBER&gt;)         - INS = E2         - P1 = 9x (x = &lt;B1_P1&gt;)         - P2 = &lt;STORE_DATA_BLOCK_NUM&gt;         - LC = &lt;L&gt;         - &lt;DATA_SUB_PART&gt;         - LE = 00 -- present only if paramCase4Command = TRUE     </pre> <p>Increase the &lt;STORE_DATA_BLOCK_NUM&gt; by 1</p> <p>End</p> <p>End</p>

<b>Method</b>	MTD_TEST_ES8+_GET_BPP_PPK
Description	Tests the received boundProfilePackage element according to #R_GET_BPP_RESP_OP1_PPK
Parameter(s)	<ul style="list-style-type: none"> <li>paramResponse the response to GetBoundProfilePackage</li> <li>paramS_MAC the 128 bit SCP03t MACing Session key</li> </ul>

	<ul style="list-style-type: none"> <li>paramS_ENC the 128 bit SCP03t Encryption Session key</li> <li>paramPPK_MAC the 128 bit Profile Protection MACing Key</li> <li>paramPPK_ENC the 128 bit Profile Protection Encryption Key</li> <li>paramMetaData the ASN.1 StoreMetadataRequest element associated to a RSP profile</li> </ul>
Details	<p>Parse paramResponse into #R_GET_BPP_RESP_OP1_PPK and perform the following tests:</p> <ul style="list-style-type: none"> <li>Verify that each element in firstSequenceOf87, sequenceOf88, secondSequenceOf87 and sequenceOf86 has a total length (including tag and length fields) of 1020 or less</li> <li>Verify the integrity of each element in firstSequenceOf87, sequenceOf88 and secondSequenceOf87 using paramS_MAC</li> <li>Verify that &lt;TRANSACTION_ID_ISC&gt; in #INIT_SC_PROF1 matches &lt;S_TRANSACTION_ID&gt;</li> <li>Verify the validity of smdpSign &lt;SM_DP+_SIGN&gt; in #INIT_SC_PROF1 using #PK_SM_DPpb_ECDSA</li> <li>Retrieve #CONF_ISDP_PROF1_SMDP from &lt;CONF_ISDP_PROF1_ENC&gt; using paramS_ENC and validate the content of #CONF_ISDP_PROF1_SMDP</li> <li>Construct the complete metadata element from the &lt;SMDP_METADATA_SEG_MAC&gt; segment(s) and verify that it matches paramMetaData</li> <li>Retrieve #REPLACE_S_KEYS_REQ from &lt;REPLACE_S_KEYS_REQ_ENC&gt; using paramS_ENC and validate the content of #REPLACE_S_KEYS_REQ</li> <li>Verify that the lengths of paramPPK_ENC and paramPPK_MAC in #REPLACE_S_KEYS_REQ are each 16 bytes</li> <li>Verify the integrity of each &lt;PPP_OP_PROF1_SEG_PPK&gt; element using paramPPK_MAC</li> <li>Retrieve the &lt;UPP_OP_PROF1_SEG&gt; segment(s) from the &lt;PPP_OP_PROF1_SEG_PPK&gt; segment(s) using paramPPK_ENC, construct the complete Profile from the &lt;UPP_OP_PROF1_SEG&gt; segment(s), then verify that the complete Profile matches #UPP_OP_PROF1</li> </ul>

<b>Method</b>	MTD_TEST_ES8+_GET_BPP_SK
Description	Tests the received boundProfilePackage element according to #R_GET_BPP_RESP_OP1_SK
Parameter(s)	<ul style="list-style-type: none"> <li>paramResponse the response to GetBoundProfilePackage</li> <li>paramS_MAC the 128 bit SCP03t MACing Session key</li> <li>paramS_ENC the 128 bit SCP03t Encryption Session key</li> <li>paramMetaData the ASN.1 StoreMetadataRequest element associated to a RSP profile</li> </ul>
Details	<p>Parse paramResponse into #R_GET_BPP_RESP_OP1_SK and perform the following tests:</p> <ul style="list-style-type: none"> <li>Verify that each element in firstSequenceOf87, sequenceOf88 and sequenceOf86 has a total length (including tag and length fields) of 1020 or less</li> <li>Verify the integrity of each element in firstSequenceOf87, sequenceOf88 and sequenceOf86 using paramSMAC</li> <li>Verify that &lt;TRANSACTION_ID_ISC&gt; in #INIT_SC_PROF1 matches &lt;S_TRANSACTION_ID&gt;</li> <li>Verify the validity of smdpSign &lt;SM_DP+_SIGN&gt; in #INIT_SC_PROF1 using #PK_SM_DPpb_ECDSA</li> <li>Retrieve #CONF_ISDP_PROF1_SMDP from &lt;CONF_ISDP_PROF1_ENC&gt; using paramS_ENC and validate the content of #CONF_ISDP_PROF1_SMDP</li> </ul>

	<ul style="list-style-type: none"> <li>Construct the complete metadata element from the &lt;SMDP_METADATA_SEG_MAC&gt; segment(s) and verify that it matches paramMetaData</li> <li>Retrieve the &lt;UPP_OP_PROF1_SEG&gt; segment(s) from the &lt;PPP_OP_PROF1_SEG_SK&gt; segment(s) using paramS_ENC, then construct the complete Profile from the &lt;UPP_OP_PROF1_SEG&gt; segment(s), then verify that the complete Profile matches #UPP_OP_PROF1</li> </ul>
--	--

<b>Method</b>	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)	<ul style="list-style-type: none"> <li>paramClientKeyExchange: ClientKeyExchange message</li> </ul>
Details	Sends the session key information in TLS ClientKeyExchange message, ChangeCipherSpec and Finished message.

<b>Method</b>	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)	<ul style="list-style-type: none"> <li>paramTLSversion: TLS protocol version</li> <li>paramAlgs: cipher suite types supported</li> <li>paramSessionID: Session ID</li> <li>paramExts: Extensions data for "supported_signature_algorithms", "trusted_ca_keys" or other (optional)</li> </ul>
Details	<p>Sends or receives a TLS ClientHello message according to the parameters defined above.</p> <p>In addition the following parameters will be set:</p> <ul style="list-style-type: none"> <li>The list of compression algorithms supported by the client is not explicitly defined, but by default it will be set to NULL.</li> <li>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</li> </ul> <p>NOTE: The Supported Elliptic Curves Extension and the Supported Point Formats Extension extensions MAY be sent by the Client.</p>

<b>Method</b>	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)	<ul style="list-style-type: none"> <li>paramClientCertificate: TLS Client certificate for authentication used in the Client Certificate Message</li> <li>paramClientKeyExchange: The Client TLS Ephemeral Key used in the ClientKeyExchange message</li> </ul>
Details	<p>Sends the TLS Client Certificate, ClientKeyExchange, Certificate Verify, ChangeCipherSpec and Finished message in this order according to the parameters defined above.</p> <p>NOTE 1: The CertificateVerify Message is not explicitly defined in this method but the CLIENT or test tool implementation SHALL be responsible for</p>

	<p>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.</p> <p>NOTE 2: ChangeCipherSpec messages, alerts, and any other record type are not handshake messages and are not included in the signature computations.</p>
--	--

<b>Method</b>	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)	<ul style="list-style-type: none"> <li>• paramTLSVersion: TLS protocol version used in the Server Hello Message</li> <li>• paramAlgs: cipher suite selected used in the Server Hello Message</li> <li>• paramSessionID: Session ID used in the Server Hello Message</li> <li>• paramServerCertificate: TLS Server certificate for authentication used in the Server Certificate Message</li> <li>• paramServerTLSEphemeralKey: TLS Server ephemeral key used in the Server Key Exchange Message</li> <li>• paramClientCertificateType: type of certificate requested used in the Client Certificate Request Message</li> <li>• paramSignatureAndHashAlgorithm: Signature and Hash Algorithm to be verified used in the Client Certificate Request Message</li> <li>• paramDistinguishedName: DN of the CI that signed and issued the certificate used in the Client Certificate Request Message</li> </ul>
Details	<p>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:</p> <ul style="list-style-type: none"> <li>• ServerHello                         <ul style="list-style-type: none"> <li>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.</li> </ul> </li> <li>• ServerKeyExchange                         <ul style="list-style-type: none"> <li>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.</li> </ul> </li> </ul> <p>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.</p>

<b>Method</b>	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)	<ul style="list-style-type: none"> <li>• paramChangeCipherSpec: ChangeCipherSpec message</li> <li>• paramFinish: Finished message</li> </ul>
Details	Sends a ChangeCipherSpec and Finished message in this order according to the parameters defined above.

<b>Method</b>	MTD_TLS_SERVER_HELLO_ETC
<b>Description</b>	Send or Receives to the Client Hello in the Transport Layer Security (TLS) handshake in Server authentication mode on ES9+, or ES11.
<b>Parameter(s)</b>	<ul style="list-style-type: none"> <li>paramTLSversion: TLS protocol version</li> <li>paramAlgs: cipher suite selected</li> <li>paramSessionID: Session ID</li> <li>paramCertificate: TLS server certificate for authentication</li> <li>paramServerTLSEphemeralKey: TLS Server ephemeral key.</li> </ul>
<b>Details</b>	<p>Sends or Receives a TLS ServerHello, Server Certificate, ServerKeyExchange and ServerHelloDone message in this order according to the parameters defined above.</p> <p>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.</p> <p>NOTE 2: If no parameter mentioned paramServerTLSEphemeralKey, the value SHALL be set as defined in [24] for ServerKeyExchange. No verification required.</p>

## C.2 Procedures

<b>Procedure</b>	PROC_ES11_AUTH_CLIENT			
<b>Description</b>	Authenticate Server procedure and Event Retrieval from SM-SD.			
For LPA <sub>d</sub> testing, execute the following steps:				
Step	Direction	Sequence / Description	Expected result	REQ
1	LPA <sub>d</sub> → S <sub>SM-DS</sub>	Send ES11.AuthenticateClient method	MTD_HTTP_REQ(#TEST_ROT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO))	
2	S <sub>SM-DS</sub> → LPA <sub>d</sub>	MTD_HTTP_RESP (#AUTH_CLIENT_DS_OK)	No error	

<b>Procedure</b>	PROC_ES11_VERIFY_EVENT_RETRIEVAL			
<b>Description</b>	Performs Common Mutual Authentication on ES11 from S <sub>LPA<sub>d</sub></sub> to SM-DS under test and verifies that the pending Event #EVENT_ENTRY_1 is retrieved.			
Step	Direction	Sequence / Description	Expected result	REQ
1	S <sub>LPA<sub>d</sub></sub> → SM-DS	PROC_TLS_INITIALIZATION_SERVER_AUTH		
2	S <sub>LPA<sub>d</sub></sub> → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)	

3	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHIN G_ID_EVENT_ID))	MTD_HTTP_RESP( #R_AUTH_CLIENT_DS_EVE NT_ENTRY_1_OK)	
---	----------------	---	---	--

<b>Procedure</b>	PROC_ES11_VERIFY_EVENT_RETRIEVAL_ERROR
<b>Description</b>	Performs Common Mutual Authentication on ES11 from S_LPAd to SM-DS under test 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, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION ( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS))	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	
3	S_LPAd → SM-DS	MTD_HTTP_REQ( #IUT_SM_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHIN G_ID_EVENT_ID))	MTD_HTTP_RESP( #R_ERROR_8_9_5_3_9)	

<b>Procedure</b>	PROC_ES11_INIT_AUTH
<b>Description</b>	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_ADDRES S, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTIC ATION( <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRES S))	



2	S_SM-DS → LPAd	MTD_HTTP_RESP( #INITIATE_AUTH_DS_OK)	No error	
---	----------------	---	----------	--

<b>Procedure</b>		PROC_EUICC_INITIALIZATION_SEQUENCE		
<b>Description</b>		Initialize communication between the S_Device and the eUICC.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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	

<b>Procedure</b>		PROC_EUICC_INITIALIZATION_SEQUENCE_eUICCProfileStateChang ed		
<b>Description</b>		Initialize communication between the S_Device and the eUICC.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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_eUICCProfil eStateChanged]	Toolkit initialization THEN SW=0x9000	

<b>Procedure</b>		PROC_OPEN_LOGICAL_CHANNEL_AND_SELECT_ISDR		
<b>Description</b>		The LPAd opens a logical channel and selects the ISD-R.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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	

<b>Procedure</b>	PROC_ES9+_AUTH_CLIENT
------------------	-----------------------

<b>Description</b>		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).		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
For LPAAd testing, execute the following steps:				
1	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK)	No error	
For SM-DP+ testing, execute the following steps:				
1	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_ACT_CO DE_UC_OK))	MTD_HTTP_RESP(#R_AUT H_CLIENT_OK)	

<b>Procedure</b>		PROC_ES9+_AUTH_CLIENT_CC		
<b>Description</b>		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).		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
For LPAAd testing, execute the following steps:				
1	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK_CC)	No error	
For SM-DP+ testing, execute the following steps:				
1	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_ACT_CO DE_UC_OK))	MTD_HTTP_RESP(#R_AUT H_CLIENT_OK_CC)	

		#AUTH_SERVER_RESP_ACT_CODE_UC_OK))		
--	--	------------------------------------	--	--

Procedure		PROC_ES9+_GET_BPP		
Description		Get BPP procedure without Confirmation Code.		
Step	Direction	Sequence / Description	Expected result	REQ
For LPAAd testing, execute the following steps:				
1	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error	
For SM-DP+ testing, execute the following steps:				
1	S_LPAAd → 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 LPAAd testing, execute the following steps:				
1	LPAAd → 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+ → LPAAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error	
For SM-DP+ testing, execute the following steps:				
1	S_LPAAd → 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)	

		#PREP_DOWNLOAD_RESP_CC))		
--	--	--------------------------	--	--

		<b>Procedure</b>	PROC_ES9+_HANDLE_NOTIF		
		<b>Description</b>	Handle Notification procedure.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>	
For LPAAd testing, execute the following steps:					
1	LPAAd → 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+ → LPAAd	#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).					

		<b>Procedure</b>	PROC_ES9+_AUTH_CLIENT_FAIL_DEF_DP_USE_CASE_INVALID_MATCHING_ID		
		<b>Description</b>	AuthenticateClient fails due to an Invalid Matching ID.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>	
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+				
2	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATIO N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP( #R_INITI ATE_AUTH_OK)		
3	S_LPAAd → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_ACT_CO DE_UC_OK))	MTD_HTTP_RESP( #R_ERROR_8_2_6_3_8)		

		<b>Procedure</b>	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CANCEL_SESSION_SK		
		<b>Description</b>	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.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>	
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_S K)	
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.	

	<b>Procedure</b>	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CANCEL_SESSION_PPK		
	<b>Description</b>	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.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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_P PK)	
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.	

<b>Procedure</b>	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CC_CANCEL_SESSION_PPK
<b>Description</b>	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_C C)	
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_U C_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_O P1_PPK)	

Procedure		PROC_ES9+_PROF_DOWNLOAD_ACT_CODE_USE_CASE_CAN CEL_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_IN ITIATE_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_CO DE_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_HTTP_RESP( #R_GET_BPP_RESP_OP 1_PPK)	

		MTD_GET_BPP( <S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))		
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.

<b>Procedure</b>	PROC_ES9+_PROF_DOWNLOAD_SM_DS_USE_CASE_CANCEL_SESSION
<b>Description</b>	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_I NITIATE_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_O P1_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.

<b>Procedure</b>	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC
<b>Description</b>	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_AUTHE NTICATE_CLIENT( <S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_U C_OK))	MTD_HTTP_RESP( #R_AUTH_CLIENT_OK)	

<b>Procedure</b>	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_CC
<b>Description</b>	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_U C_OK))	MTD_HTTP_RESP( #R_AUTH_CLIENT_OK_ CC)	

<b>Procedure</b>	PROC_ES9+_INIT_AUTH
<b>Description</b>	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_AUTHENTI CATION(  <EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1))	

2	S_SM-DP+ → LPA <sub>d</sub>	MTD_HTTP_RESP( #INITIATE_AUTH_OK)	No error	
For SM-DP+ testing, execute the following steps:				
1	S_LPA <sub>d</sub> → 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)	

<b>Procedure</b>	PROC_ES9+_VERIFY_CMA_PD_DEF_SMDP_ADDRESS_NO_CC_FAIL
<b>Description</b>	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_LPA <sub>d</sub> → SM-DP+	MTD_HTTP_REQ( #IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATI ON( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INI TIATE_AUTH_OK)	
3	S_LPA <sub>d</sub> → 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)	

<b>Procedure</b>	PROC_VERIFY_SESSION_IS_CANCELLED			
<b>Description</b>	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_LPA <sub>d</sub> → eUICC	MTD_STORE_DATA_SCRIPT( #PREP_DOWNLOAD_NO_CC)	#R_PREP_DOWN_NO_SES SION	

			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)	

	<b>Procedure</b>	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CC_CANCEL_SESSION_SK		
	<b>Description</b>	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.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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_AUTHENTICATIO N( #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.	

