RSP Test Specification for the Device
SGP.23-2 Version 3.1
01 December 2023

Security Classification: Non-confidential

Copyright Notice

Disclaimer

Antitrust Notice

Table of Contents

1 Introduction

1.1 Overview

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

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

1.2 Scope

This document is intended for:

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

The Test Plan consists of a set of relevant test cases for the Device/LPA testing The only Implementations Under Test (IUT) within this document is the LPA. Test cases for the eUICC are defined in [30], Test cases for the Servers (SM-DP+, SM-DS) are defined in [31].

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 V3.1 [2]

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

1.3 Definition of Terms

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

Term	Description
End User	The person using the Device.
Integrated eUICC Test Interface	An external interface provided by its manufacturer for the purpose of testing eUICC functionality.

Term	Description
Standalone Device	A Device which provides all the capabilities to be able to be used in an RSP environment and needs no other Device for the purpose of Remote SIM Provisioning.
Test Plan	Current document describing the test cases that allow the RSP ecosystem to be tested.

1.4 Abbreviations

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

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

1.5 Document Cross-references

Ref	Document Number	Title
[1]	SGP.02	GSMA "Remote Provisioning of Embedded UICC Technical specification" V3.1
[2]	SGP.22	RSP Technical Specification V3.1
[3]	SGP.21	RSP Architecture V3.1
[4]	eUICC Profile Package	Trusted Connectivity Alliance (formerly SIMalliance) eUICC Profile Package: Interoperable Format Technical Specification V2.0 or later
[5]	ETSI TS 102 221	Smart Cards; UICC-Terminal interface

Ref	Document Number	Title
[6]	GPC_SPE_034	GlobalPlatform Card Specification v.2.3
[7]	ISO/IEC 7816-4:2013	Identification cards – Integrated circuit cards - Part 4: Organization, security and commands for interchange
[8]	RFC 5639	Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation
[9]	ANSSI ECC FRP256V1	Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français. JORF n°0241 du 16 octobre 2011 page 17533. texte n° 30. 2011
[10]	ITU E.118	The international telecommunication charge card
[11]	NIST SP 800-56A	NIST Special Publication SP 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revision 2), May 2013
[12]	3GPP TS 23.003	Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification
[13]	ETSI TS 102 225	Secured packet structure for UICC based applications; Release 12
[14]	ETSI TS 102 226	Remote APDU structure for UICC based applications; Release 9
[15]	TS.26	GSMA NFC Handset Requirements V9.0
[16]	ITU-T X.690 (11/2008)	ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) including Corrigendum 1 and 2
[17]	ETSI TS 102 241	Smart cards; UICC Application Programming Interface (UICC API) for Java Card™
[18]	3GPP TS 31.102	Characteristics of the Universal Subscriber Identity Module (USIM) application
[19]	GPC_SPE_095	GlobalPlatform Card - Digital Letter of Approval - Version 1.0
[20]	RFC 2119	Key words for use in RFCs to Indicate Requirement Levels, S. Bradner http://www.ietf.org/rfc/rfc2119.txt
[21]	Void	
[22]	3GPP TS 23.040	Technical realization of the Short Message Service (SMS)
[23]	TCA Test	Trusted Connectivity Alliance (TCA) eUICC Profile Package: Interoperable Format Test Specification Version 2.3.1
[24]	RFC 4492	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)
[25]	SGP.26	RSP Test Certificates Definition v1.5
[26]	3GPP TS 29.002	Mobile Application Part (MAP) specification
[27]	RFC 5246	The Transport Layer Security (TLS) Protocol Version 1.2
[28]	GSMA PRD AA.35	Procedures for Industry Specifications Product

Ref	Document Number	Title
[29]	CCID Rev 1.1	CCID Specification for Integrated Circuit(s) Cards Interface Devices
[30]	SGP.23-1 v3.1	RSP Test Specification for the eUICC
[31]	SGP.23-3 v3.1	RSP Test Specification for Servers

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.
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.