<b>Procedure</b>		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_CC_RETRY		
<b>Description</b>		Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case without a confirmation code.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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_AUTHENTICATIO N( #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)	

<b>Procedure</b>		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_INVALID_CC		
<b>Description</b>		Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case with an invalid confirmation code provided in the GetBoundProfilePackage.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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)	

<b>Procedure</b>		PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC_RETRY		
<b>Description</b>		Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case without a confirmation code.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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_AUTHENTICATIO	MTD_HTTP_RESP( #R_INITIATE_AUTH_OK)	

		N( #S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))		
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)	

<b>Procedure</b>	PROC_TLS_INITIALIZATION_SERVER_AUTH
<b>Description</b>	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(#TL S_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SERVER_TLS)	MTD_TLS_CLIENT_KEY_EXC H_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_E TC( #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>)	

<b>Procedure</b>	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_U C_NO_CC
<b>Description</b>	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_AUTHENTICAT ION( #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_P PK)	

	<b>Procedure</b>	PROC_ES9+_VERIFY_PROFILE_DOWNLOAD_DEF_SMDP_ADD RESS_UC		
	<b>Description</b>	Verifies that Common Mutual Authentication occurs successfully and that the Bound Profile Package is generated and successfully delivered to the LPAd.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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_AUTHENTICAT ION( #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_P PK)	

		#PREP_DOWNLOAD_RESP))	
--	--	-----------------------	--

<b>Procedure</b>		PROC_TLS_INITIALIZATION_MUTUAL_AUTH		
<b>Description</b>		Establishes the Transport Layer Security (TLS) v1.2 connection between the Client and Server using Mutual authentication mode on ES12 or ES15. For Client and Server testing the Server MAY be the SM-DS or the SM-DP+.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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>)	

<b>Procedure</b>		PROC_TLS_INITIALIZATION_MUTUAL_AUTH_INV_OID		
<b>Description</b>		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+.		
<b>Step</b>	<b>Direction</b>	<b>Sequence / Description</b>	<b>Expected result</b>	<b>REQ</b>
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,	MTD_TLS_SERVER_END( #CHANGE_CIPHER_SPEC, <SERVER_FINISHED>)	

		<CLIENT_TLS_EPHEM_KEY> )		
--	--	-----------------------------	--	--



## 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	<pre> 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 }                     </pre>
CONF_ISDP_PROF1	<pre> req ConfigureISDPRequest ::= {     dpProprietaryData {         dpOid #S_SM_DP+_OID     } }                     </pre>
CONF_ISDP_PROF1_SMDP	<pre> req ConfigureISDPRequest ::= {     dpProprietaryData {         dpOid #IUT_SM_DP_OID     } -- optional }                     </pre>
CONF_ISDP_SIZE_EXCEEDED	<pre> 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                     </pre>

	<pre> the   -- SM-DP+ MAY follow }  -- the following structure is used to test the -- DpProprietaryData size: DpProprietaryData ::= SEQUENCE {   dpOid OBJECT IDENTIFIER,   additionalSmdpData OCTET STRING OPTIONAL }         </pre>
<p>FULL_METADATA</p>	<pre> 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} }         </pre>
<p>INIT_SC_INVALID_CRT</p>	<pre> req InitialiseSecureChannelRequest ::= {   remoteOpId #REMOTE_OP_ID_INSTALL,   transactionId &lt;S_TRANSACTION_ID&gt;,   controlRefTemplate {     keyType #INVALID_KEY_TYPE,     keyLen #KEY_LENGTH,     hostId #HOST_ID   },   smdpOtpk &lt;OTPK_S_SM_DP+_ECKA&gt;,   smdpSign &lt;S_SM_DP+_SIGN&gt; }         </pre>
<p>INIT_SC_INVALID_OP_ID</p>	<pre> req InitialiseSecureChannelRequest ::= {   remoteOpId #INVALID_REMOTE_OP_ID,   transactionId &lt;S_TRANSACTION_ID&gt;,   controlRefTemplate {     keyType #KEY_TYPE,     keyLen #KEY_LENGTH,     hostId #HOST_ID   },   smdpOtpk &lt;OTPK_S_SM_DP+_ECKA&gt;,   smdpSign &lt;S_SM_DP+_SIGN&gt; }         </pre>

	} }
INIT_SC_INVALID_SIGN	<pre>req InitialiseSecureChannelRequest ::= {   remoteOpId #REMOTE_OP_ID_INSTALL,   transactionId &lt;S_TRANSACTION_ID&gt;,   controlRefTemplate {     keyType #KEY_TYPE,     keyLen #KEY_LENGTH,     hostId #HOST_ID   },   smdpOtpk &lt;OTPK_S_SM_DP+_ECKA&gt;,   smdpSign &lt;S_SM_DP+_SIGN&gt; } <i>The &lt;S_SM_DP+_SIGN&gt; SHALL NOT be computed using the #SK_S_SM_DPpb_ECDSA but SHALL have the same length as for a valid signature</i></pre>
INIT_SC_INVALID_TRANS_ID	<pre>req InitialiseSecureChannelRequest ::= {   remoteOpId #REMOTE_OP_ID_INSTALL,   transactionId   &lt;INVALID_TRANSACTION_ID&gt;,   controlRefTemplate {     keyType #KEY_TYPE,     keyLen #KEY_LENGTH,     hostId #HOST_ID   },   smdpOtpk &lt;OTPK_S_SM_DP+_ECKA&gt;,   smdpSign &lt;S_SM_DP+_SIGN&gt; }</pre>
INIT_SC_PROF1	<pre>req InitialiseSecureChannelRequest ::= {   remoteOpId #REMOTE_OP_ID_INSTALL,   transactionId &lt;TRANSACTION_ID_ISC&gt;,   controlRefTemplate {     keyType #KEY_TYPE,     keyLen #KEY_LENGTH,     hostId #IUT_SM_DP_HOST_ID   },   smdpOtpk &lt;OTPK_SM_DP+_ECKA&gt;,   smdpSign &lt;SM_DP+_SIGN&gt; }</pre>
METADATA_ICCID_MISMATCH	<pre>metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF2,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1 }</pre>
METADATA_MCCMNC_MISMATCH	<pre>metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   profileOwner {     mccMnc #MCC_MNC2   },   profilePolicyRules {ppr2}</pre>

	}
METADATA_NO_CLASS	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   notificationConfigurationInfo {     { profileManagementOperation {       notificationInstall,       notificationEnable,       notificationDisable,       notificationDelete     },       notificationAddress #TEST_DP_ADDRESS1     }   } } </pre>
METADATA_OP_PROF1	<pre> 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   } } </pre>
METADATA_OP_PROF1_EN	<pre> 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   } } </pre>

	<pre> } }         </pre>
<p>METADATA_OP_PROF1_INST_DIFF</p>	<pre> 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   } }         </pre>
<p>METADATA_OP_PROF1_MEMRES1</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   iconType png,   icon #ICON_OP_PROF1,   profileClass operational,   notificationConfigurationInfo {     { profileManagementOperation {       notificationDelete     },       notificationAddress #TEST_DP_ADDRESS1     }   },   profileOwner {     mccMnc #MCC_MNC1   },   profilePolicyRules { ppr2 } }         </pre>
<p>METADATA_OP_PROF5_MEMRES2</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF5,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF5,   iconType png,   icon #ICON_OP_PROF5,   profileClass operational,   notificationConfigurationInfo {     { profileManagementOperation {       notificationDisable,       notificationDelete     },       notificationAddress #TEST_DP_ADDRESS1     }   } }         </pre>

	<pre> }, profileOwner {     mccMnc #MCC_MNC1 }, profilePolicyRules { ppr1,ppr2 } }         </pre>
<p>METADATA_OP_PROF2</p>	<pre> 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     } }         </pre>
<p>METADATA_OP_PROF2_NO_INSTALL</p>	<pre> 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     } }         </pre>
<p>METADATA_OP_PROF1_NO_INSTALL</p>	<pre> metadataReq StoreMetadataRequest ::= {     iccid #ICCID_OP_PROF1,     serviceProviderName #SP_NAME1,     profileName #NAME_OP_PROF1,     iconType png,     icon #ICON_OP_PROF1, }         </pre>

	<pre> profileClass operational, notificationConfigurationInfo {     {         profileManagementOperation {             notificationEnable,             notificationDisable,             notificationDelete         },         notificationAddress #TEST_DP_ADDRESS1     } }, profileOwner {     mccMnc #MCC_MNC1 } }                 </pre>
<p>METADATA_OP_PROF2_TEST_DP_ADDRESS1</p>	<pre> 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 } }                 </pre>
<p>METADATA_OP_PROF3</p>	<pre> 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 } }                 </pre>
<p>METADATA_OP_PROF4</p>	<pre> metadataReq StoreMetadataRequest ::= {     iccid #ICCID_OP_PROF4,     serviceProviderName #SP_NAME4,     profileName #NAME_OP_PROF4,     iconType png,     icon #ICON_OP_PROF4,     profileClass operational,     notificationConfigurationInfo {                 </pre>

	<pre>         { profileManagementOperation {             notificationInstall,             notificationEnable,             notificationDisable,             notificationDelete         },         notificationAddress #TEST_DP_ADDRESS4     } }, profileOwner {     mccMnc #MCC_MNC4 }, profilePolicyRules {     ppr1 }     }     </pre>
<p>METADATA_OP_PROF5</p>	<pre> 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     } }     </pre>
<p>METADATA_OP_PROF6</p>	<pre> 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     } },     </pre>



	<pre> profileOwner {   mccMnc #MCC_MNC2 } </pre>
<p>METADATA_OP_PROF7</p>	<pre> 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_MNC8 }, profilePolicyRules {   ppr2 } } </pre>
<p>METADATA_OP_PROF8</p>	<pre> 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_MNC8 }, profilePolicyRules {   ppr2 } } </pre>

<p>METADATA_OP_PROF9</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF9,   serviceProviderName #SP_NAME9,   profileName #NAME_OP_PROF9,   profileOwner {     mccMnc #MCC_MNC9,     gid1 #GID1,     gid2 #GID2   },   profilePolicyRules {     ppr2   } } </pre>
<p>METADATA_OP1_GID1GID2_PRESENT</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   profileOwner {     mccMnc #MCC_MNC1,     gid1 #GID1,     gid2 #GID2   },   profilePolicyRules {ppr2} } </pre>
<p>METADATA_OP9_GID1GID2_MISSING</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF9,   serviceProviderName #SP_NAME9,   profileName #NAME_OP_PROF9,   profileOwner {     mccMnc #MCC_MNC9   } } </pre>
<p>METADATA_PPR_NO_OWNER</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   profilePolicyRules {ppr2} } </pre>
<p>METADATA_WILDCARD</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   profileOwner {     mccMnc #MCC_MNC_WILDCARD   },   profilePolicyRules {ppr2} } </pre>
<p>METADATA_WITH_JPG</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   iconType jpg,   icon #ICON_JPG } </pre>

	}
METADATA_WITH_NOTIFS	<pre> 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     }   } } </pre>

<p>METADATA_WITH_PPR1_PPR2</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   profileOwner {     mccMnc #MCC_MNC1   },   profilePolicyRules {ppr1,ppr2} }                     </pre>
<p>METADATA_WITH_PPR2</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   profileOwner {     mccMnc #MCC_MNC1   },   profilePolicyRules {ppr2} }                     </pre>
<p>METADATA_WITH_PPRS_AND_ICON</p>	<pre> 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} }                     </pre>
<p>METADATA_WITHOUT_ICON</p>	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   iconType jpg }                     </pre>
<p>REPLACE_S_KEYS_REQ</p>	<pre> req ReplaceSessionKeysRequest ::= {   initialMacChainingValue &lt;PPK_INIT_MAC&gt;,   ppkEnc &lt;PPK_ENC&gt;,   ppkCmac &lt;PPK_MAC&gt; }                     </pre>
<p>REPLACE_S_KEYS_REQ_INV_SIZE</p>	<pre> req ReplaceSessionKeysRequest ::= {   initialMacChainingValue #PPK_INIT_MAC_INV_SIZE,   ppkEnc #PPK_ENC_INV_SIZE,   ppkCmac #PPK_MAC_INV_SIZE }                     </pre>
<p>S_INIT_SC_PROF1</p>	<pre> req InitialiseSecureChannelRequest ::= {   remoteOpId #REMOTE_OP_ID_INSTALL,   transactionId &lt;S_TRANSACTION_ID&gt;,   controlRefTemplate {     keyType #KEY_TYPE,                     </pre>

	<pre> keyLen #KEY_LENGTH, hostId #HOST_ID }, smdpOtpk &lt;OTPK_S_SM_DP+_ECKA&gt;, smdpSign &lt;S_SM_DP+_SIGN&gt; } </pre>
SMDP_METADATA_ABS	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1 } </pre>
SMDP_METADATA_ALL	<pre> 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 } } </pre>
SMDP_METADATA_NON_ASCII	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME_NON_ASCII,   profileName #NAME_OP_PROF1_NON_ASCII } </pre>
SMDP_METADATA_NOTIF_MULTI	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   notificationConfigurationInfo {     {       profileManagementOperation {         notificationInstall,         notificationEnable,         notificationDisable,         notificationDelete       },     },     notificationAddress </pre>

	<pre>#IUT_SM_DP_ADDRESS }, {     profileManagementOperation {         notificationInstall,         notificationEnable,         notificationDisable,         notificationDelete     },     notificationAddress #TEST_DP_ADDRESS1 } } }</pre>
<p>SMDP_METADATA_OP_PROF1_EN</p>	<pre>metadataReq StoreMetadataRequest ::= {     iccid #ICCID_OP_PROF1,     serviceProviderName #SP_NAME1,     profileName #NAME_OP_PROF1,     profileClass operational,     notificationConfigurationInfo {         { profileManagementOperation {             notificationEnable         },         notificationAddress #IUT_SM_DP_ADDRESS     } } }</pre>
<p>SMDP_METADATA_OP_PROF1_PPR2</p>	<pre>metadataReq StoreMetadataRequest ::= {     iccid #ICCID_OP_PROF1,     serviceProviderName #SP_NAME1,     profileName #NAME_OP_PROF1,     profileClass operational,     profileOwner {         mccMnc #MCC_MNC1     },     profilePolicyRules { ppr2 } }</pre>
<p>SMDP_METADATA_PN_LONG</p>	<pre>metadataReq StoreMetadataRequest ::= {     iccid #ICCID_OP_PROF1,     serviceProviderName #SP_NAME1,     profileName #NAME_OP_PROF_LONG }</pre>
<p>SMDP_METADATA_SPN_LONG</p>	<pre>metadataReq StoreMetadataRequest ::= {     iccid #ICCID_OP_PROF1,     serviceProviderName #SP_NAME_LONG,     profileName #NAME_OP_PROF1 }</pre>

## D.2 ES9+ Requests And Responses

### D.2.1 ES9+ Requests

Name	Content
AUTH_SERVER_RESP_ACT_CODE_UC_OK	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_ACT_CODE     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA } </pre>
AUTH_SERVER_RESP_ACT_CODE_2_UC_OK	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_ACT_CODE_2     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA } </pre>
AUTH_SERVER_RESP_DEF_DP_UC_8_1_1_3_8	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     }, </pre>

	<pre> euiccSignature1 &lt;EUICC_SIGNATURE1&gt;, euiccCertificate #CERT_EUICC_ECDSA_EID2, eumCertificate #CERT_EUM_ECDSA }                 </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_4_8</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2         #S_EUICC_INFO2_INSUF_MEM_ERROR,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }                 </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_BC_cA</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_EX_BC_cA }                 </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_BC_PLC</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate }                 </pre>



	<pre>#CERT_EUM_ECDSA_INVALID_EX_BC_PLC }</pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_CP</p>	<pre>resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_EX_CP }</pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_KU</p>	<pre>resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_EX_KU }</pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_SIG</p>	<pre>resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_SIG }</pre>

	}
AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_3	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_EXPIRED }                 </pre>
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_EX_CP	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA_INVALID_EX_CP,     eumCertificate     #CERT_EUM_ECDSA }                 </pre>
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_EX_KU	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA_INVALID_EX_KU,     eumCertificate     #CERT_EUM_ECDSA }                 </pre>

	}
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_SIG</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA_INVALID_SIG,     eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_SUB_ORG</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA_INVALID_SUB_ORG,     eumCertificate     #CERT_EUM_ECDSA }         </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_SUB_SN</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA_INVALID_SUB_SN,     eumCertificate     #CERT_EUM_ECDSA }         </pre>

	}
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_3	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA_EXPIRED,     eumCertificate #CERT_EUM_ECDSA }         </pre>
AUTH_SERVER_RESP_DEF_DP_UC_8_1_6_1_CHA	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE_INVALID&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }         </pre>
AUTH_SERVER_RESP_DEF_DP_UC_8_1_6_1_SIG	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1_INVALID&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }         </pre>

<p>AUTH_SERVER_RESP_DEF_DP_UC_8_2_5_4_3</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2_PPR2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_10_1_3_9</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;INVALID_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_8_11_1_3_9</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_UNKNOWN }         </pre>
<p>AUTH_SERVER_RESP_DEF_DP_UC_OK</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         </pre>

	<pre> transactionId &lt;S_TRANSACTION_ID&gt;, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge &lt;SMDP_CHALLENGE&gt;, euiccInfo2 #S_EUICC_INFO2 ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 &lt;EUICC_SIGNATURE1&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA } </pre>
<p>AUTH_SERVER_RESP_MATCHING_ID_OMITTED</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2         ctxParams1         #CTX_PARAMS1_MATCHING_ID_OMITTED     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA } </pre>
<p>AUTH_SERVER_RESP_SMDS_UC_OK</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DP_ADDRESS,         serverChallenge         &lt;SMDP_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1 #CTX_PARAMS1_SMDS     },     euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA } </pre>
<p>CS_RESP_ERROR_8_1_6_1</p>	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId &lt;S_TRANSACTION_ID&gt;,         smdpOid #IUT_SM_DP_OID,         reason postponed     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE_INVALID&gt; } </pre>

	<pre>         }     </pre>
CS_RESP_ERROR_8_8_3_10	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId &lt;S_TRANSACTION_ID&gt;,         smdpOid &lt;INVALID_SM_DP_OID&gt;,         reason postponed     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }     </pre>
CS_RESP_ERROR_8_10_1_3_9	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId         &lt;INVALID_TRANSACTION_ID&gt;,         smdpOid #IUT_SM_DP_OID,         reason postponed     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }     </pre>
CS_RESP_OK_EU_REJ	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId &lt;S_TRANSACTION_ID&gt;,         smdpOid #IUT_SM_DP_OID,         reason endUserRejection     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }     </pre>
CS_RESP_OK_L_BPP_EXE_ERROR	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId &lt;S_TRANSACTION_ID&gt;,         smdpOid #IUT_SM_DP_OID,         reason loadBppExecutionError     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }     </pre>
CS_RESP_OK_M_DATA_MISMATCH	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId &lt;S_TRANSACTION_ID&gt;,         smdpOid #IUT_SM_DP_OID,         reason metadataMismatch     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }     </pre>

<p>CS_RESP_OK_POSTPONED</p>	<pre>resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #IUT_SM_DP_OID,     reason postponed   },   euiccCancelSessionSignature   &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }</pre>
<p>CS_RESP_OK_PPR_NOT_ALLOWED</p>	<pre>resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #IUT_SM_DP_OID,     reason pprNotAllowed   },   euiccCancelSessionSignature   &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }</pre>
<p>CS_RESP_OK_TIMEOUT</p>	<pre>resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #IUT_SM_DP_OID,     reason timeout   },   euiccCancelSessionSignature   &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }</pre>
<p>CS_RESP_OK_UNDEFINED</p>	<pre>resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #IUT_SM_DP_OID,     reason undefinedReason   },   euiccCancelSessionSignature   &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }</pre>
<p>CTX_PARAMS1_ACT_CODE</p>	<pre>ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {   matchingId #MATCHING_ID_1,   deviceInfo #S_DEVICE_INFO }</pre>
<p>CTX_PARAMS1_ACT_CODE_2</p>	<pre>ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {   matchingId #MATCHING_ID_2,   deviceInfo #S_DEVICE_INFO }</pre>
<p>CTX_PARAMS1_MATCHING_ID_EMPTY</p>	<pre>ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {</pre>



	<pre> matchingId #MATCHING_ID_EMPTY, deviceInfo #S_DEVICE_INFO } </pre>
CTX_PARAMS1_SMDS	<pre> ctx CtxParams1 ::= ctxParamsForCommonAuthentication : {   matchingId &lt;MATCHING_ID_EVENT&gt;,   deviceInfo #S_DEVICE_INFO } </pre>
EUICC_FIRMWARE_VER	0x01 00 00
EXT_CARD_RESOURCE_LIMITED_SPACE	<p>The Extended Card Resource Information according to ETSI TS 102 226 and set as:</p> <pre> 0x81 &lt;L&gt; #INSTALLED_PROFILES 0x82 &lt;L&gt; #NON_VOLATILE_MEM_LIMITED_SPACE 0x83 &lt;L&gt; #S_VOLATILE_MEM </pre>
INITIATE_AUTH_DS_OK	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDS_SIGNED1&gt;,   "serverSignature1" :   &lt;S_SMDS_SIGNATURE1&gt;,   "euiccCiPKIdTobeUsed" :   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" :   #CERT_S_SM_DSauth_ECDSA } -- NOTE: select the CI as defined in the note in the chapter 2.1.4 of SGP.23 </pre>
INITIATE_AUTH_DS_OK_1	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDS_SIGNED_ADDR1&gt;,   "serverSignature1" :   &lt;S_SMDS_SIGNATURE1&gt;,   "euiccCiPKIdTobeUsed" :   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" :   #CERT_S_SM_DSauth_ECDSA } -- NOTE: select the CI as defined in the note in the chapter 2.1.4 of SGP.23 </pre>
INITIATE_AUTH_INV_CERT_DS	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   } } </pre>

	<pre>     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDS_SIGNED1&gt;,   "serverSignature1" : &lt;S_SMDS_SIGNATURE1&gt;,   "euiccCiPKidTobeUsed" : &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;, -- NOTE: select the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; "serverCertificate" : #CERT_S_SM_DSauth_INV_SIGN } </pre>
<p>INITIATE_AUTH_INV_CI_DS</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDS_SIGNED1&gt;,   "serverSignature1" : &lt;S_SMDS_SIGNATURE1&gt;,   "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 &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; } </pre>
<p>INITIATE_AUTH_INV_SIGN_DS</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDS_SIGNED1&gt;,   "serverSignature1" : &lt;S_SMDS_SIGNATURE_INV&gt;,   "euiccCiPKidTobeUsed" : &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" : #CERT_S_SM_DSauth_ECDSA } -- NOTE: select the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; and choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate </pre>
<p>INITIATE_AUTH_INV_SMDS_ADDRESS</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   } } </pre>

	<pre>     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDS_SIGNED_INV_ADDR&gt;,   "serverSignature1" : &lt;S_SMDS_SIGNATURE1&gt;,   "euiccCiPKidTobeUsed" : &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" : #CERT_S_SM_DSauth_ECDSA } -- NOTE: select the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; and choose the #CERT_S_SM_DSauth_ECDSA leading to the same Root CI certificate </pre>
<p>INITIATE_AUTH_OK</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDP_SIGNED1&gt;,   "serverSignature1" : &lt;S_SMDP_SIGNATURE1&gt;,   "euiccCiPKidTobeUsed" : &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" : #CERT_S_SM_DPauth_ECDSA } -- NOTE: select the CI as defined in the note in the chapter 2.1.4 of SGP.23 </pre>
<p>INITIATE_AUTH_INV_CERT</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDP_SIGNED1&gt;,   "serverSignature1" : &lt;S_SMDP_SIGNATURE1&gt;,   "euiccCiPKidTobeUsed" : &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;, -- NOTE: select the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;   "serverCertificate" : #CERT_S_SM_DPauth_INV_SIGN } </pre>
<p>INITIATE_AUTH_INV_CI</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   } } </pre>

	<pre>     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDP_SIGNED1&gt;,   "serverSignature1" :   &lt;S_SMDP_SIGNATURE1&gt;,   "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 &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; } </pre>
<p>INITIATE_AUTH_INV_OID</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDP_SIGNED1&gt;,   "serverSignature1" :   &lt;S_SMDP_SIGNATURE1&gt;,   "euiccCiPKidTobeUsed" :   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" :   #CERT_S_SM_DP2auth_ECDSA } -- NOTE: select the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; -- NOTE: serverSignature1 SHALL be calculated correctly, using the secret key related to CERT_S_SM_DP2auth_ECDSA. </pre>
<p>INITIATE_AUTH_INV_SIGN</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" : &lt;S_SMDP_SIGNED1&gt;,   "serverSignature1" :   &lt;S_SMDP_SIGNATURE_INV&gt;,   "euiccCiPKidTobeUsed" :   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" :   #CERT_S_SM_DPauth_ECDSA } -- NOTE: select the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; and choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate </pre>

<p>INITIATE_AUTH_INV_SMDP+_ADDRESS</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "serverSigned1" :   &lt;S_SMDP_SIGNED_INV_ADDR&gt;,   "serverSignature1" :   &lt;S_SMDP_SIGNATURE1&gt;,   "euiccCiPKIdToBeUsed" :   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   "serverCertificate" :   #CERT_S_SM_DPauth_ECDSA } -- NOTE: select the CI Key ID in highest priority from the &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt; and choose the #CERT_S_SM_DPauth_ECDSA leading to the same Root CI certificate -- NOTE: serverSignature1 SHALL be calculated correctly, using &lt;S_SMDP_SIGNED_INV_ADDR&gt;.</pre>
<p>MATCHING_ID_EMPTY</p>	
<p>NON_VOLATILE_MEM_LIMITED_SPACE</p>	<p>'0x00 01'</p>
<p>PENDING_NOTIF_DEL1</p>	<pre>response PendingNotification ::= otherSignedNotification :{   tbsOtherNotification {     seqNumber &lt;SEQ_NUMBER&gt;,     profileManagementOperation {       notificationDelete     },     notificationAddress     #TEST_DP_ADDRESS1,     iccid #ICCID_OP_PROF1   },   euiccNotificationSignature   &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }</pre>
<p>PENDING_NOTIF_DEL2</p>	<pre>response PendingNotification ::= otherSignedNotification : {   tbsOtherNotification {     seqNumber &lt;SEQ_NUMBER&gt;,     profileManagementOperation     {       notificationDelete     },     notificationAddress     #TEST_DP_ADDRESS2,     iccid #ICCID_OP_PROF2   } }</pre>

	<pre> },   euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>PENDING_NOTIF_DEL5</p>	<pre> response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification {   seqNumber &lt;SEQ_NUMBER&gt;,   profileManagementOperation {     notificationDelete   },   notificationAddress   #TEST_DP_ADDRESS1,   iccid #ICCID_OP_PROF5 },   euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>PENDING_NOTIF_DEL6</p>	<pre> response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification {   seqNumber &lt;SEQ_NUMBER&gt;,   profileManagementOperation {     notificationDelete   },   notificationAddress   #TEST_DP_ADDRESS2,   iccid #ICCID_OP_PROF6 },   euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>PENDING_NOTIF_DIS1</p>	<pre> response PendingNotification ::= otherSignedNotification : {    tbsOtherNotification {     seqNumber &lt;SEQ_NUMBER&gt;,     profileManagementOperation {       notificationDisable     },     notificationAddress     #TEST_DP_ADDRESS1,     iccid #ICCID_OP_PROF1   },   euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }         </pre>

<p>PENDING_NOTIF_DIS5</p>	<pre> response PendingNotification ::= otherSignedNotification : {      tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation {             notificationDisable         },         notificationAddress         #TEST_DP_ADDRESS1,         iccid #ICCID_OP_PROF5     },     euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>PENDING_NOTIF_DIS8</p>	<pre> response PendingNotification ::= otherSignedNotification : {      tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation {             notificationDisable         },         notificationAddress         #TEST_DP_ADDRESS8,         iccid #ICCID_OP_PROF8     },     euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>PENDING_NOTIF_EN1</p>	<pre> response PendingNotification ::= otherSignedNotification : {     tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation {             notificationEnable         },         notificationAddress         #TEST_DP_ADDRESS1,         iccid #ICCID_OP_PROF1     },     euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>PENDING_NOTIF_EN2</p>	<pre> response PendingNotification ::= otherSignedNotification : {     tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation {                     </pre>

	<pre> notificationEnable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF2 }, euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>PENDING_NOTIF_EN5</p>	<pre> response PendingNotification ::= otherSignedNotification : {   tbsOtherNotification {     seqNumber &lt;SEQ_NUMBER&gt;,     profileManagementOperation {       notificationEnable     },     notificationAddress     #TEST_DP_ADDRESS1,     iccid #ICCID_OP_PROF5   },   euiccNotificationSignature   &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>PENDING_NOTIF_EN6</p>	<pre> response PendingNotification ::= otherSignedNotification : {   tbsOtherNotification {     seqNumber &lt;SEQ_NUMBER&gt;,     profileManagementOperation {       notificationEnable     },     notificationAddress     #TEST_DP_ADDRESS2,     iccid #ICCID_OP_PROF6   },   euiccNotificationSignature   &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>PP_VERSION</p>	<p>0x01 00 00</p>
<p>PREP_DOWNLOAD_RESP_8_1_6_1</p>	<pre> resp PrepareDownloadResponse ::= downloadResponseOk : {   euiccSigned2 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccOtpk &lt;BPP_OTPK_EUICC_ECKA&gt;   },   euiccSignature2   &lt;EUICC_SIGNATURE2_INVALID&gt; }                     </pre>



<p>PREP_DOWNLOAD_RESP_8_2_7_3_8</p>	<pre>resp PrepareDownloadResponse ::=   downloadResponseOk : {     euiccSigned2 {       transactionId &lt;S_TRANSACTION_ID&gt;,       euiccOtpk &lt;BPP_OTPK_EUICC_ECKA&gt;,       hashCc &lt;S_HASHED_CC_ERROR&gt;     },     euiccSignature2 &lt;EUICC_SIGNATURE2&gt;   } </pre>
<p>PREP_DOWNLOAD_RESP_8_10_1_3_9</p>	<pre>resp PrepareDownloadResponse ::=   downloadResponseOk : {     euiccSigned2 {       transactionId       &lt;INVALID_TRANSACTION_ID&gt;,       euiccOtpk &lt;BPP_OTPK_EUICC_ECKA&gt;     },     euiccSignature2 &lt;EUICC_SIGNATURE2&gt;   } </pre>
<p>PREP_DOWNLOAD_RESP</p>	<pre>resp PrepareDownloadResponse ::=   downloadResponseOk : {     euiccSigned2 {       transactionId &lt;S_TRANSACTION_ID&gt;,       euiccOtpk &lt;BPP_OTPK_EUICC_ECKA&gt;     },     euiccSignature2 &lt;EUICC_SIGNATURE2&gt;   } </pre>
<p>PREP_DOWNLOAD_RESP_CC</p>	<pre>resp PrepareDownloadResponse ::=   downloadResponseOk : {     euiccSigned2 {       transactionId &lt;S_TRANSACTION_ID&gt;,       euiccOtpk &lt;BPP_OTPK_EUICC_ECKA&gt;,       hashCc &lt;S_HASHED_CC&gt;     },     euiccSignature2 &lt;EUICC_SIGNATURE2&gt;   } </pre>
<p>PREP_DOWNLOAD_RESP_NEW_OTPK</p>	<pre>resp PrepareDownloadResponse ::=   downloadResponseOk : {     euiccSigned2 {       transactionId &lt;S_TRANSACTION_ID&gt;,       euiccOtpk &lt;OTPK_EUICC_ECKA_NEW&gt;     },     euiccSignature2 &lt;EUICC_SIGNATURE2&gt;   } </pre>
<p>PREP_DOWNLOAD_RESP_NEW_OTPK_CC</p>	<pre>resp PrepareDownloadResponse ::=   downloadResponseOk : {     euiccSigned2 {       transactionId &lt;S_TRANSACTION_ID&gt;,       euiccOtpk &lt;OTPK_EUICC_ECKA_NEW&gt;,       hashCc &lt;S_HASHED_CC&gt;     },     euiccSignature2 &lt;EUICC_SIGNATURE2&gt;   } </pre>

PROFILE_VERSION	0x02 01 00
RSP_CAPABILITY	<pre>rspCapability RspCapability ::= {     additionalProfile, rpmSupport,     testProfileSupport }</pre>
S_EUICC_INFO2_INSUF_MEM_ERROR	<pre>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,     sasAccreditationNumber     #SAS_ACREDITATION_NUMBER }</pre>
S_EUICC_INFO2_PPR2	<pre>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,     sasAccreditationNumber     #SAS_ACREDITATION_NUMBER }</pre>
S_EXT_CARD_RESOURCE	<p>The Extended Card Resource Information according to ETSI TS 102 226:</p> <pre>0x81 &lt;L&gt; #INSTALLED_PROFILES 0x82 &lt;L&gt; #S_NON_VOLATILE_MEM 0x83 &lt;L&gt; #S_VOLATILE_MEM</pre>
S_NON_VOLATILE_MEM	0xA0 00
S_PN_PIR_OK1	<pre>response PendingNotification ::= profileInstallationResult : {     profileInstallationResultData {         transactionId &lt;S_TRANSACTION_ID&gt;,</pre>

	<pre> notificationMetadata {   seqNumber &lt;SEQ_NUMBER&gt;,   profileManagementOperation {     notificationInstall   },   notificationAddress #IUT_SM_DP_ADDRESS,   iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult successResult : {   aid &lt;ISD_P_AID&gt;,   simaResponse #SIMA_RESULT_OK } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>S_PN_PIR_INVALID_TRANS_ID</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;INVALID_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult successResult : {       aid &lt;ISD_P_AID&gt;,       simaResponse #SIMA_RESULT_OK     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>S_PN_PIR_INCORRECT_INPUT_VALUES</p>	<pre> response PendingNotification ::= profileInstallationResult : profileInstallationResultData {   transactionId &lt;S_TRANSACTION_ID&gt;,   notificationMetadata {     seqNumber &lt;SEQ_NUMBER&gt;,     profileManagementOperation {       notificationInstall     },     notificationAddress #IUT_SM_DP_ADDRESS,     iccid #ICCID_OP_PROF1   },   smdpOid #IUT_SM_DP_OID,   finalResult errorResult : {     bppCommandId configureISDP,     errorReason incorrectInputValues   } } </pre>

	<pre> },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
<p>S_PN_PIR_INVALID_SIGN</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress     }   },   #IUT_SM_DP_ADDRESS,   iccid #ICCID_OP_PROF1 },   smdpOid #IUT_SM_DP_OID,   finalResult errorResult : {     bppCommandId   },   initialiseSecureChannel,   errorReason invalidSignature },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>

<p>S_PN_PIR_UNSUPPORTED_CRT</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress       #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {       bppCommandId     }   }   initialiseSecureChannel,   errorReason unsupportedCrtValues } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
<p>S_PN_PIR_UNSUP_REMOTE_OP_TYPE</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress       #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {       bppCommandId     }   }   initialiseSecureChannel,   errorReason   unsupportedRemoteOperationType } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
<p>S_PN_PIR_UNSUP_PROFILE_CLASS</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress       #IUT_SM_DP_ADDRESS,         </pre>

	<pre>         iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {         bppCommandId storeMetadata,         errorReason unsupportedProfileClass     }     },     euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }     </pre>
<p>S_PN_PIR_SCP03T_STRUCTURE_ERROR</p>	<pre> response PendingNotification ::= profileInstallationResult : {     profileInstallationResultData {         transactionId &lt;S_TRANSACTION_ID&gt;,         notificationMetadata {             seqNumber &lt;SEQ_NUMBER&gt;,             profileManagementOperation {                 notificationInstall             },             notificationAddress             #IUT_SM_DP_ADDRESS,             iccid #ICCID_OP_PROF1         },         smdpOid #IUT_SM_DP_OID,         finalResult errorResult : {             bppCommandId storeMetadata,             errorReason scp03tStructureError         }     },     euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }     </pre>
<p>S_PN_PIR_SCP03T_SECURITY_ERROR</p>	<pre> response PendingNotification ::= profileInstallationResult : {     profileInstallationResultData {         transactionId &lt;S_TRANSACTION_ID&gt;,         notificationMetadata {             seqNumber &lt;SEQ_NUMBER&gt;,             profileManagementOperation {                 notificationInstall             },             notificationAddress             #IUT_SM_DP_ADDRESS,             iccid #ICCID_OP_PROF1         },         smdpOid #IUT_SM_DP_OID,         finalResult errorResult : {             bppCommandId replaceSessionKeys,             errorReason scp03tSecurityError         }     },     euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }     </pre>
<p>S_PN_PIR_ICCID_ALREADY_EXISTS</p>	<pre> response PendingNotification ::= profileInstallationResult : {     profileInstallationResultData {         transactionId &lt;S_TRANSACTION_ID&gt;,     </pre>

	<pre> notificationMetadata {   seqNumber &lt;SEQ_NUMBER&gt;,   profileManagementOperation {     notificationInstall   },   notificationAddress #IUT_SM_DP_ADDRESS,   iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : {   bppCommandId storeMetadata,   errorReason }  installFailedDueToIccidAlreadyExistsOnEuicc } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>S_PN_PIR_INSUFFICIENT_MEMORY</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {       bppCommandId storeMetadata,       errorReason     }   }   installFailedDueToInsufficientMemoryForProfile } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>S_PN_PIR_INSTALL_INTERRUPTION</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID, </pre>

	<pre> finalResult errorResult : {   bppCommandId storeMetadata,   errorReason   installFailedDueToInterruption } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
<p>S_PN_PIR_PE_PROCESSING_ERROR</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress       #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {       bppCommandId loadProfileElements,       errorReason       installFailedDueToPEProcessingError     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
<p>S_PN_PIR_DATA_MISMATCH</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress       #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {       bppCommandId loadProfileElements,       errorReason       installFailedDueToDataMismatch     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
<p>S_PN_PIR_TEST_PROFILE_INVALID_NAA_KEY</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,         </pre>



	<pre> notificationMetadata {   seqNumber &lt;SEQ_NUMBER&gt;,   profileManagementOperation {     notificationInstall   },   notificationAddress #IUT_SM_DP_ADDRESS,   iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : {   bppCommandId loadProfileElements,   errorReason testProfileInstallFailedDueToInvalidNaaKey } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>S_PN_PIR_PPR_NOT_ALLOWED</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {       bppCommandId storeMetadata,       errorReason pprNotAllowed     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>S_PN_PIR_UNKNOWN_ERROR</p>	<pre> response PendingNotification ::= profileInstallationResult : {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #IUT_SM_DP_ADDRESS,       iccid #ICCID_OP_PROF1     },     smdpOid #IUT_SM_DP_OID,     finalResult errorResult : {       bppCommandId storeMetadata,       errorReason installFailedDueToUnknownError } } </pre>

	<pre>         }     },     euiccSignPIR &lt;EUICC_SIGN_PIR&gt;     }     </pre>
<p>S_PENDING_NOTIF_OTHER_INST1</p>	<pre> response PendingNotification ::= otherSignedNotification : {     tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation         {             notificationInstall         },         notificationAddress         #IUT_SM_DP_ADDRESS,         iccid #ICCID_OP_PROF1     },     euiccNotificationSignature     &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }     </pre>
<p>S_PENDING_NOTIF_EN1</p>	<pre> response PendingNotification ::= otherSignedNotification : {     tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation {             notificationEnable         },         notificationAddress         #IUT_SM_DP_ADDRESS,         iccid #ICCID_OP_PROF1     },     euiccNotificationSignature     &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }     </pre>
<p>S_PENDING_NOTIF_DIS1</p>	<pre> response PendingNotification ::= otherSignedNotification : {     tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation {             notificationDisable         },         notificationAddress         #IUT_SM_DP_ADDRESS,         iccid #ICCID_OP_PROF1     },     euiccNotificationSignature     &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }     </pre>

S_PENDING_NOTIF_DE1	<pre> response PendingNotification ::= otherSignedNotification :{   tbsOtherNotification {     seqNumber &lt;SEQ_NUMBER&gt;,     profileManagementOperation {       notificationDelete     },     notificationAddress #IUT_SM_DP_ADDRESS,     iccid #ICCID_OP_PROF1   },   euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } </pre>
S_SMDP_SIGNED2	<pre> req SmdpSigned2 ::= {   transactionId &lt;S_TRANSACTION_ID&gt;,   ccRequiredFlag FALSE } </pre>
S_SMDP_SIGNED2_CC	<pre> req SmdpSigned2 ::= {   transactionId &lt;S_TRANSACTION_ID&gt;,   ccRequiredFlag TRUE } </pre>
S_SMDP_SIGNED2_INV_TRANSACTION_ID	<pre> req SmdpSigned2 ::= {   transactionId &lt;INVALID_TRANSACTION_ID&gt;,   ccRequiredFlag FALSE } </pre>
S_VOLATILE_MEM	'0x01 00'
SAS_ACREDITATION_NUMBER	GSMA_SAS_123456789
UICC_CAPABILITY	<pre> uiccCapability UICCCapability ::= {   contactlessSupport, usimSupport,   isimSupport,   akaMilenage, akaTuak128,   gbaAuthenUsim, eapClient,   javacard, multipleUsimSupport } </pre>

## D.2.2 ES9+ Responses

Name	Content
AUTH_CLIENT_OK	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "profileMetadata" : #METADATA_OP_PROF1, </pre>

	<pre>"smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : &lt;S_SM_DP+_SIGNATURE2&gt;, "smdpCertificate" : #CERT_S_SM_DPpb_ECDSA }</pre>
<p>AUTH_CLIENT_OK_CC</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2_CC, "smdpSignature2" : &lt;S_SM_DP+_SIGNATURE2&gt;, "smdpCertificate" : #CERT_S_SM_DPpb_ECDSA }</pre>
<p>AUTH_CLIENT_INV_PB_CERT</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : &lt;S_SM_DP+_SIGNATURE2&gt;, "smdpCertificate" : #CERT_S_SM_DPpb_INV_SIGN }</pre>
<p>AUTH_CLIENT_INV_CI</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : &lt;S_SM_DP+_SIGNATURE2&gt;, "smdpCertificate" : #CERT_S_SM_DP2pb_ECDSA }</pre>
<p>AUTH_CLIENT_INV_SIGN</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   } }</pre>