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

Table 3: Applicability and Notations

2.1.4 Optional Features Table

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

Device Options	Mnemonic
The Device supports LPAd	O_D_LPAD
The Device supports GSM/GERAN	O_D_GSM_GERAN
The Device supports UMTS/UTRAN	O_D_UMTS_UTRAN
The Device supports cdma2000 1X	O_D_CDMA2000_1X
The Device supports cdma2000 HRPD	O_D_CDMA2000_HRPD
The Device supports cdma2000 eHRPD	O_D_CDMA2000_EHRPD
The Device supports LTE/E-UTRAN	O_D_LTE
The Device supports NFC as defined in TS26	O_D_NFC_TS26
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
The Device provides the LUI functionality to postpone Profile Download after GetBPP	O_D_EU_POSTPONED_AFTER_GETBPP
Device supports Power-on Profile discovery	O_D_POW_ON_PROF_DISCOVERY

The Device provides the LUI functionality to reject Profile Download	O_D_EU_REJECT
The Device supports Set/Edit Nickname procedure as defined in SGP.22 [2] section 3.2.6 and displaying the profile's Nickname	O_D_NICKNAME
The Device supports additional verification of TLS certificate content (i.e. key usage, extended key usage and certificate policy)	O_D_TLS_FULL_VERIFICATION
The Device supports Add Profile and Enable Profile in one combined operation (See NOTE 2)	O_D_ADD_ENABLE_COMBINED
The Device supports Add Profile and Enable Profile as separated operations (See NOTE 2)	O_D_ADD_ENABLE_SEPARATED
Initiation of Add Profile procedure is allowed even if the currently enabled Profile has PPR1	O_D_ADDPREPPR1
The Device supports Set/Edit Default SM-DP+ Address procedure as defined in SGP.22 [2] section 3.3.4	O_D_DEFAULT_DP_ADDRESS
The Device supports a removable eUICC and downloading a profile containing PPRs to the removable eUICC.	O_D_REMOVABLE_DOWNLOAD_PPR
The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR1 is allowed and End User Consent is required.	O_D_EMB_ALLOWS_PPR1_EUC_REQ
The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR1 is allowed and End User Consent is NOT required.	O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ
The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR2 is allowed and End User Consent is required.	O_D_EMB_ALLOWS_PPR2_EUC_REQ
The Device supports a non-removable eUICC and eUICC RAT configurations in which PPR2 is allowed and End User Consent is NOT required.	O_D_EMB_ALLOWS_PPR2_EUC_NOT_REQ
The Device supports NR EPC	O_D_NR_EPC
The Device supports NR 5GC	O_D_NR_5GC
The Device supports E-UTRAN 5GC	O_D_EUTRAN_5GC
The Device supports RPM	O_D_RPM
The Device is Enterprise capable	O_D_ENTERPRISE
The Device supports indication of supported Card Application Toolkit letter classes	O_D_CAT_CLASSES
The Device supports Multiple Enabled Profiles	O_D_MEP
The Device supports Activation QR code scanning.	O_D_QR_SCANNING
NOTE 1: Devices which supports O_D_ONLY_CELLULAR_CONNECTIVITY are out of scope of the current version of this document. NOTE 2: The Device SHALL support at least one of O_D_ADD_ENABLE_COMBINED or O_D_ADD_ENABLE_SEPARATED. It is valid to support both options.	