	<pre>         }     },     "transactionId" : &lt;S_TRANSACTION_ID&gt;,     "profileMetadata" :     #METADATA_OP_PROF1,     "smdpSigned2" : #S_SMDP_SIGNED2,     "smdpSignature2" :     &lt;S_SM_DP+_SIGNATURE2&gt;,     "smdpCertificate" :     #CERT_S_SM_DPpb_ECDSA     }      The &lt;S_SM_DP+_SIGNATURE2&gt; SHALL NOT be     computed using the #SK_S_SM_DPpb_ECDSA but     SHALL have the same length as for a valid     signature     </pre>
<p>AUTH_CLIENT_INV_TRANSACTION_ID</p>	<pre>     {         "header" : {             "functionExecutionStatus" : {                 "status" : "Executed-Success"             }         },         "transactionId" :         &lt;S_TRANSACTION_ID&gt;,         "profileMetadata" :         #METADATA_OP_PROF1,         "smdpSigned2" :         #S_SMDP_SIGNED2_INV_TRANSACTION_ID,         "smdpSignature2" :         &lt;S_SM_DP+_SIGNATURE2&gt;,         "smdpCertificate" :         #CERT_S_SM_DPpb_ECDSA     }     </pre>
<p>CS_OK_EU_LOAD_BPP_ERROR</p>	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId &lt;S_TRANSACTION_ID&gt;,         smdpOid #S_SM_DP+_OID,         reason loadBppExecutionError     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }     </pre>
<p>CS_OK_EU_POSTPONED</p>	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {     euiccCancelSessionSigned {         transactionId &lt;S_TRANSACTION_ID&gt;,         smdpOid #S_SM_DP+_OID,         reason postponed     },     euiccCancelSessionSignature     &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }     </pre>
<p>CS_OK_EU_REJ</p>	<pre> resp CancelSessionResponse ::=     </pre>

	<pre>cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason endUserRejection   },   euiccCancelSessionSignature   &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }</pre>
<p>CS_OK_PPR_NOT_ALLOWED</p>	<pre>resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason pprNotAllowed   },   euiccCancelSessionSignature   &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }</pre>
<p>CS_OK_TIMEOUT</p>	<pre>resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason timeout   },   euiccCancelSessionSignature   &lt;EUICC_CANCEL_SESSION_SIGNATURE&gt; }</pre>
<p>GET_BPP_LOAD_ERROR</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "boundProfilePackage" :   BoundProfilePackage {     #S_INIT_SC_PROF1,     firstSequenceOf87 {       #CONF_ISDP_PROF1     },     sequenceOf88 {       &lt;METADATA_OP_PROF1_SEG&gt;       ...       &lt;METADATA_OP_PROF1_SEG&gt;     }   } }</pre> <p>NOTE 1: boundProfilePackage is encoded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1.</p> <p>NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending</p>

<p>GET_BPP_LOAD_ERROR_UNKNOWN_TAG</p>	<p>on the size of StoreMetadata.</p> <pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "boundProfilePackage" {     #S_INIT_SC_PROF1,     #UNKNOWN_BPP_SEGMENT,     firstSequenceOf87 {       #CONF_ISDP_PROF1     },     sequenceOf88 {       &lt;METADATA_OP_PROF1_SEG&gt;       ...       &lt;METADATA_OP_PROF1_SEG&gt;     },     sequenceOf86 {       &lt;PPP_OP_PROF1_SEG_SK&gt;       ...       &lt;PPP_OP_PROF1_SEG_SK&gt;     }   } }</pre> <p>NOTE 1: boundProfilePackage is encoded as base64 therefore the test tool shall decode boundProfilePackage to access the ASN.1.</p> <p>NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.</p>
<p>GET_BPP_OK</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "boundProfilePackage" :   BoundProfilePackage {     #S_INIT_SC_PROF1,     firstSequenceOf87 {       #CONF_ISDP_PROF1     },     sequenceOf88 {       &lt;METADATA_OP_PROF1_SEG&gt;       ...       &lt;METADATA_OP_PROF1_SEG&gt;     },     sequenceOf86 {       &lt;PPP_OP_PROF1_SEG_SK&gt;       ...       &lt;PPP_OP_PROF1_SEG_SK&gt;     }   } }</pre> <p>NOTE 1: boundProfilePackage is encoded as base64 therefore the test tool SHALL decode</p>

	<p>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.</p>
<p>GET_BPP_OK_PPK</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "boundProfilePackage" :   BoundProfilePackage {     #S_INIT_SC_PROF1,     firstSequenceOf87 {       0x87 &lt;L&gt; #CONF_ISDP_PROF1     },     sequenceOf88 {       &lt;METADATA_OP_PROF1_SEG&gt;       ...       &lt;METADATA_OP_PROF1_SEG&gt;     },     secondSequenceOf87 {       0x87 &lt;L&gt; #REPLACE_S_KEYS_REQ     },     sequenceOf86 {       &lt;PPP_OP_PROF1_SEG_SK&gt;       ...       &lt;PPP_OP_PROF1_SEG_SK&gt;     }   } }</pre>
<p>GET_BPP_INV</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "boundProfilePackage" :   BoundProfilePackage {     #S_INIT_SC_PROF1,     firstSequenceOf87 {       0x87 &lt;L&gt; #CONF_ISDP_PROF1     },     sequenceOf88 {       &lt;METADATA_OP_PROF1_SEG&gt;       ...       &lt;METADATA_OP_PROF1_SEG&gt;     },     sequenceOf86 {       &lt;PPP_OP_PROF1_SEG_SK_INV&gt;       ...       &lt;PPP_OP_PROF1_SEG_SK_INV&gt;     }   } }</pre>



	<pre> } </pre>
PENDING_NOTIF_INST_ADDRESS2	<pre> response PendingNotification ::= otherSignedNotification : {     tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation         {             notificationInstall         },         notificationAddress #TEST_DP_ADDRESS2,         iccid #ICCID_OP_PROF1     },     euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA } </pre>
PENDING_NOTIF_INST1	<pre> response PendingNotification ::= otherSignedNotification : {     tbsOtherNotification {         seqNumber &lt;SEQ_NUMBER&gt;,         profileManagementOperation         {             notificationInstall         },         notificationAddress #TEST_DP_ADDRESS1,         iccid #ICCID_OP_PROF1     },     euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA } </pre>
R_AUTH_CLIENT_META_ABS	<pre> {     "header" : {         "functionExecutionStatus" : {             "status" : "Executed-Success"         }     },     "transactionId" : &lt;TRANSACTION_ID_AC&gt;,     "profileMetadata" : #SMDP_METADATA_ABS,     "smdpSigned2" : #SMDP_SIGNED2,     "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;,     "smdpCertificate" : #CERT_SM_DPpb_ECDSA } </pre>
R_AUTH_CLIENT_META_ALL	<pre> {     "header" : {         "functionExecutionStatus" : {             "status" : "Executed-Success"         }     } } </pre>

	<pre> }, "transactionId" : &lt;TRANSACTION_ID_AC&gt;, "profileMetadata" : #SMDP_METADATA_ALL, "smdpSigned2" : #SMDP_SIGNED2, "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;, "smdpCertificate" : #CERT_SM_DPpb_ECDSA }                 </pre>
<p>R_AUTH_CLIENT_META_LARGE_ICON</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" : #SMDP_METADATA_OP_PROF1_2_SEG,   "smdpSigned2" : #SMDP_SIGNED2,   "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;,   "smdpCertificate" : #CERT_SM_DPpb_ECDSA }                 </pre>
<p>R_AUTH_CLIENT_META_NON_ASCII</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" : #SMDP_METADATA_NON_ASCII,   "smdpSigned2" : #SMDP_SIGNED2,   "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;,   "smdpCertificate" : #CERT_SM_DPpb_ECDSA }                 </pre>
<p>R_AUTH_CLIENT_META_NOTIF_MULTI</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" : #SMDP_METADATA_NOTIF_MULTI,   "smdpSigned2" : #SMDP_SIGNED2,   "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;,   "smdpCertificate" : #CERT_SM_DPpb_ECDSA }                 </pre>
<p>R_AUTH_CLIENT_META_PN_LONG</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" :                 </pre>

	<pre>#SMDP_METADATA_PN_LONG, "smdpSigned2" : #SMDP_SIGNED2, "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;, "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
<p>R_AUTH_CLIENT_META_SPN_LONG</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" : #SMDP_METADATA_SPN_LONG, "smdpSigned2" : #SMDP_SIGNED2, "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;, "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
<p>R_AUTH_CLIENT_OK</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" : #SMDP_METADATA_OP_PROF1, "smdpSigned2" : #SMDP_SIGNED2, "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;, "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
<p>R_AUTH_CLIENT_OK_CC</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" : #SMDP_METADATA_OP_PROF1, "smdpSigned2" : #SMDP_SIGNED2_CC, "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;, "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
<p>R_AUTH_CLIENT_OK_EN</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" : #SMDP_METADATA_OP_PROF1_EN, "smdpSigned2" : #SMDP_SIGNED2,</pre>

	<pre>"smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;, "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
R_AUTH_CLIENT_OK_PPR2	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" :   #SMDP_METADATA_OP_PROF1_PPR2,   "smdpSigned2" : #SMDP_SIGNED2,   "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;,   "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
R_AUTH_CLIENT_RETRY_OK	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" :   #SMDP_METADATA_OP_PROF1,   "smdpSigned2" : #SMDP_SIGNED2_RETRY,   "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;,   "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
R_AUTH_CLIENT_RETRY_OK_CC	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_AC&gt;,   "profileMetadata" :   #SMDP_METADATA_OP_PROF1,   "smdpSigned2" : #SMDP_SIGNED2_CC_RETRY,   "smdpSignature2" : &lt;SMDP_SIGNATURE2&gt;,   "smdpCertificate" : #CERT_SM_DPpb_ECDSA }</pre>
R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO	<pre>resp AuthenticateServerResponse ::authenticateResponseOk : {   euiccSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     serverAddress #TEST_ROOT_DS_ADDRESS,     serverChallenge &lt;S_SMDS_CHALLENGE&gt;,     euiccInfo2 #R_EUICC_INFO2, -- check     only that the field is present but not the     values     ctxParams1   }   #CTX_PARAMS1_MATCH_ID_DEV_INFO },</pre>

	<pre> euiccSignature1 &lt;EUICC_SIGNATURE1&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO_1</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     serverAddress #TEST_DS_ADDRESS1,     serverChallenge &lt;S_SMDS_CHALLENGE&gt;,     euiccInfo2 #R_EUICC_INFO2, -- check only that the field is present but not the values     ctxParams1 #CTX_PARAMS1_MATCH_ID_DEV_INFO   },   euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>R_AUTH_SERVER_MATCH_ID_DEV_INFO</p>	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk {   euiccSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     serverAddress #TEST_DP_ADDRESS1,     serverChallenge &lt;S_SMDP_CHALLENGE&gt;,     euiccInfo2 #R_EUICC_INFO2, -- check only that the field is present but not the values     ctxParams1 #CTX_PARAMS1_MATCH_ID_DEV_INFO   },   euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>R_GET_BPP_RESP_OP1_PPK          (Pre-generated PPP for Profiles)</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId": &lt;TRANSACTION_ID_GBPP&gt;,   "boundProfilePackage" : BoundProfilePackage {   #INIT_SC_PROF1,   firstSequenceOf87 {     &lt;CONF_ISDP_PROF1_ENC&gt;   },   sequenceOf88 {     &lt;SMDP_METADATA_SEG_MAC&gt;     ...   } }         </pre>

	<pre>         &lt;SMDP_METADATA_SEG_MAC&gt;     },     secondSequenceOf87 {         &lt;REPLACE_S_KEYS_REQ_ENC&gt;     },     sequenceOf86 {         &lt;PPP_OP_PROF1_SEG_PPK&gt;         ...         &lt;PPP_OP_PROF1_SEG_PPK&gt;     } }     </pre> <p><b>NOTE 1:</b> boundProfilePackage is encoded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1.</p> <p><b>NOTE 2:</b> For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.</p>
<p>R_GET_BPP_RESP_OP1_SK                  (Dynamically-generated PPP for Profiles)</p>	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId": &lt;TRANSACTION_ID_GBPP&gt;,   "boundProfilePackage" :   BoundProfilePackage {     #INIT_SC_PROF1,     firstSequenceOf87 {       &lt;CONF_ISDP_PROF1_ENC&gt;     },     sequenceOf88 {       &lt;SMDP_METADATA_SEG_MAC&gt;       ...       &lt;SMDP_METADATA_SEG_MAC&gt;     },     sequenceOf86 {       &lt;PPP_OP_PROF1_SEG_SK&gt;       ...       &lt;PPP_OP_PROF1_SEG_SK&gt;     }   } }     </pre> <p><b>NOTE 1:</b> boundProfilePackage is encoded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1.</p> <p><b>NOTE 2:</b> For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.</p>
<p>R_HTTP_204_OK</p>	<p>HTTP/1.1 204 No Content</p> <p>X-Admin-Protocol: gsma/rsp/v#RSP_SVN</p>

	<p><b>NOTE:</b> If the HTTP response is being received from the server under test, then the "Content-type" header MAY be present.</p>
<p>R_INITIATE_AUTH_OK</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_IA&gt;,   "serverSigned1" : #SERVER_SIGNED1,   "serverSignature1" :   &lt;SERVER_SIGNATURE1&gt;,   "euiccCiPKIdTobeUsed" : #CI_PKI_ID1,   "serverCertificate" :   #CERT_SM_XXauth_ECDSA }</pre>
<p>R_INITIATE_AUTH_OK_2</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;TRANSACTION_ID_2&gt;,   "serverSigned1" : #SERVER_SIGNED1_2,   "serverSignature1" :   &lt;SERVER_SIGNATURE1_2&gt;,   "euiccCiPKIdTobeUsed" : #CI_PKI_ID1,   "serverCertificate" :   #CERT_SM_XXauth_ECDSA }</pre>
<p>SERVER_SIGNED1</p>	<p>For InitiateAuthentication testing XX = IA, and for AuthenticateClient testing XX = AC:</p> <pre>ssl ServerSigned1 ::= {   transactionId   &lt;TRANSACTION_ID_SIGNED_IA&gt;,   euiccChallenge #S_EUICC_CHALLENGE,   serverAddress   #SERVER_ADDRESS,   serverChallenge &lt;SERVER_CHALLENGE&gt; }</pre>
<p>SERVER_SIGNED1_2</p>	<pre>ssl_2 ServerSigned1 ::= {   transactionId   &lt;TRANSACTION_ID_SIGNED_2&gt;,   euiccChallenge #S_EUICC_CHALLENGE_2,   serverAddress #SERVER_ADDRESS,   serverChallenge &lt;SERVER_CHALLENGE_2&gt; }</pre>
<p>SMDP_METADATA_OP_PROF1</p>	<pre>metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF1,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   profileClass operational }</pre>

	} }
SMDP_METADATA_OP_PROF1_2_SEG	<pre> 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 } } </pre>
SMDP_METADATA_OP_PROF3	<pre> metadataReq StoreMetadataRequest ::= {   iccid #ICCID_OP_PROF3,   serviceProviderName #SP_NAME3,   profileName #NAME_OP_PROF3,   profileClass operational,   profileOwner {     mccMnc #MCC_MNC2   },   profilePolicyRules { ppr2 } } </pre>
SMDP_SIGNED2	<pre> smdpSigned2 SmdpSigned2 ::= {   transactionId   &lt;TRANSACTION_ID_SIGNED_AC&gt;,   ccRequiredFlag FALSE } </pre>
SMDP_SIGNED2_CC	<pre> smdpSigned2 SmdpSigned2 ::= {   transactionId   &lt;TRANSACTION_ID_SIGNED_AC&gt;,   ccRequiredFlag TRUE } </pre>
SMDP_SIGNED2_CC_RETRY	<pre> smdpSigned2 SmdpSigned2 ::= {   transactionId   &lt;TRANSACTION_ID_SIGNED_AC&gt;,   ccRequiredFlag TRUE,   bppEuiccOtpk &lt;BPP_OTPK_EUICC_ECKA&gt; } </pre>
SMDP_SIGNED2_RETRY	<pre> smdpSigned2 SmdpSigned2 ::= {   transactionId } </pre>



	<pre>&lt;TRANSACTION_ID_SIGNED_AC&gt;, ccRequiredFlag FALSE, bppEuiccOtpk &lt;BPP_OTPK_EUICC_ECKA&gt; }</pre>
--	---

### D.3 ES10x Requests And Responses

#### D.3.1 ES10x Requests

Name	Content
AUTH_SMDP_MATCH_ID	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge &lt;EUICC_CHALLENGE&gt;,     serverAddress #TEST_DP_ADDRESS1,     serverChallenge &lt;S_SMDP_CHALLENGE&gt;   },   serverSignature1 &lt;S_SMDP_SIGNATURE1&gt;,   euiccCiPKIdToBeUsed   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   serverCertificate #CERT_S_SM_DPauth_ECDSA,   ctxParams1 #CTX_PARAMS1_MATCH_ID }</pre>
AUTH_SMDP_IMEI	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge &lt;EUICC_CHALLENGE&gt;,     serverAddress #TEST_DP_ADDRESS1,     serverChallenge &lt;S_SMDP_CHALLENGE&gt;   },   serverSignature1 &lt;S_SMDP_SIGNATURE1&gt;,   euiccCiPKIdToBeUsed   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   serverCertificate #CERT_S_SM_DPauth_ECDSA,   ctxParams1 #CTX_PARAMS1_IMEI }</pre>
AUTH_SMDP_INV_CERT	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge &lt;EUICC_CHALLENGE&gt;,     serverAddress #TEST_DP_ADDRESS1,     serverChallenge &lt;S_SMDP_CHALLENGE&gt;   },   serverSignature1 &lt;S_SMDP_SIGNATURE1&gt;,   euiccCiPKIdToBeUsed   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   serverCertificate #CERT_S_SM_DPauth_INV_SIGN,   ctxParams1 #CTX_PARAMS1 }</pre>
AUTH_SMDP_INV_CURV	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge &lt;EUICC_CHALLENGE&gt;,     serverAddress #TEST_DP_ADDRESS1,</pre>

	<pre> serverChallenge &lt;S_SMDP_CHALLENGE&gt;     }, serverSignature1 &lt;RANDOM_SM_DP+_SIGN&gt;, euiiccCiPKIdToBeUsed &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;, serverCertificate #CERT_S_SM_DPauth_INV_CURVE, ctxParams1 #CTX_PARAMS1     }         </pre>
<p>AUTH_SMDP_INV_CHALLENGE</p>	<pre> req AuthenticateServerRequest ::= {     serverSigned1 {         transactionId &lt;S_TRANSACTION_ID&gt;,         euiiccChallenge #S_EUICC_CHALLENGE,         serverAddress #TEST_DP_ADDRESS1,         serverChallenge &lt;S_SMDP_CHALLENGE&gt;     },     serverSignature1 &lt;S_SMDP_SIGNATURE1&gt;,     euiiccCiPKIdToBeUsed     &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,     serverCertificate #CERT_S_SM_DPauth_ECDSA,     ctxParams1 #CTX_PARAMS1 }         </pre>
<p>AUTH_SMDP_INV_OID</p>	<pre> req AuthenticateServerRequest ::= {     serverSigned1 {         transactionId &lt;S_TRANSACTION_ID&gt;,         euiiccChallenge &lt;EUICC_CHALLENGE&gt;,         serverAddress #TEST_DP_ADDRESS1,         serverChallenge &lt;S_SMDP_CHALLENGE&gt;     },     serverSignature1 &lt;S_SMDP_SIGNATURE1&gt;,     euiiccCiPKIdToBeUsed     &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,     serverCertificate #CERT_S_SM_DPpb_ECDSA,     ctxParams1 #CTX_PARAMS1 }         </pre>
<p>AUTH_SMDS_IMEI</p>	<pre> req AuthenticateServerRequest ::= {     serverSigned1 {         transactionId &lt;S_TRANSACTION_ID&gt;,         euiiccChallenge &lt;EUICC_CHALLENGE&gt;,         serverAddress #TEST_ROOT_DS_ADDRESS,         serverChallenge &lt;S_SMDS_CHALLENGE&gt;     },     serverSignature1 &lt;S_SMDS_SIGNATURE1&gt;,     euiiccCiPKIdToBeUsed     &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,     serverCertificate #CERT_S_SM_DSauth_ECDSA,     ctxParams1 #CTX_PARAMS1_EVENT_ID_IMEI }         </pre>
<p>AUTH_SMDS_INV_CERT</p>	<pre> req AuthenticateServerRequest ::= {     serverSigned1 {         transactionId &lt;S_TRANSACTION_ID&gt;,         euiiccChallenge &lt;EUICC_CHALLENGE&gt;,         serverAddress #TEST_ROOT_DS_ADDRESS,         serverChallenge &lt;S_SMDS_CHALLENGE&gt;     },         </pre>

	<pre>serverSignature1 &lt;S_SMDS_SIGNATURE1&gt;, euiccCiPKIdToBeUsed &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;, serverCertificate #CERT_S_SM_DSauth_INV_SIGN, ctxParams1 #CTX_PARAMS1_EVENT_ID }</pre>
<p>AUTH_SMDS_INV_CHALLENGE</p>	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge #S_EUICC_CHALLENGE,     serverAddress #TEST_ROOT_DS_ADDRESS,     serverChallenge &lt;S_SMDS_CHALLENGE&gt;   },   serverSignature1 &lt;S_SMDS_SIGNATURE1&gt;,   euiccCiPKIdToBeUsed   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   serverCertificate #CERT_S_SM_DSauth_ECDSA,   ctxParams1 #CTX_PARAMS1_EVENT_ID }</pre>
<p>AUTH_SMDS_INV_CURV</p>	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge &lt;EUICC_CHALLENGE&gt;,     serverAddress #TEST_ROOT_DS_ADDRESS,     serverChallenge &lt;S_SMDS_CHALLENGE&gt;   },   serverSignature1 &lt;RANDOM_SM_DS_SIGN&gt;,   euiccCiPKIdToBeUsed &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   serverCertificate #CERT_S_SM_DSauth_INV_CURVE,   ctxParams1 #CTX_PARAMS1_EVENT_ID }</pre>
<p>AUTHENTICATE_SMDP</p>	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge &lt;EUICC_CHALLENGE&gt;,     serverAddress #TEST_DP_ADDRESS1,     serverChallenge &lt;S_SMDP_CHALLENGE&gt;   },   serverSignature1 &lt;S_SMDP_SIGNATURE1&gt;,   euiccCiPKIdToBeUsed   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   serverCertificate #CERT_S_SM_DPauth_ECDSA,   ctxParams1 #CTX_PARAMS1 }</pre>
<p>AUTHENTICATE_SMDS</p>	<pre>req AuthenticateServerRequest ::= {   serverSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccChallenge &lt;EUICC_CHALLENGE&gt;,     serverAddress #TEST_ROOT_DS_ADDRESS,     serverChallenge &lt;S_SMDS_CHALLENGE&gt;   },   serverSignature1 &lt;S_SMDS_SIGNATURE1&gt;,   euiccCiPKIdToBeUsed   &lt;EUICC_CI_PK_ID_TO_BE_USED&gt;,   serverCertificate #CERT_S_SM_DSauth_ECDSA,</pre>

	<pre>                 ctxParams1 #CTX_PARAMS1_EVENT_ID             }         </pre>
CANCEL_SESSION_INV_TRANS_ID	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;INVALID_TRANSACTION_ID&gt;,                     reason endUserRejection                 }             </pre>
CANCEL_SESSION_REJECT	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;S_TRANSACTION_ID&gt;,                     reason endUserRejection                 }             </pre>
CANCEL_SESSION_POSTPONED	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;S_TRANSACTION_ID&gt;,                     reason postponed                 }             </pre>
CANCEL_SESSION_TIMEOUT	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;S_TRANSACTION_ID&gt;,                     reason timeout                 }             </pre>
CANCEL_SESSION_PPR	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;S_TRANSACTION_ID&gt;,                     reason pprNotAllowed                 }             </pre>
CANCEL_SESSION_METADATA	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;S_TRANSACTION_ID&gt;,                     reason metadataMismatch                 }             </pre>
CANCEL_SESSION_LOAD_BPP	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;S_TRANSACTION_ID&gt;,                     reason loadBppExecutionError                 }             </pre>
CANCEL_SESSION_UNDEF	<pre>                 req CancelSessionRequest ::= {                     transactionId &lt;S_TRANSACTION_ID&gt;,                     reason undefinedReason                 }             </pre>
EUICC_MEMORY_RESET	<pre>                 req EuiccMemoryResetRequest ::= {                     resetOptions {                         deleteOperationalProfiles,                         resetDefaultSmdpAddress                     }                 }             </pre>
EUICC_MEMORY_RESET_DEF_SMDP ADDRESS	<pre>                 req EuiccMemoryResetRequest ::= {                     resetOptions { resetDefaultSmdpAddress }                 }             </pre>
EUICC_MEMORY_RESET_OP_PRO	<pre>                 req EuiccMemoryResetRequest ::= {                     resetOptions { deleteOperationalProfiles }                 }             </pre>

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_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_TAGLIS T1	request ProfileInfoListRequest ::= { searchCriteria iccid: #ICCID_OP_PROF1, tagList '9F70'H --state }
GET_PROFILES_INFO_ICCID_TAGLIS T2	request ProfileInfoListRequest ::= { searchCriteria iccid: #ICCID_OP_PROF1, tagList '93'H --icon type }
GET_PROFILES_INFO_ICCID_TAGLIS T3	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_ISDP AID	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	request ProfileInfoListRequest ::= {

FILE_OWNER	<pre>tagList 'B7'H }</pre>
GET_PROFILES_INFO_TAGLIST_SMD P_PROP_DATA	<pre>request ProfileInfoListRequest ::= {   tagList 'B8'H }</pre>
GET_PROFILES_INFO_TAGLIST_SP_N AME	<pre>request ProfileInfoListRequest ::= {   tagList '91'H }</pre>
GET_PROFILES_INFO_TAGLIST1	<pre>request ProfileInfoListRequest ::= {   tagList '5A9F70'H -- ICCID and State }</pre>
GET_PROFILES_INFO_TAGLIST2	<pre>request ProfileInfoListRequest ::= {   tagList '909F70'H --Nickname and State }</pre>
GET_PROFILES_INFO_TAGLIST3	<pre>request ProfileInfoListRequest ::= {   tagList '9493'H --Icon, Icon type }</pre>
GET_PROFILES_INFO_TAGLIST4	<pre>request ProfileInfoListRequest ::= {   tagList '949F70'H --Icon, state }</pre>
GET_PROFILES_OWNERS	<pre>request ProfileInfoListRequest ::= {   tagList 'B7'H }</pre>
GET_RAT	<pre>request GetRatRequest ::= {}</pre>
LIST_NOTIF_ALL	<pre>request ListNotificationRequest ::= {   profileManagementOperation {     notificationInstall,     notificationEnable,     notificationDisable,     notificationDelete   } }</pre>
LIST_NOTIF_OMITTED	<pre>request ListNotificationRequest ::= {}</pre>
LIST_NOTIF_NONE	<pre>request ListNotificationRequest ::= {   profileManagementOperation {} }</pre>
LIST_NOTIF_INSTALL	<pre>request ListNotificationRequest ::= {   profileManagementOperation {     notificationInstall   } }</pre>
LIST_NOTIF_ENABLE	<pre>request ListNotificationRequest ::= {   profileManagementOperation {     notificationEnable   } }</pre>

	<pre> } } </pre>
LIST_NOTIF_DISABLE	<pre> request ListNotificationRequest ::= {   profileManagementOperation {     notificationDisable   } } </pre>
LIST_NOTIF_DELETE	<pre> request ListNotificationRequest ::= {   profileManagementOperation {     notificationDelete   } } </pre>
LIST_NOTIF_INSTALL_ENABLE	<pre> request ListNotificationRequest ::= {   profileManagementOperation {     notificationInstall,     notificationEnable   } } </pre>
LIST_NOTIF_DISABLE_DELETE	<pre> request ListNotificationRequest ::= {   profileManagementOperation {     notificationDisable,     notificationDelete   } } </pre>
LIST_NOTIF_DISABLE_ENABLE	<pre> request ListNotificationRequest ::= {   profileManagementOperation {     notificationDisable,     notificationEnable   } } </pre>
LIST_NOTIF_INSTALL_ENABLE_DISABLE	<pre> request ListNotificationRequest ::= {   profileManagementOperation {     notificationInstall,     notificationEnable,     notificationDisable   } } </pre>
LIST_NOTIF_ENABLE_DISABLE_DELETE	<pre> request ListNotificationRequest ::= {   profileManagementOperation {     notificationEnable,     notificationDisable,     notificationDelete   } } </pre>
METADATA_EN_DI_DE_NOTIFS	<pre> iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, notificationConfigurationInfo {   { profileManagementOperation { </pre>



	<pre> notificationEnable, notificationDisable, notificationDelete }, notificationAddress #TEST_DP_ADDRESS1 }, { profileManagementOperation { notificationEnable, notificationDisable, notificationDelete }, notificationAddress #TEST_DP_ADDRESS2 } } } </pre>
PREP_DOWNLOAD_INVALID_CC	<pre> req PrepareDownloadRequest ::= { smdpSigned2 { transactionId &lt;S_TRANSACTION_ID&gt;, ccRequiredFlag TRUE }, smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;, smdpCertificate #CERT_S_SM_DPpb_ECDSA } </pre>
RETRIEVE_NOTIF_ALL	<pre> request RetrieveNotificationsListRequest ::= { searchCriteria profileManagementOperation { notificationInstall, notificationEnable, notificationDisable, notificationDelete } } </pre>
RETRIEVE_NOTIF_OMITTED	<pre> request RetrieveNotificationsListRequest ::= { } </pre>
RETRIEVE_NOTIF_NONE	<pre> request RetrieveNotificationsListRequest ::= { searchCriteria profileManagementOperation {} } </pre>
RETRIEVE_NOTIF_INSTALL	<pre> request RetrieveNotificationsListRequest ::= { searchCriteria profileManagementOperation { notificationInstall } } </pre>
RETRIEVE_NOTIF_ENABLE	<pre> request RetrieveNotificationsListRequest ::= { searchCriteria profileManagementOperation { notificationEnable } } </pre>
RETRIEVE_NOTIF_DISABLE	<pre> request RetrieveNotificationsListRequest ::= { searchCriteria profileManagementOperation { notificationDisable } } </pre>

	} }
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>, }

	<pre>                 ccRequiredFlag FALSE             },             smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;,             smdpCertificate #CERT_S_SM_DPpb_INV_SIGN         }     </pre>
PREP_DOWNLOAD_INV_OID	<pre> req PrepareDownloadRequest ::= {     smdpSigned2 {         transactionId &lt;S_TRANSACTION_ID&gt;,         ccRequiredFlag FALSE     },     smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;,     smdpCertificate #CERT_S_SM_DPauth_ECDSA }     </pre>
PREP_DOWNLOAD_INV_SIGN	<pre> req PrepareDownloadRequest ::= {     smdpSigned2 {         transactionId &lt;S_TRANSACTION_ID&gt;,         ccRequiredFlag FALSE     },     smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;,     smdpCertificate #CERT_S_SM_DPpb_ECDSA }     </pre> <p><b>NOTE:</b> The &lt;S_SM_DP+_SIGNATURE2&gt; SHALL NOT be computed using the #SK_S_SM_DPpb_ECDSA but SHALL have the same length as for a valid signature.</p>
PREP_DOWNLOAD_INV_TRANS_ID	<pre> req PrepareDownloadRequest ::= {     smdpSigned2 {         transactionId &lt;INVALID_TRANSACTION_ID&gt;,         ccRequiredFlag FALSE     },     smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;,     smdpCertificate #CERT_S_SM_DPpb_ECDSA }     </pre>
PREP_DOWNLOAD_NO_AUTH	<pre> req PrepareDownloadRequest ::= {     smdpSigned2 {         transactionId &lt;S_TRANSACTION_ID&gt;,         ccRequiredFlag FALSE     },     smdpSignature2 &lt;RANDOM_SM_DP+_SIGN&gt;,     smdpCertificate #CERT_S_SM_DPpb_ECDSA }     </pre>
PREP_DOWNLOAD_NO_CC	<pre> req PrepareDownloadRequest ::= {     smdpSigned2 {         transactionId &lt;S_TRANSACTION_ID&gt;,         ccRequiredFlag FALSE     },     smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;,     smdpCertificate #CERT_S_SM_DPpb_ECDSA }     </pre>
PREP_DOWNLOAD_RETRY_CC	<pre> req PrepareDownloadRequest ::= {     smdpSigned2 {         transactionId &lt;S_TRANSACTION_ID&gt;,     </pre>

	<pre>                 ccRequiredFlag TRUE,                 bppEuiccOtpk &lt;OTPK_EUICC_ECKA&gt;             },             smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;,             hashCc &lt;S_HASHED_CC&gt;,             smdpCertificate #CERT_S_SM_DPpb_ECDSA         }     </pre>
PREP_DOWNLOAD_WITH_CC	<pre> req PrepareDownloadRequest ::= {     smdpSigned2 {         transactionId &lt;S_TRANSACTION_ID&gt;,         ccRequiredFlag TRUE     },     smdpSignature2 &lt;S_SM_DP+_SIGNATURE2&gt;,     hashCc &lt;S_HASHED_CC&gt;,     smdpCertificate #CERT_S_SM_DPpb_ECDSA }     </pre>
SET_EUICC_CONFIGURED_ADDRESS_1	<pre> request SetDefaultDpAddressRequest::={     defaultDpAddress #TEST_DP_ADDRESS1 }     </pre>
SET_EUICC_CONFIGURED_ADDRESS_2	<pre> request SetDefaultDpAddressRequest::={     defaultDpAddress #TEST_DP_ADDRESS2 }     </pre>
SET_EUICC_CONFIGURED_ADDRESS_EMPTY	<pre> request SetDefaultDpAddressRequest::={     defaultDpAddress "" }     </pre>
SET_NICKNAME_EMPTY_OP_PROF1	<pre> setNicknameReq SetNicknameRequest ::= {     iccid #ICCID_OP_PROF1,     profileNickname "" }     </pre>
SET_NICKNAME_ICCID_UNKNOWN	<pre> setNicknameReq SetNicknameRequest ::= {     iccid #ICCID_UNKNOWN,     profileNickname #NICKNAME2 }     </pre>
SET_NICKNAME_OP_PROF1	<pre> setNicknameReq SetNicknameRequest ::= {     iccid #ICCID_OP_PROF1,     profileNickname #NICKNAME2 }     </pre>

### D.3.2 ES10x Responses

Name	Content
NOTIF_METADATA_DELETE1 (NotificationMetadata)	<pre> {     seqNumber &lt;NOTIF_SEQ_NO_DE1&gt;,     profileManagementOperation { notificationDelete },     notificationAddress #TEST_DP_ADDRESS1,     iccid #ICCID_OP_PROF1 }     </pre>
NOTIF_METADATA2_DELETE1	<pre> {     seqNumber &lt;NOTIF_SEQ_NO2_DE1&gt;,     </pre>

(NotificationMetadata)	<pre>profileManagementOperation { notificationDelete }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF1 }</pre>
NOTIF_METADATA_DISABLE1 (NotificationMetadata)	<pre>{ seqNumber &lt;NOTIF_SEQ_NO_DI1&gt;, profileManagementOperation { notificationDisable }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }</pre>
NOTIF_METADATA2_DISABLE1 (NotificationMetadata)	<pre>{ seqNumber &lt;NOTIF_SEQ_NO2_DI1&gt;, profileManagementOperation { notificationDisable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF1 }</pre>
NOTIF_METADATA_ENABLE1 (NotificationMetadata)	<pre>{ seqNumber &lt;NOTIF_SEQ_NO_EN1&gt;, profileManagementOperation { notificationEnable }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }</pre>
NOTIF_METADATA2_ENABLE1 (NotificationMetadata)	<pre>{ seqNumber &lt;NOTIF_SEQ_NO2_EN1&gt;, profileManagementOperation { notificationEnable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF1 }</pre>
NOTIF_METADATA_ENABLE2 (NotificationMetadata)	<pre>{ seqNumber &lt;NOTIF_SEQ_NO_EN2&gt;, profileManagementOperation { notificationEnable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF2 }</pre>
NOTIF_METADATA_INSTALL1 (NotificationMetadata)	<pre>{ seqNumber &lt;NOTIF_SEQ_NO_IN1&gt;, profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }</pre>
NOTIF_METADATA_INSTALL1_PIR (NotificationMetadata)	<pre>{ seqNumber &lt;NOTIF_SEQ_NO_IN1_PIR&gt;, profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }</pre>

<p>NOTIF_METADATA_INSTALL2                  (NotificationMetadata)</p>	<pre>{   seqNumber &lt;NOTIF_SEQ_NO_IN2&gt;,   profileManagementOperation {     notificationInstall   },   notificationAddress #TEST_DP_ADDRESS2,   iccid #ICCID_OP_PROF2 }</pre>
<p>NOTIF_METADATA_INSTALL2_PIR                  (NotificationMetadata)</p>	<pre>{   seqNumber &lt;NOTIF_SEQ_NO_IN2_PIR&gt;,   profileManagementOperation {     notificationInstall   },   notificationAddress #TEST_DP_ADDRESS2,   iccid #ICCID_OP_PROF2 }</pre>
<p>PPR1_WITH_OWNER_GID                  (ProfilePolicyAuthorisationRule)</p>	<pre>{   pprIds { ppr1 },   allowedOperators {     { mccMnc #MCC_MNC2,       gid1 #GID1,       gid2 #GID2     }   },   pprFlags { consentRequired} }</pre>
<p>PPR1_WITHOUT_GID                  (ProfilePolicyAuthorisationRule)</p>	<pre>{   pprIds { ppr1 },   allowedOperators {     { mccMnc #MCC_MNC4 }   },   pprFlags { consentRequired} }</pre>
<p>PPR2_WITHOUT_CONSENT                  (ProfilePolicyAuthorisationRule)</p>	<pre>{   pprIds { ppr2 },   allowedOperators {     { mccMnc '92EEEE'H, gid1 ''H, gid2 ''H}   },   pprFlags { } }</pre>
<p>PPRS_ALLOWED                  (ProfilePolicyAuthorisationRule)</p>	<pre>{   pprIds { ppr1, ppr2 },   allowedOperators {     { mccMnc 'EEEEEE'H, gid1 ''H, gid2 ''H}   },   pprFlags { consentRequired} }</pre>
<p>PROFILE_INFO1                  (ProfileInfo)</p>	<pre>{   iccid #ICCID_OP_PROF1,   isdpAid &lt;ISD_P_AID1&gt;,   profileState enabled,   serviceProviderName #SP_NAME1, }</pre>

	<pre> profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational } </pre>
<p>PROFILE_INFO1_DISABLED          (ProfileInfo)</p>	<pre> {   iccid #ICCID_OP_PROF1,   isdpAid &lt;ISD_P_AID1&gt;,   profileState disabled,   serviceProviderName #SP_NAME1,   profileName #NAME_OP_PROF1,   iconType png,   icon #ICON_OP_PROF1,   profileClass operational } </pre>
<p>PROFILE_INFO2          (ProfileInfo)</p>	<pre> {   iccid #ICCID_OP_PROF2,   isdpAid &lt;ISD_P_AID2&gt;,   profileState disabled,   serviceProviderName #SP_NAME2,   profileName #NAME_OP_PROF2,   iconType png,   icon #ICON_OP_PROF2,   profileClass operational } </pre>
<p>PROFILE_INFO2_ENABLED          (ProfileInfo)</p>	<pre> {   iccid #ICCID_OP_PROF2,   isdpAid &lt;ISD_P_AID2&gt;,   profileState enabled,   serviceProviderName #SP_NAME2,   profileName #NAME_OP_PROF2,   iconType png,   icon #ICON_OP_PROF2,   profileClass operational } </pre>
<p>PROFILE_INFO3          (ProfileInfo)</p>	<pre> {   iccid #ICCID_OP_PROF3,   isdpAid &lt;ISD_P_AID3&gt;,   profileState disabled,   profileNickname #NICKNAME3,   serviceProviderName #SP_NAME3,   profileName #NAME_OP_PROF3,   iconType png,   icon #ICON_OP_PROF3,   profileClass operational } </pre>
<p>PROFILE_INFO4          (ProfileInfo)</p>	<pre> {   iccid #ICCID_OP_PROF4,   isdpAid &lt;ISD_P_AID4&gt;,   profileState disabled,   serviceProviderName #SP_NAME4,   profileName #NAME_OP_PROF4,   iconType png, } </pre>

	<pre>icon #ICON_OP_PROF4, profileClass operational }</pre>
PROFILE_INFO4_ENABLED (ProfileInfo)	<pre>{   iccid #ICCID_OP_PROF4,   isdpAid &lt;ISD_P_AID4&gt;,   profileState enabled,   serviceProviderName #SP_NAME4,   profileName #NAME_OP_PROF4,   iconType png,   icon #ICON_OP_PROF4,   profileClass operational }</pre>
PROFILES_INFO_ICCID_TAGLIST1 (ProfileInfo)	<pre>{profileState enabled}</pre>
PROFILES_INFO_ICCID_TAGLIST2 (ProfileInfo)	<pre>{iconType png}</pre>
PROFILES_INFO_ICCID_TAGLIST3 (ProfileInfo)	<pre>{profileClass operational }</pre>
PROFILES_INFO_ICCID_TAGLIST4 (ProfileInfo)	<pre>notificationConfigurationInfo from #METADATA_OP_PROF1</pre>
PROFILES_INFO_ICCID_TAGLIST5 (ProfileInfo)	<pre>profilePolicyRules from #METADATA_OP_PROF3</pre>
PROFILES_INFO_TAGLIST_ICCID (ProfileInfo)	<pre>{iccid #ICCID_OP_PROF1}, {iccid #ICCID_OP_PROF2}, {iccid #ICCID_OP_PROF3}</pre>
PROFILES_INFO_TAGLIST_ICON (ProfileInfo)	<pre>{icon #ICON_OP_PROF1}, {icon #ICON_OP_PROF2}, {icon #ICON_OP_PROF3}</pre>
PROFILES_INFO_TAGLIST_ISDPAID (ProfileInfo)	<pre>{isdpAid &lt;ISD_P_AID1&gt;}, {isdpAid &lt;ISD_P_AID2&gt;}, {isdpAid &lt;ISD_P_AID3&gt;}</pre>
PROFILES_INFO_TAGLIST_PROFILE _NAME (ProfileInfo)	<pre>{profileName #NAME_OP_PROF1}, {profileName #NAME_OP_PROF2}, {profileName #NAME_OP_PROF3}</pre>
PROFILES_INFO_TAGLIST_PROFILE _NICKNAME (ProfileInfo)	<pre>{profileNickname #NICKNAME3}</pre>
PROFILES_INFO_TAGLIST_PROFILE _OWNER (ProfileInfo)	<pre>{profileOwner #OWNER_OP_PROF1}, {profileOwner #OWNER_OP_PROF2}, {profileOwner #OWNER_OP_PROF2}</pre>
PROFILES_INFO_TAGLIST_SMDP_P ROP_DATA (ProfileInfo)	<pre>{dpProprietaryData #SMDP_PROP_DATA1}</pre>



PROFILES_INFO_TAGLIST_SP_NAME (ProfileInfo)	<pre>{serviceProviderName #SP_NAME1}, {serviceProviderName #SP_NAME2}, {serviceProviderName #SP_NAME3}</pre>
PROFILES_INFO_TAGLIST1 (ProfileInfo)	<pre>{   iccid #ICCID_OP_PROF1,   profileState enabled }, {   iccid #ICCID_OP_PROF2,   profileState disabled }, {   iccid #ICCID_OP_PROF3,   profileState disabled }</pre>
PROFILES_INFO_TAGLIST2 (ProfileInfo)	<pre>{   profileState enabled }, {   profileState disabled }, {   profileState disabled,   profileNickname #NICKNAME3 }</pre>
PROFILES_INFO_TAGLIST3 (ProfileInfo)	<pre>{   iconType png,   icon #ICON_OP_PROF1 }, {   iconType png,   icon #ICON_OP_PROF2 }, {   iconType png,   icon #ICON_OP_PROF3 }</pre>
PROFILES_INFO_TAGLIST4 (ProfileInfo)	<pre>{   profileState enabled,   icon #ICON_OP_PROF1 }, {   profileState disabled,   icon #ICON_OP_PROF2 }, {   profileState disabled,   icon #ICON_OP_PROF3 }</pre>
R_AUTH_SMDP_MATCH_ID	<pre>resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,</pre>

	<pre> serverAddress #TEST_DP_ADDRESS1, serverChallenge &lt;S_SMDP_CHALLENGE&gt;, euiiccInfo2 #R_EUICC_INFO2, -- check only that the field is present and has a valid TLV asn.1 structure     ctxParams1 #CTX_PARAMS1_MATCH_ID }, euiiccSignature1 &lt;EUICC_SIGNATURE1&gt;, euiiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA } </pre>
R_AUTH_SMDP_IMEI	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {     euiiccSigned1 {         transactionId &lt;S_TRANSACTION_ID&gt;,         serverAddress #TEST_DP_ADDRESS1,         serverChallenge &lt;S_SMDP_CHALLENGE&gt;,         euiiccInfo2 #R_EUICC_INFO2, -- check only that         the field is present and has a valid TLV asn.1         structure             ctxParams1 #CTX_PARAMS1_IMEI         },         euiiccSignature1 &lt;EUICC_SIGNATURE1&gt;,         euiiccCertificate #CERT_EUICC_ECDSA,         eumCertificate #CERT_EUM_ECDSA     } } </pre>
R_AUTH_SERVER_INV_CERT	<pre> resp AuthenticateServerResponse ::= authenticateResponseError : {     transactionId &lt;S_TRANSACTION_ID&gt;,     authenticateErrorCode invalidCertificate } </pre>
R_AUTH_SERVER_INV_SIGN	<pre> resp AuthenticateServerResponse ::= authenticateResponseError : {     transactionId &lt;S_TRANSACTION_ID&gt;,     authenticateErrorCode invalidSignature } </pre>
R_AUTH_SERVER_INV_CURV	<pre> resp AuthenticateServerResponse ::= authenticateResponseError : {     transactionId &lt;S_TRANSACTION_ID&gt;,     authenticateErrorCode unsupportedCurve } </pre>
R_AUTH_SERVER_INV_CHALLENGE	<pre> resp AuthenticateServerResponse ::= authenticateResponseError : {     transactionId &lt;S_TRANSACTION_ID&gt;,     authenticateErrorCode euiiccChallengeMismatch } </pre>
R_AUTH_SERVER_INV_CI	<pre> resp AuthenticateServerResponse ::= authenticateResponseError : {     transactionId &lt;S_TRANSACTION_ID&gt;,     authenticateErrorCode ciPKUnknown } </pre>

R_AUTH_SERVER_INV_OID	<pre> resp AuthenticateServerResponse ::= authenticateResponseError : {   transactionId &lt;S_TRANSACTION_ID&gt;,   authenticateErrorCode invalidOid } </pre>
R_AUTH_SERVER_NO_SESSION	<pre> resp AuthenticateServerResponse ::= authenticateResponseError : {   transactionId &lt;S_TRANSACTION_ID&gt;,   authenticateErrorCode noSessionContext } </pre>
R_AUTH_SMDS_IMEI	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     serverAddress #TEST_ROOT_DS_ADDRESS,     serverChallenge &lt;S_SMDS_CHALLENGE&gt;,     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 &lt;EUICC_SIGNATURE1&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } </pre>
R_AUTHENTICATE_SMDP	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk: {   euiccSigned1 #EUICC_SIGNED1,   euiccSignature1 &lt;EUICC_SIGNATURE1&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } </pre>
R_AUTHENTICATE_SMDS	<pre> resp AuthenticateServerResponse ::= authenticateResponseOk: {   euiccSigned1 {     transactionId &lt;S_TRANSACTION_ID&gt;,     serverAddress #TEST_ROOT_DS_ADDRESS,     serverChallenge &lt;S_SMDS_CHALLENGE&gt;,     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 &lt;EUICC_SIGNATURE1&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } </pre>
R_CANCEL_SESSION_INV_TRANS_I D	<pre> resp CancelSessionResponse ::= cancelSessionResponseError : invalidTransactionId </pre>
R_CANCEL_SESSION_METADATA	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned { </pre>

	<pre> transactionId &lt;S_TRANSACTION_ID&gt;, smdpOid #S_SM_DP+_OID, reason metadataMismatch }, euiccCancelSessionSignature &lt;EUICC_CS_SIGNATURE&gt; } </pre>
R_CANCEL_SESSION_REJ	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason endUserRejection   },   euiccCancelSessionSignature &lt;EUICC_CS_SIGNATURE&gt; } </pre>
R_CANCEL_SESSION_POSTPONED	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason postponed   },   euiccCancelSessionSignature &lt;EUICC_CS_SIGNATURE&gt; } </pre>
R_CANCEL_SESSION_TIMEOUT	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason timeout   },   euiccCancelSessionSignature &lt;EUICC_CS_SIGNATURE&gt; } </pre>
R_CANCEL_SESSION_PPR	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason pprNotAllowed   },   euiccCancelSessionSignature &lt;EUICC_CS_SIGNATURE&gt; } </pre>
R_CANCEL_SESSION_LOAD_BPP	<pre> resp CancelSessionResponse ::= cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason loadBppExecutionError   },   euiccCancelSessionSignature &lt;EUICC_CS_SIGNATURE&gt; } </pre>
R_CANCEL_SESSION_UNDEF	<pre> resp CancelSessionResponse ::= </pre>

	<pre>cancelSessionResponseOk : {   euiccCancelSessionSigned {     transactionId &lt;S_TRANSACTION_ID&gt;,     smdpOid #S_SM_DP+_OID,     reason undefinedReason   },   euiccCancelSessionSignature &lt;EUICC_CS_SIGNATURE&gt; }</pre>
R_CHALLENGE	<pre>response GetEuiccChallengeResponse ::= {   euiccChallenge &lt;EUICC_CHALLENGE&gt; }</pre>
R_CONF_OP_PROF1	<pre>resp ProfileInfoListResponse ::= profileInfoListOk :{   {     isdpAid &lt;ISD_P_AID&gt;,     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 }</pre>
R_DEFAULT_RAT	<pre>response GetRatResponse ::= {   rat {     #PPRS_ALLOWED   } }</pre>
R_DELETE_PROFILE_DISALLOWED BYPOLICY	<pre>respDelProf DeleteProfileResponse ::= {   deleteResult disallowedByPolicy }</pre>
R_DELETE_PROFILE_NOTDISABLE STATE	<pre>respDelProf DeleteProfileResponse ::= {   deleteResult profileNotInDisabledState }</pre>
R_DELETE_PROFILE_OK	<pre>respDelProf DeleteProfileResponse ::= {   deleteResult ok }</pre>

R_DELETE_PROFILE_ICCID_ISDP_N OTFOUND	resp DeleteProfileResponse ::= { deleteResult iccidOrAidNotFound }
R_DISABLE_PROFILE_DISALLOWED byPOLICY	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_N OTFOUND	respEnaPro EnableProfileResponse ::= { enableResult iccidOrAidNotFound }
R_ENABLE_PROFILE_NOT_DISABLE _STATE	respEnaPro EnableProfileResponse ::= { enableResult profileNotInDisabledState }
R_ENABLE_PROFILE_DISALLOWED byPOLICY	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 ::= {

	<pre> svn #RSP_SVN_H, -- for device testing, check only that the field is present and has a valid TLV asn.1 structure     euiccCiPKIdListForVerification         &lt;EUICC_CI_PK_ID_LIST_FOR_VERIFICATION&gt;,     euiccCiPKIdListForSigning         &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;     }         </pre>
<p>R_EUICC_INFO2</p>	<pre> response EUICCInfo2 ::= {     profileVersion #IUT_SIMA_VERSION,     svn #RSP_SVN_H,     euiccFirmwareVer #IUT_EUICC_FIRMWARE_VER,     extCardResource &lt;EXT_CARD_RESOURCE&gt;,     uiccCapability #IUT_UICC_CAPABILITY,     ts102241Version #IUT_TS102241_VERSION,     globalplatformVersion         #IUT_GLOBALPLATFORM_VERSION,     rspCapability &lt;EUICC_RSP_CAPABILITY&gt;,     euiccCiPKIdListForVerification         &lt;EUICC_CI_PK_ID_LIST_FOR_VERIFICATION&gt;,     euiccCiPKIdListForSigning         &lt;EUICC_CI_PK_ID_LIST_FOR_SIGNING&gt;,     euiccCategory #IUT_EUICC_CATEGORY, -- OPTIONAL     forbiddenProfilePolicyRules &lt;PPR_IDS&gt;, -- OPTIONAL     ppVersion #IUT_PP_VERSION,     sasAccreditationNumber         #IUT_SAS_ACCREDITATION_NUMBER,     certificationDataObject {         platformLabel #IUT_PLATFORM_LABEL,         discoveryBaseUrl #IUT_DLOA_URL     } -- OPTIONAL }         </pre>
<p>R_EUICC_MEMORY_RESET_OK</p>	<pre> resp EuiccMemoryResetResponse ::= {     resetResult ok }         </pre>
<p>R_GET_UPDATE_N1</p>	<pre> resp ProfileInfoListResponse ::= profileInfoListOk :{     {         serviceProviderName #SP_NAME1,         profileName #NAME_OP_PROF1,         iconType png,         icon #ICON_OP_PROF1,         profilePolicyRules { ppr2 }     } }         </pre>
<p>R_GET_UPDATE_N2</p>	<pre> resp ProfileInfoListResponse ::= profileInfoListOk :{     {         serviceProviderName #SP_NAME1,         profileName #NAME_OP_PROF1,         iconType jpg,         icon #ICON_JPG,         } }         </pre>

	<pre>         profilePolicyRules { ppr1 }     } } </pre>
R_GET_UPDATE_N3	<pre> resp ProfileInfoListResponse ::=   profileInfoListOk :{     {       serviceProviderName #SP_NAME2,       profileName #NAME_OP_PROF2,       iconType png,       icon #ICON_OP_PROF1       -- profilePolicyRules SHALL not be present     }   } </pre>
R_GET_UPDATE_N4	<pre> resp ProfileInfoListResponse ::=   profileInfoListOk :{     {       -- serviceProviderName SHALL not be present       -- profileName SHALL not be present       iconType png,       icon #ICON_OP_PROF1       -- profilePolicyRules SHALL not be present     }   } </pre>
R_GET_UPDATE_N6	<pre> resp ProfileInfoListResponse ::=   profileInfoListOk :{     {       serviceProviderName #SP_NAME2,       profileName #NAME_OP_PROF2,       iconType png,       icon #ICON_OP_PROF1       -- profilePolicyRules SHALL not be present     }   } </pre>
R_LIST_NOTIF_DI1_EN2	<pre> response ListNotificationResponse ::= notificationMetadataList : {   #NOTIF_METADATA_DISABLE1,   #NOTIF_METADATA_ENABLE2 } </pre>
R_METADATA_UNCHANGED	<pre> resp ProfileInfoListResponse ::=   profileInfoListOk :{     {       serviceProviderName #SP_NAME1,       profileName #NAME_OP_PROF1,       iconType png,       icon #ICON_OP_PROF1,       profilePolicyRules {ppr1,ppr2}     }   } </pre>
R_PIR_DATA_MISMATCH	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata { </pre>



	<pre> seqNumber &lt;SEQ_NUMBER&gt;, profileManagementOperation {   notificationInstall }, notificationAddress #TEST_DP_ADDRESS1, ... }, smdpOid #S_SM_DP+_OID, finalResult errorResult : {   bppCommandId loadProfileElements,   errorReason installFailedDueToDataMismatch,   ... } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_PIR_OK_PROF9	<pre> response ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #TEST_DP_ADDRESS1,       iccid #ICCID_OP_PROF9     },     smdpOid #S_SM_DP+_OID,     finalResult successResult : {       aid &lt;ISD_P_AID&gt;,       simaResponse #SIMA_RESULT_OK     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_PIR_PPR_NOT_ALLOWED	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #TEST_DP_ADDRESS1,       ...     },     smdpOid #S_SM_DP+_OID,     finalResult errorResult : {       bppCommandId storeMetadata,       errorReason pprNotAllowed     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_GET_METADATA_OP_PROF1	<pre> resp ProfileInfoListResponse ::=   profileInfoListOk :{ </pre>

	<pre> {     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} }     </pre>
<p>R_GET_PROF_NOTIF_CONF</p>	<pre> 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     }, }     </pre>

	<pre>         { profileManagementOperation {             notificationDelete         },         notificationAddress #TEST_DP_ADDRESS1     },     { profileManagementOperation {         notificationDelete     },     notificationAddress #TEST_DP_ADDRESS3     }     }     }     }     </pre>
R_ISDR_SELECTION	<pre> resp ISDRProprietaryApplicationTemplate ::= {     svn #RSP_SVN_H }     </pre>
R_LIST_NOTIF_DE1	<pre> response ListNotificationResponse ::= notificationMetadataList : {     #NOTIF_METADATA_DELETE1 }     </pre>
R_LIST_NOTIF_DE1_DE1	<pre> response ListNotificationResponse ::= notificationMetadataList : {     #NOTIF_METADATA_DELETE1,     #NOTIF_METADATA2_DELETE1 }     </pre>
R_LIST_NOTIF_DI1	<pre> response ListNotificationResponse ::= notificationMetadataList : {     #NOTIF_METADATA_DISABLE1 }     </pre>
R_LIST_NOTIF_DI1_DE1	<pre> response ListNotificationResponse ::= notificationMetadataList : {     #NOTIF_METADATA_DISABLE1,     #NOTIF_METADATA_DELETE1 }     </pre>
R_LIST_NOTIF_DI1_DI1	<pre> response ListNotificationResponse ::= notificationMetadataList : {     #NOTIF_METADATA_DISABLE1,     #NOTIF_METADATA2_DISABLE1 }     </pre>
R_LIST_NOTIF_EN1	<pre> response ListNotificationResponse ::= notificationMetadataList: {     #NOTIF_METADATA_ENABLE1 }     </pre>
R_LIST_NOTIF_EN1_EN1	<pre> response ListNotificationResponse ::= notificationMetadataList : {     #NOTIF_METADATA_ENABLE1,     #NOTIF_METADATA2_ENABLE1 }     </pre>
R_LIST_NOTIF_IN1	<pre> response ListNotificationResponse ::=     </pre>

	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 {

	<pre> transactionId &lt;S_TRANSACTION_ID&gt;, notificationMetadata {   seqNumber &lt;SEQ_NUMBER&gt;,   profileManagementOperation {     notificationInstall   },   notificationAddress #TEST_DP_ADDRESS1,   iccid #ICCID_OP_PROF1 }, smdpOid #S_SM_DP+_OID, finalResult errorResult : {   bppCommandId storeMetadata,   errorReason     installFailedDueToIccidAlreadyExistsOnEuicc } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_PIR_INVALID_CRT	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #TEST_DP_ADDRESS1     },     smdpOid #S_SM_DP+_OID,     finalResult errorResult : {       bppCommandId initialiseSecureChannel,       errorReason unsupportedCrtValues     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_PIR_INVALID_DATA	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #TEST_DP_ADDRESS1     },     smdpOid #S_SM_DP+_OID,     finalResult errorResult : {       bppCommandId configureISDP,       errorReason incorrectInputValues     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_PIR_INVALID_OP_ID	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData { </pre>

	<pre> transactionId &lt;S_TRANSACTION_ID&gt;, notificationMetadata {   seqNumber &lt;SEQ_NUMBER&gt;,   profileManagementOperation {     notificationInstall   },   notificationAddress #TEST_DP_ADDRESS1 }, smdpOid #S_SM_DP+_OID, finalResult errorResult : {   bppCommandId initialiseSecureChannel,   errorReason unsupportedRemoteOperationType } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>R_PIR_INVALID_SIGN</p>	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #TEST_DP_ADDRESS1     },     smdpOid #S_SM_DP+_OID,     finalResult errorResult : {       bppCommandId initialiseSecureChannel,       errorReason invalidSignature     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>R_PIR_INVALID_TRANS_ID</p>	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;INVALID_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #TEST_DP_ADDRESS1     },     smdpOid #S_SM_DP+_OID,     finalResult errorResult : {       bppCommandId initialiseSecureChannel,       errorReason invalidTransactionId     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
<p>R_PIR_METADATA_INVALID</p>	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;, </pre>

	<pre> profileManagementOperation {   notificationInstall }, notificationAddress #TEST_DP_ADDRESS1, ... }, smdpOid #S_SM_DP+_OID, finalResult errorResult : {   bppCommandId storeMetadata,   errorReason     scp03tStructureError     OR     incorrectInputValues } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
R_PIR_OK	<pre> response ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     notificationMetadata {       seqNumber &lt;SEQ_NUMBER&gt;,       profileManagementOperation {         notificationInstall       },       notificationAddress #TEST_DP_ADDRESS1,       iccid #ICCID_OP_PROF1     },     smdpOid #S_SM_DP+_OID,     finalResult successResult : {       aid &lt;ISD_P_AID&gt;,       simaResponse #SIMA_RESULT_OK     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
R_PIR_PPK_INV	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     ...     finalResult errorResult : {       bppCommandId replaceSessionKeys,       errorReason         incorrectInputValues         OR         scp03tStructureError         OR         scp03tSecurityError     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }         </pre>
R_PIR_SECU_INVALID	<pre> resp ProfileInstallationResult ::= {   profileInstallationResultData {     transactionId &lt;S_TRANSACTION_ID&gt;,     ...     smdpOid #S_SM_DP+_OID,         </pre>

	<pre> finalResult errorResult : {   bppCommandId loadProfileElements,   errorReason incorrectInputValues   OR   scp03tStructureError   OR   scp03tSecurityError   ... } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_PREP_DOWN_INV_CURVE	<pre> resp PrepareDownloadResponse ::= downloadResponseError : {   transactionId &lt;S_TRANSACTION_ID&gt;,   downloadErrorCode unsupportedCurve } </pre>
R_PREP_DOWN_INV_TRANS_ID	<pre> resp PrepareDownloadResponse ::= downloadResponseError : {   transactionId &lt;INVALID_TRANSACTION_ID&gt;,   downloadErrorCode invalidTransactionId } </pre>
R_PREP_DOWN_NO_SESSION	<pre> resp PrepareDownloadResponse ::= downloadResponseError : {   transactionId &lt;S_TRANSACTION_ID&gt;,   downloadErrorCode noSessionContext } </pre>
R_PREP_DOWNLOAD_INV_CERT	<pre> resp PrepareDownloadResponse ::= downloadResponseError : {   transactionId &lt;S_TRANSACTION_ID&gt;,   downloadErrorCode invalidCertificate } </pre>
R_PREP_DOWNLOAD_INV_SIGN	<pre> resp PrepareDownloadResponse ::= downloadResponseError : {   transactionId &lt;S_TRANSACTION_ID&gt;,   downloadErrorCode invalidSignature } </pre>
R_PREP_DOWNLOAD_NO_CC	<pre> resp PrepareDownloadResponse ::= downloadResponseOk : {   euiccSigned2 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccOtpk &lt;OTPK_EUICC_ECKA&gt;   },   euiccSignature2 &lt;EUICC_SIGNATURE2&gt; } </pre>
R_PREP_DOWNLOAD_WITH_CC	<pre> resp PrepareDownloadResponse ::= downloadResponseOk : {   euiccSigned2 {     transactionId &lt;S_TRANSACTION_ID&gt;,     euiccOtpk &lt;OTPK_EUICC_ECKA&gt;,     hashCc &lt;S_HASHED_CC&gt;   } } </pre>



	<pre> },   euiccSignature2 &lt;EUICC_SIGNATURE2&gt; }         </pre>
R_RAT_WITH_OTHER_RULES	<pre> response GetRatResponse ::= {   rat {     #PPR1_WITH_OWNER_GID,     #PPR1_WITHOUT_GID,     #PPR2_WITHOUT_CONSENT,     #PPRS_ALLOWED   } }         </pre>
R_REMOVE_NOTIF_NOTHING_TO_DELETE	<pre> response NotificationSentResponse ::= {   deleteNotificationStatus nothingToDelete }         </pre>
R_REMOVE_NOTIF_OK	<pre> response NotificationSentResponse ::= {   deleteNotificationStatus ok }         </pre>
R_RETRIEVE_NOTIF_IN1_IN1_PIR	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {   profileInstallationResult : {     profileInstallationResultData {       transactionId &lt;S_TRANSACTION_ID&gt;,       notificationMetadata       #NOTIF_METADATA_INSTALL1_PIR,       smdpOid #S_SM_DP+_OID,       finalResult successResult : {         aid &lt;ISD_P_AID&gt;,         simaResponse #SIMA_RESULT_OK       }     },     euiccSignPIR &lt;EUICC_SIGN_PIR&gt;   },   otherSignedNotification : {     tbsOtherNotification #NOTIF_METADATA_INSTALL1,     euiccNotificationSignature     &lt;TBS_EUICC_NOTIF_SIG&gt;,     euiccCertificate #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA   } }         </pre>
R_RETRIEVE_NOTIF_IN1_PIR	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {   profileInstallationResult : {     profileInstallationResultData {       transactionId &lt;S_TRANSACTION_ID&gt;,       notificationMetadata       #NOTIF_METADATA_INSTALL1_PIR,       smdpOid #S_SM_DP+_OID,       finalResult successResult : {         aid &lt;ISD_P_AID&gt;,         simaResponse #SIMA_RESULT_OK       }     }, }         </pre>

	<pre> euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_RETRIEVE_NOTIF_IN1	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {   otherSignedNotification : {     tbsOtherNotification #NOTIF_METADATA_INSTALL1,     euiccNotificationSignature   }   &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } </pre>
R_RETRIEVE_NOTIF_EN1	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {   otherSignedNotification : {     tbsOtherNotification #NOTIF_METADATA_ENABLE1,     euiccNotificationSignature   }   &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } </pre>
R_RETRIEVE_NOTIF_IN2_PIR	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {   profileInstallationResult : {     profileInstallationResultData {       transactionId &lt;S_TRANSACTION_ID&gt;,       notificationMetadata     }     #NOTIF_METADATA_INSTALL2_PIR,     smdpOid #S_SM_DP+_OID2,     finalResult successResult : {       aid &lt;ISD_P_AID&gt;,       simaResponse #SIMA_RESULT_OK     }   },   euiccSignPIR &lt;EUICC_SIGN_PIR&gt; } </pre>
R_RETRIEVE_NOTIF_DI1	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {   otherSignedNotification : {     tbsOtherNotification #NOTIF_METADATA_DISABLE1,     euiccNotificationSignature   }   &lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } </pre>
R_RETRIEVE_NOTIF_DE1	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {   otherSignedNotification : {     tbsOtherNotification #NOTIF_METADATA_DELETE1, </pre>

	<pre> euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA } </pre>
<p>R_RETRIEVE_NOTIF_NONE</p>	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {} </pre>
<p>R_RETRIEVE_NOTIF_IN1_PIR_EN1</p>	<pre> resp RetrieveNotificationsListResponse ::= notificationList : { profileInstallationResult : { profileInstallationResultData { transactionId &lt;S_TRANSACTION_ID&gt;, notificationMetadata #NOTIF_METADATA_INSTALL1_PIR, smdpOid #S_SM_DP+_OID, finalResult successResult : { aid &lt;ISD_P_AID&gt;, simaResponse #SIMA_RESULT_OK } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }, otherSignedNotification : { tbsOtherNotification #NOTIF_METADATA_ENABLE1, euiccNotificationSignature &lt;TBS_EUICC_NOTIF_SIG&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA } } </pre>
<p>R_RETRIEVE_NOTIF_IN1_PIR_IN2_PIR</p>	<pre> resp RetrieveNotificationsListResponse ::= notificationList : { profileInstallationResult : { profileInstallationResultData { transactionId &lt;S_TRANSACTION_ID&gt;, notificationMetadata #NOTIF_METADATA_INSTALL1_PIR, smdpOid #S_SM_DP+_OID, finalResult successResult : { aid &lt;ISD_P_AID&gt;, simaResponse #SIMA_RESULT_OK } }, euiccSignPIR &lt;EUICC_SIGN_PIR&gt; }, profileInstallationResult : { profileInstallationResultData { transactionId &lt;S_TRANSACTION_ID&gt;, notificationMetadata #NOTIF_METADATA_INSTALL2_PIR, smdpOid #S_SM_DP+_OID2, finalResult successResult : { aid &lt;ISD_P_AID&gt;, simaResponse #SIMA_RESULT_OK } } } </pre>

	<pre>         },         euiccSignPIR &lt;EUICC_SIGN_PIR&gt;     } }     </pre>
<p>R_RETRIEVE_NOTIF_DI1_DE1</p>	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {     otherSignedNotification : {         tbsOtherNotification #NOTIF_METADATA_DISABLE1,         euiccNotificationSignature         &lt;TBS_EUICC_NOTIF_SIG&gt;,         euiccCertificate #CERT_EUICC_ECDSA,         eumCertificate #CERT_EUM_ECDSA     },     otherSignedNotification : {         tbsOtherNotification #NOTIF_METADATA_DELETE1,         euiccNotificationSignature         &lt;TBS_EUICC_NOTIF_SIG&gt;,         euiccCertificate #CERT_EUICC_ECDSA,         eumCertificate #CERT_EUM_ECDSA     } }     </pre>
<p>R_RETRIEVE_NOTIF_IN1_EN1</p>	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {     otherSignedNotification : {         tbsOtherNotification #NOTIF_METADATA_INSTALL1,         euiccNotificationSignature         &lt;TBS_EUICC_NOTIF_SIG&gt;,         euiccCertificate #CERT_EUICC_ECDSA,         eumCertificate #CERT_EUM_ECDSA     },     otherSignedNotification : {         tbsOtherNotification #NOTIF_METADATA_ENABLE1,         euiccNotificationSignature         &lt;TBS_EUICC_NOTIF_SIG&gt;,         euiccCertificate #CERT_EUICC_ECDSA,         eumCertificate #CERT_EUM_ECDSA     } }     </pre>
<p>R_RETRIEVE_NOTIF_EN1_IN2_PIR</p>	<pre> resp RetrieveNotificationsListResponse ::= notificationList : {     profileInstallationResult : {         profileInstallationResultData {             transactionId &lt;S_TRANSACTION_ID&gt;,             notificationMetadata             #NOTIF_METADATA_INSTALL2_PIR,             smdpOid #S_SM_DP+_OID2,             finalResult successResult : {                 aid &lt;ISD_P_AID&gt;,                 simaResponse #SIMA_RESULT_OK             }         },         euiccSignPIR &lt;EUICC_SIGN_PIR&gt;     },     otherSignedNotification : {         tbsOtherNotification#NOTIF_METADATA_ENABLE1,         euiccNotificationSignature     } }     </pre>

	<pre>&lt;TBS_EUICC_NOTIF_SIG&gt;,   euiccCertificate #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA } }</pre>
SMDP_PROP_DATA1 (DpProprietaryData)	<pre>{   dpOid #S_SM_DP+_OID }</pre>

## D.4 APDU

### D.4.1 APDU Commands

Name	Content
DELETE_SSD	<ul style="list-style-type: none"> <li>- CLA = 80, INS = E4, P1 = 00, P2 = 80, LC = &lt;L&gt;</li> <li>- Data = 4F &lt;L&gt; #SSD_AID</li> <li>- LE = 00</li> </ul>
GET_RESPONSE	<ul style="list-style-type: none"> <li>- CLA = 0x (x = &lt;CHANNEL_NUMBER&gt;), INS = C0,</li> <li>- P1 = 00, P2 = 00, LE = &lt;L&gt;</li> </ul>
GET_MNO_SD	<ul style="list-style-type: none"> <li>- CLA = 80, INS = F2, P1 = 80, P2 = 02, LC = &lt;L&gt;</li> <li>- Data = 4F 00</li> <li>- LE = 00</li> </ul>
INSTALL_PERSO_RES_ISDP	<ul style="list-style-type: none"> <li>- CLA = 80, INS = E6, P1 = 20, P2 = 00, LC = 16</li> <li>- Data = 00 00 10 A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 0F 00 00 00 00</li> <li>- LE = 00</li> </ul>
MANAGE_CHANNEL_OPEN	<ul style="list-style-type: none"> <li>- CLA = 00, INS = 70, P1 = 00, P2 = 00, LE = 01</li> </ul>
READ_BINARY	<ul style="list-style-type: none"> <li>- CLA = 00, INS = B0, P1 = 00, P2 = 00, LE = &lt;L&gt;</li> </ul>
SELECT_MF	<ul style="list-style-type: none"> <li>- CLA = 00, INS = A4, P1 = 00, P2 = 04, LC = &lt;L&gt;</li> <li>- Data = 3F 00</li> <li>- LE = 00</li> </ul>
SELECT_ICCID	<ul style="list-style-type: none"> <li>- CLA = 00, INS = A4, P1 = 00, P2 = 0C, LC = 02</li> <li>- Data = 2F E2</li> </ul>
SELECT_USIM	<ul style="list-style-type: none"> <li>- CLA = 00, INS = A4, P1 = 04, P2 = 04, LC = &lt;L&gt;</li> <li>- Data = #USIM_AID</li> <li>- LE = 00</li> </ul>
TERMINAL_CAPABILITY_LPAd	<ul style="list-style-type: none"> <li>- CLA = 80, INS = AA, P1 = 00, P2 = 00, LC = &lt;L&gt;</li> <li>- Data = A9 05 81 00 83 01 07</li> </ul>
TERMINAL_PROFILE	<ul style="list-style-type: none"> <li>- CLA = 80, INS = 10, P1 = 00, P2 = 00, LC = &lt;L&gt;</li> <li>- 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 02 00</li> </ul>
TERMINAL_PROFILE_eUICCProfileStateChanged	<ul style="list-style-type: none"> <li>- CLA = 80, INS = 10, P1 = 00, P2 = 00, LC = &lt;L&gt;</li> <li>- Data = FF FF FF FF FF FF 1F FF FF 03 02 FF FF 9F</li> </ul>

	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 "", }

	<pre>profilePolicyRules {} }</pre>
UPD_PPR_CONTROL	<pre>metadataReq UpdateMetadataRequest ::= {   serviceProviderName #SP_NAME2,   profileName #NAME_OP_PROF2,   iconType jpg,   icon #ICON_JPG,   profilePolicyRules {pprUpdateControl, ppr1} }</pre>
UPD_NO_METADATA	<pre>metadataReq UpdateMetadataRequest ::= { }</pre>
UPD_ICON_NO_TYPE	<pre>metadataReq UpdateMetadataRequest ::= {   serviceProviderName #SP_NAME2,   profileName #NAME_OP_PROF2,   icon #ICON_JPG,   profilePolicyRules {} }</pre>
UPD_ICON_TYPE_ONLY	<pre>metadataReq UpdateMetadataRequest ::= {   serviceProviderName #SP_NAME2,   profileName #NAME_OP_PROF2,   iconType jpg,   profilePolicyRules {} }</pre>

## D.6 ES11 Requests And Responses

### D.6.1 ES11 Requests

Name	Content
AUTH_SERVER_RESP_MATCHING_ID_EMPTY	<pre>resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {     transactionId     &lt;S_TRANSACTION_ID&gt;,     serverAddress     #IUT_SM_DS_ADDRESS,     serverChallenge     &lt;SMDS_CHALLENGE&gt;,     euiccInfo2 #S_EUICC_INFO2,     ctxParams1     #CTX_PARAMS1_MATCHING_ID_EMPTY   },   euiccSignature1   &lt;EUICC_SIGNATURE1&gt;,   euiccCertificate   #CERT_EUICC_ECDSA,   eumCertificate #CERT_EUM_ECDSA }</pre>
AUTH_SERVER_RESP_MATCHING_ID_EVENT_ID	<pre>resp authenticateServerResponse ::= authenticateResponseOk : {   euiccSigned1 {     transactionId     &lt;S_TRANSACTION_ID&gt;,</pre>

	<pre> serverAddress #IUT_SM_DS_ADDRESS, serverChallenge &lt;SMDS_CHALLENGE&gt;, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EVENT_ID }, euiccSignature1 &lt;EUICC_SIGNATURE1&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA } </pre>
<p>AUTH_SERVER_RESP_MATCHING_ID_OMITTED</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_OMITTED     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA } </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_BC_cA</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_EX_BC_cA } </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_BC_PLC</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 { </pre>



	<pre> transactionId &lt;S_TRANSACTION_ID&gt;, serverAddress #IUT_SM_DS_ADDRESS, serverChallenge &lt;SMDS_CHALLENGE&gt;, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 &lt;EUICC_SIGNATURE1&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_BC_PLC } </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_CP</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_EX_CP } </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_KU</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_EX_KU } </pre>

	<pre> }     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_2_6_1_SIG</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_INVALID_SIG }     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_2_6_3</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate     #CERT_EUM_ECDSA_EXPIRED }     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_3_6_1_EX_CP</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     </pre>

	<pre> euiCCCertificate #CERT_EUICC_ECDSA_INVALID_EX_CP, eumCertificate #CERT_EUM_ECDSA     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_3_6_1_EX_KU</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiCCSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiCCInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiCCSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiCCCertificate     #CERT_EUICC_ECDSA_INVALID_EX_KU,     eumCertificate #CERT_EUM_ECDSA }     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_3_6_1_SIG</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiCCSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiCCInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiCCSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiCCCertificate     #CERT_EUICC_ECDSA_INVALID_SIG,     eumCertificate #CERT_EUM_ECDSA }     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_3_6_1_SUB_ORG</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiCCSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiCCInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiCCSignature1 }     </pre>

	<pre> &lt;EUICC_SIGNATURE1&gt;, euiCCCertificate #CERT_EUICC_ECDSA_INVALID_SUB_ORG, eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_3_6_1_SUB_SN</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiCCSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiCCInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiCCSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiCCCertificate     #CERT_EUICC_ECDSA_INVALID_SUB_SN,     eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_3_6_3</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiCCSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiCCInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiCCSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiCCCertificate     #CERT_EUICC_ECDSA_EXPIRED,     eumCertificate #CERT_EUM_ECDSA }         </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_6_1_CHA</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiCCSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE_INVALID&gt;,         euiCCInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     }, }         </pre>

	<pre> euiccSignature1 &lt;EUICC_SIGNATURE1&gt;, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_1_6_1_SIG</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;S_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1_INVALID&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>AUTH_SERVER_RESP_SMDS_8_10_1_3_9</p>	<pre> resp authenticateServerResponse ::= authenticateResponseOk : {     euiccSigned1 {         transactionId         &lt;INVALID_TRANSACTION_ID&gt;,         serverAddress         #IUT_SM_DS_ADDRESS,         serverChallenge         &lt;SMDS_CHALLENGE&gt;,         euiccInfo2 #S_EUICC_INFO2,         ctxParams1         #CTX_PARAMS1_MATCHING_ID_EMPTY     },     euiccSignature1     &lt;EUICC_SIGNATURE1&gt;,     euiccCertificate     #CERT_EUICC_ECDSA,     eumCertificate #CERT_EUM_ECDSA }                     </pre>
<p>CTX_PARAMS1_MATCHING_ID_EVENT_ID (CtxParams1)</p>	<pre> ctxParamsForCommonAuthentication : {     matchingId #EVENT_ID_1,     deviceInfo #S_DEVICE_INFO }                     </pre>
<p>CTX_PARAMS1_MATCHING_ID_OMITTED (CtxParams1)</p>	<pre> ctxParamsForCommonAuthentication : {     deviceInfo #S_DEVICE_INFO }                     </pre>

## D.6.2 ES11 Responses

Name	Content
AUTH_CLIENT_DS_OK	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "eventEntries" : #EVENT_ENTRY }</pre>
AUTH_CLIENT_DS_OK1	<pre>{   "header" :{     "functionExecutionStatus":{       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "eventEntries" : [ #EVENT_ENTRY_1 ] }</pre>
AUTH_CLIENT_DS_OK2	<pre>{   "header" :{     "functionExecutionStatus":{       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "eventEntries" : [ #EVENT_ENTRY_2 ] }</pre>
AUTH_CLIENT_DS_OK_DSADDR1	<pre>{   "header" :{     "functionExecutionStatus":{       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "eventEntries" : [ #EVENT_ENTRY_DSADDR1 ] }</pre>
EVENT_ENTRY	<pre>{   "eventId" : &lt;EVENT_ID&gt;,   "rspServerAddress" : &lt;RSP_SERVER_ADDRESS&gt; }</pre>
EVENT_ENTRY_1	<pre>{   "eventId" : #EVENT_ID_1,   "rspServerAddress" : #TEST_DP_ADDRESS1 }</pre>
EVENT_ENTRY_2	<pre>{   "eventId" : #EVENT_ID_2,   "rspServerAddress" : #TEST_DP_ADDRESS1 }</pre>
EVENT_ENTRY_DSADDR1	<pre>{   "eventId" : #EVENT ID 1,</pre>

	<pre>"rspServerAddress" : #TEST_DS_ADDRESS1 }</pre>
EVENT_ENTRY_MULTI	<pre>{   "eventId" : #EVENT_ID_1,   "rspServerAddress" : #TEST_DP_ADDRESS1 }, {   "eventId" : #EVENT_ID_2,   "rspServerAddress" : #TEST_DP_ADDRESS2 }</pre>
R_AUTH_CLIENT_DS_EVENT_ENTRY_1_OK	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "eventEntries" : [#EVENT_ENTRY_1] }</pre>
R_AUTH_CLIENT_DS_EVENT_ENTRY_EMPTY_OK	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "eventEntries" : [] }</pre>
R_AUTH_CLIENT_DS_EVENT_ENTRY_MULTI_OK	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Executed-Success"     }   },   "transactionId" : &lt;S_TRANSACTION_ID&gt;,   "eventEntries" : [#EVENT_ENTRY_MULTI] }</pre>

### 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	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "1.2",         "reasonCode" : "4.2"       }     }   } }</pre>

	<pre>         }     }     </pre>
R_ERROR_8_1_1_3_8	<pre>     {       "header" : {         "functionExecutionStatus" : {           "status" : "Failed",           "statusCodeData" : {             "subjectCode" : "8.1.1",             "reasonCode" : "3.8"           }         }       }     }     </pre>
R_ERROR_8_1_2_6_1	<pre>     {       "header" : {         "functionExecutionStatus" : {           "status" : "Failed",           "statusCodeData" : {             "subjectCode" : "8.1.2",             "reasonCode" : "6.1"           }         }       }     }     </pre>
R_ERROR_8_1_2_6_3	<pre>     {       "header" : {         "functionExecutionStatus" : {           "status" : "Failed",           "statusCodeData" : {             "subjectCode" : "8.1.2",             "reasonCode" : "6.3"           }         }       }     }     </pre>
R_ERROR_8_1_3_6_1	<pre>     {       "header" : {         "functionExecutionStatus" : {           "status" : "Failed",           "statusCodeData" : {             "subjectCode" : "8.1.3",             "reasonCode" : "6.1"           }         }       }     }     </pre>
R_ERROR_8_1_3_6_3	<pre>     {       "header" : {         "functionExecutionStatus" : {           "status" : "Failed",           "statusCodeData" : {             "subjectCode" : "8.1.3",             "reasonCode" : "6.3"           }         }       }     }     </pre>



	<pre>         }       }     }   } </pre>
R_ERROR_8_1_4_8	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.1",         "reasonCode" : "4.8"       }     }   } } </pre>
R_ERROR_8_1_6_1	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.1",         "reasonCode" : "6.1"       }     }   } } </pre>
R_ERROR_8_2_1_2	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.2",         "reasonCode" : "1.2"       }     }   } } </pre>
R_ERROR_8_2_3_7	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.2",         "reasonCode" : "3.7"       }     }   } } </pre>
R_ERROR_8_2_5_4_3	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : { </pre>

	<pre>                 "subjectCode" : "8.2.5",                 "reasonCode" : "4.3"             }         }     } </pre>
R_ERROR_8_2_6_3_8	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.2.6",         "reasonCode" : "3.8"       }     }   } } </pre>
R_ERROR_8_2_7_2_2	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.2.7",         "reasonCode" : "2.2"       }     }   } } </pre>
R_ERROR_8_2_7_3_8	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.2.7",         "reasonCode" : "3.8"       }     }   } } </pre>
R_ERROR_8_2_7_6_4	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.2.7",         "reasonCode" : "6.4"       }     }   } } </pre>
R_ERROR_8_8_1_3_8	<pre> {   "header" : {     "functionExecutionStatus" : { </pre>

	<pre>                 "status" : "Failed",                 "statusCodeData" : {                     "subjectCode" : "8.8.1",                     "reasonCode" : "3.8"                 }             }         }     } </pre>
R_ERROR_8_8_2_3_1	<pre> {     "header" : {         "functionExecutionStatus" : {             "status" : "Failed",             "statusCodeData" : {                 "subjectCode" : "8.8.2",                 "reasonCode" : "3.1"             }         }     } } </pre>
R_ERROR_8_8_3_3_1	<pre> {     "header" : {         "functionExecutionStatus" : {             "status" : "Failed",             "statusCodeData" : {                 "subjectCode" : "8.8.3",                 "reasonCode" : "3.1"             }         }     } } </pre>
R_ERROR_8_8_3_10	<pre> {     "header" : {         "functionExecutionStatus" : {             "status" : "Failed",             "statusCodeData" : {                 "subjectCode" : "8.8",                 "reasonCode" : "3.10"             }         }     } } </pre>
R_ERROR_8_8_4_3_7	<pre> {     "header" : {         "functionExecutionStatus" : {             "status" : "Failed",             "statusCodeData" : {                 "subjectCode" : "8.8.4",                 "reasonCode" : "3.7"             }         }     } } </pre>
R_ERROR_8_8_5_4_10	<pre> { </pre>

	<pre> "header" : {   "functionExecutionStatus" : {     "status" : "Failed",     "statusCodeData" : {       "subjectCode" : "8.8.5",       "reasonCode" : "4.10"     }   } } </pre>
R_ERROR_8_8_5_6_4	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.8.5",         "reasonCode" : "6.4"       }     }   } } </pre>
R_ERROR_8_9_1_3_8	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9.1",         "reasonCode" : "3.8"       }     }   } } </pre>
R_ERROR_8_9_2_3_1	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9.2",         "reasonCode" : "3.1"       }     }   } } </pre>
R_ERROR_8_9_3_3_1	<pre> {   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9.3",         "reasonCode" : "3.1"       }     }   } } </pre>

<p>R_ERROR_8_9_4_2</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9",         "reasonCode" : "4.2"       }     }   } }</pre>
<p>R_ERROR_8_9_4_3_7</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9.4",         "reasonCode" : "3.7"       }     }   } }</pre>
<p>R_ERROR_8_9_5_1</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9",         "reasonCode" : "5.1"       }     }   } }</pre>
<p>R_ERROR_8_9_5_3_3</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9.5",         "reasonCode" : "3.3"       }     }   } }</pre>
<p>R_ERROR_8_9_5_3_9</p>	<pre>{   "header" : {     "functionExecutionStatus" : {       "status" : "Failed",       "statusCodeData" : {         "subjectCode" : "8.9.5",         "reasonCode" : "3.9"       }     }   } }</pre>

	} }
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_SMXX_1_3_8	The error response will be as follows dependent on the entity under test: <ul style="list-style-type: none"> <li>• for SM-DP+ testing on ES9+ SHALL be #R_ERROR_8_8_1_3_8</li> <li>• for SM-DS testing on ES11 SHALL be #R_ERROR_8_9_1_3_8</li> </ul>
R_ERROR_SMXX_2_3_1	The error response will be as follows dependent on the entity under test: <ul style="list-style-type: none"> <li>• for SM-DP+ testing on ES9+ SHALL be #R_ERROR_8_8_2_3_1</li> <li>• for SM-DS testing on ES11 SHALL be #R_ERROR_8_9_2_3_1</li> </ul>
R_ERROR_SMXX_3_3_1	The error response will be as follows dependent on the entity under test: <ul style="list-style-type: none"> <li>• for SM-DP+ testing on ES9+ SHALL be #R_ERROR_8_8_3_3_1</li> <li>• for SM-DS testing on ES11 SHALL be #R_ERROR_8_9_3_3_1</li> </ul>
R_ERROR_SMXX_4_3_7	The error response will be as follows dependent on the entity under test: <ul style="list-style-type: none"> <li>• for SM-DP+ testing on ES9+ SHALL be #R_ERROR_8_8_4_3_7</li> <li>• for SM-DS testing on ES11 SHALL be #R_ERROR_8_9_4_3_7</li> </ul>
R_SUCCESS	{

	<pre>"header" : {   "functionExecutionStatus" : {     "status" : "Executed-Success"   } }</pre>
--	---

## Annex E Profiles

Profile	PROFILE_OPERATIONAL1
Description	<p>Generic Operational Profile</p> <p>This Profile acts as an Operational Profile in the scope of this specification.</p> <p>NOTE: Milenage algorithm is used in this Profile</p>
Details	<p>The Profile Metadata SHALL be set to #METADATA_OP_PROF1, except if defined differently in the test sequence.</p> <p>The Unprotected Profile Package content SHALL follow the ASN.1 structures specified in Annex B.7 of SGP.11 [21] except that:</p> <ul style="list-style-type: none"> <li>• the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF1 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>• the <i>connectivityParameters</i> field SHALL not be present in the <i>ProfileHeader</i> element</li> <li>• the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF1</li> <li>• the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF1</li> <li>• the pinAttributes of pinApp1 present in the PE_PIN SHALL be set to 6</li> <li>• the SCP80 encryption key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO_SCP80_ENC_KEY</li> <li>• the SCP80 message authentication key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO_SCP80_AUTH_KEY</li> <li>• 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</li> <li>• the instance AID configured in the PE-SecurityDomain that corresponds to the Supplementary Security Domain PE_SSD SHALL be set to #SSD_AID</li> <li>• the ef-dir present in the PE-MF SHALL be configured with the AID #USIM_AID</li> <li>• the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available)</li> <li>• the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>• the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>• the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul> <p>The PROFILE_OPERATIONAL1 UPP is named #UPP_OP_PROF1 in the scope of this document.</p>

Profile	PROFILE_OPERATIONAL2
Description	<p>Generic Operational Profile</p> <p>This Profile acts as an Operational Profile in the scope of this specification.</p> <p>NOTE: Milenage algorithm is used in this Profile</p>
Details	<p>The Profile Metadata SHALL be set to #METADATA_OP_PROF2, except if defined differently in the test sequence.</p> <p>The Unprotected Profile Package content SHALL follow the ASN.1 structures specified in Annex B.7 of SGP.11 [21] except that:</p> <ul style="list-style-type: none"> <li>• the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF2 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>• the <i>connectivityParameters</i> field SHALL not be present in the ProfileHeader element</li> <li>• the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF2</li> <li>• the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF2</li> <li>• The pinAttributes of pinApp1 present in the PE_PIN SHALL be set to 6</li> <li>• the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available)</li> </ul>



	available) <ul style="list-style-type: none"> <li>• the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>• the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>• the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul> The PROFILE_OPERATIONAL2 UPP is named #UPP_OP_PROF2 in the scope of this document.
--	---

<b>Profile</b>	PROFILE_OPERATIONAL3
<b>Description</b>	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
<b>Details</b>	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 structures specified in Annex B.7 of SGP.11 [21] except that: <ul style="list-style-type: none"> <li>• the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF3 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>• the <i>connectivityParameters</i> field SHALL not be present in the <i>ProfileHeader</i> element</li> <li>• the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF3</li> <li>• the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF3</li> <li>• the pinAttributes of pinApp1 present in the PE_PIN SHALL be set to 6</li> <li>• the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available)</li> <li>• the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>• the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>• the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul> The PROFILE_OPERATIONAL3 UPP is named #UPP_OP_PROF3 in the scope of this document.

<b>Profile</b>	PROFILE_OPERATIONAL4
<b>Description</b>	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
<b>Details</b>	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 structures specified in Annex B.7 of SGP.11 [21] except that: <ul style="list-style-type: none"> <li>• the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF4 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>• the <i>connectivityParameters</i> field SHALL not be present in the <i>ProfileHeader</i> element</li> <li>• the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF4</li> <li>• the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF4</li> <li>• the pinAttributes of pinApp1 present in the PE_PIN SHALL be set to 6</li> <li>• the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not</li> </ul>

	available) <ul style="list-style-type: none"> <li>• the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>• the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>• the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul> The PROFILE_OPERATIONAL4 UPP is named #UPP_OP_PROF4 in the scope of this document.
--	---

<b>Profile</b>	PROFILE_OPERATIONAL5
<b>Description</b>	Generic 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
<b>Details</b>	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 structures specified in Annex B.7 of SGP.11 [21] except that: <ul style="list-style-type: none"> <li>• the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF5 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>• the <i>connectivityParameters</i> field SHALL not be present in the <i>ProfileHeader</i> element</li> <li>• the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF5</li> <li>• the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF5</li> <li>• the pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO1_PIN1</li> <li>• the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available)</li> <li>• the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>• the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>• the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul>

<b>Profile</b>	PROFILE_OPERATIONAL6
<b>Description</b>	Generic 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
<b>Details</b>	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 structures specified in Annex B.7 of SGP.11 [21] except that: <ul style="list-style-type: none"> <li>• the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF6 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>• the <i>connectivityParameters</i> field SHALL not be present in the <i>ProfileHeader</i> element</li> <li>• the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF6</li> <li>• the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF6</li> <li>• The pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO2_PIN1</li> <li>• the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available)</li> <li>• the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>• the Token Verification and the Receipt Generation keys SHALL not be set in the PE-</li> </ul>

	MNO-SD <ul style="list-style-type: none"> <li>the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul>
--	---

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 structures specified in Annex B.7 of SGP.11 [21] except that: <ul style="list-style-type: none"> <li>the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF7 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>the <i>connectivityParameters</i> field SHALL not be present in the <i>ProfileHeader</i> element</li> <li>the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF7</li> <li>the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF7</li> <li>the pinAttributes of pinAppl1 present in the PE_PIN SHALL be set to 6</li> <li>the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available)</li> <li>the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul>

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 structures specified in Annex B.7 of SGP.11 [21] except that: <ul style="list-style-type: none"> <li>the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF8 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>the <i>connectivityParameters</i> field SHALL not be present in the <i>ProfileHeader</i> element</li> <li>the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF8</li> <li>the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF8</li> <li>The pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO2_PIN1</li> <li>the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available)</li> <li>the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> </ul>

<b>Profile</b>	<b>PROFILE_OPERATIONAL9</b>
<b>Description</b>	<p>Generic 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</p>
<b>Details</b>	<p>The Profile Metadata SHALL be set to #METADATA_OP_PROF9, except if defined differently in the test sequence.</p> <p>The Unprotected Profile Package content SHALL follow the ASN.1 structures specified in Annex B.7 of SGP.11 [21] except that:</p> <ul style="list-style-type: none"> <li>• the <i>iccid</i> field SHALL be set to #ICCID_OP_PROF9 in the <i>ProfileHeader</i> element, in non-swapped format</li> <li>• the <i>connectivityParameters</i> field SHALL not be present in the ProfileHeader element</li> <li>• the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF9</li> <li>• the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF9</li> <li>• the pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO1_PIN1</li> <li>• the ef-ust SHALL be set to #EF_UST2 (service 17 and 18 are available)</li> <li>• the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H</li> <li>• the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD</li> <li>• the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H</li> <li>• the following new Profile Element PE_OPT_USIM SHALL be inserted right after PE_USIM:</li> </ul> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center; background-color: #c00000; color: white; margin: 0;"><b>PE_OPT_USIM</b></p> <pre> 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   } }                 </pre> </div> <p><b>NOTE :</b> The following OIDs are used:</p> <pre> id-OPT-USIM OBJECT IDENTIFIER ::= {joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) opt-usim(5)}                 </pre>
	<p>The PROFILE_OPERATIONAL9 UPP is named #UPP_OP_PROF9 in the scope of this document.</p>

## 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_CATEGORY	The category, if provided, SHALL be either not present or: <ul style="list-style-type: none"> <li>• other(0)</li> <li>• or basicEuicc(1)</li> <li>• or mediumEuicc(2)</li> <li>• or contactlessEuicc(3)</li> </ul>
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_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 SIMalliance PEDefinitions, coded as ASN.1 BIT STRING (19 bits).
IUT_SIMA_VERSION	Version of SIMalliance [4] supported by the eUICC (3 bytes representing major/minor/revision) e.g. 0x020100

### 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_CLIENT_TLS_VER	Highest TLS protocol version supported by the Client (SM-DP+ or SM-DS) under test, which SHALL be at least v1.2. For versions higher than TLS v1.2 backwards compatibility is assumed.
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.
<b>SM-DS Setting name</b>	<b>Description</b>
IUT_SM_DS_ADDRESS	FQDN of the SM-DS Under Test.
IUT_SM_DS_TLS_TIMEOUT	Timeout in seconds for SM-DS to wait for TLS Server Hello message which starts immediately after the SM-DS has sent the Client Hello message.

### 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_AuthenticatedConfirmation	Description of the way to perform Authenticated Confirmation
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}.

Device Setting name	Description
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 be either 2.1.0 or 2.2.x (x≥0).

## 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 LPA 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 LPA 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.

The default RAT configuration as detailed in SGP.21 Annex H applies for all test sequences except if the Test Case overrides it:

- Only one PPAR authorizing PPR1 and PPR2 for all MNOs with End User consent required i.e. `#PPRS_ALLOWED`

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`.
  - If the eUICC only supports `O_E_NIST`, the ECASD is configured as defined in section G.2.2.
  - If the eUICC only supports `O_E_BRP`, the ECASD is configured as defined in section G.2.3.
  - 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:
  - `#PPR1_WITH_OWNER_GID`

- #PPR1\_WITHOUT\_GID
- #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.

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.

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 Document Management

### J.1 Document History

Version	Date	C R No	Brief Description of Change	Entity	Approval Authority	Editor / Company
v1.0	9 <sup>th</sup> June 2017		Initial version of SGP.23 v1.0 Test Specification		PSMC	Yolanda Sanz, GSMA
v1.1	28 <sup>th</sup> Sept 2017		Minor version of SGP.23 Test specifications		RSPLEN	Yolanda Sanz, GSMA
v1.2	3 <sup>rd</sup> Jan 2018		Minor version of SGP.23 Test specifications		RSPLEN	Yolanda Sanz, GSMA
V1.3	01 <sup>th</sup> August		Minor version of SGP.23 Test specification		RSPLEN	Yolanda Sanz, GSMA
V1.4	18 <sup>th</sup> 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

### J.2 Other Information

Type	Description
Document Owner	Yolanda Sanz
Editor / Company	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.