Table 4: Options

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	V3.1	Test Env.
LPAd Interfaces Compliance Testing			
4.4.21.2.1	TC_LPAd_InitiateAuthentication_Nominal	C007	
4.4.21.2.2	TC_LPAd_InitiateAuthentication_ErrorCases	C007	
4.4.21.2.3	TC_LPAd_InitiateAuthentication_Nominal_V3	C007	
4.4.21.2.4	TBD	C007	
4.4.21.2.5	TC_LPAd_InitiateAuthentication_Norminal_RPM	C303	
4.4.21.2.6	TC_LPAd_InitiateAuthentication_ErrorCases_RPM	C303	
4.4.22.2.1	TC_LPAd_ES9+_GetBoundProfilePackage_Nominal	C007	
4.4.22.2.2	TC_LPAd_ES9+_GetBoundProfilePackage_Retry	C005	
4.4.22.2.3	TC_LPAd_ES9+_GetBoundProfilePackage_Error	C007	
4.4.22.2.4	TC_LPAd_ES9+_GetBoundProfilePackage_Retry_Reuse_oldOTPK	C302	
4.4.23.2.1	TC_LPAd_AuthenticatClient_Nominal	C007	
4.4.23.2.2	TC_LPAd_AuthenticateClient_ErrorCases	C007	
4.4.23.2.3	TC_LPAd_Authenticate_Client_Nominal_V3 Only the test sequences #01	C007	
4.4.23.2.3	TC_LPAd_Authenticate_Client_Nominal_V3 Only the test sequences #02	C303	
4.4.24.2.1	TC_LPAd_ES9+_HandleNotification_Nominal	C007	
4.4.25.2.1	TC_LPAd_ES9+_CancelSession_Nominal All test sequences except the sequence #02	C007	
4.4.25.2.1	TC_LPAd_ES9+_CancelSession_Nominal Only the test sequences #02	C023	
4.4.25.2.2	TC_LPAd_ES9+_CancelSession_EndUserPostponed_Nominal	C008	
4.4.25.2.3	TC_LPAd_ES9+_CancelSession_Error	C026	
4.4.25.2.4	TC_LPAd_ES9+_CancelSession_PPRs Only test sequence #1	C045	
4.4.25.2.4	TC_LPAd_ES9+_CancelSession_PPRs Only test sequence #2	C046	
4.4.25.2.5	TC_LPAd_ ES9+_CancelSession_AuthenticateClient_RPM	C303	
4.4.26.2.1	TC_LPAd_HTTPS_Nominal	C007	
4.4.26.2.2	TC_LPAd_HTTPS_ErrorCases	C007	

Test case	Name	V3.1	Test Env.
4.4.26.2.3	TC_LPAd_HTTPS_Nominal_Variants_V3	C007	
4.4.27.2.1	TC_LPAd_ES11_InitiateAuthentication_Nominal	C007	
4.4.27.2.2	TC_LPAd_ES11_InitiateAuthentication_ErrorCases	C007	
4.4.28.2.1	TC_LPAd_ES11_AuthenticateClient_Nominal	C007	
4.4.28.2.2	TC_LPAd_ES11_AuthenticateClient_ErrorCases	C007	
4.4.29.2.1	TC_LPAd_HTTPS_Nominal	C007	
4.4.29.2.2	TC_LPAd_HTTPS_Error	C007	
4.4.30.2.1	TC_LPAd_Authenticate_Client_Nominal_RPM	C303	
4.4.30.2.2	TC_LPAd_Authenticate_Client_Nominal_RPM_Enterprise_Device	C304	
4.4.32.2.1	TC_LPAd_RPM_ HandleNotification for RPM Package Download	C303	
Procedure - Behaviour Testing			
5.4.1.2.1	TC_LPAd_AddProfile_Manual_Entry	C035	
5.4.1.2.2	TC_LPAd_AddProfile_QRCode_scanning	C307	
5.4.1.2.3	TC_LPAd_AddProfile_ActivationCode_InvalidFormat_QRCode	C308	
5.4.1.2.4	TC_LPAd_AddProfile_ActivationCode_InvalidFormat_ManualEntry	C309	
5.4.1.2.5	TC_LPAd_AddProfile_ConfirmationCode_smdpSigned2_QR	C307	
5.4.1.2.6	TC_LPAd_AddProfile_ConfirmationCode_smdpSigned2_Manual_Entry	C035	
5.4.1.2.7	TC_LPAd_AddProfile_default_SM-DP+_address	C035	
5.4.1.2.8	TC_LPAd_AddProfile _with_ConfirmationCode Only test sequence #01	C307	
5.4.1.2.8	TC_LPAd_AddProfile _with_ConfirmationCode Only test sequence #02	C035	
5.4.1.2.9	TC_LPAd_AddProfile_PPRs Only test sequence #1	C047	
5.4.1.2.9	TC_LPAd_AddProfile_PPRs Only test sequence #2	C048	
5.4.1.2.9	TC_LPAd_AddProfile_PPRs Only test sequence #3	C051	
5.4.1.2.11	TC_LPAd_AddProfile_Security_Errors	C007	
5.4.1.2.12	TC_LPAd_AddProfile_Empty_MatchingID Only test sequence #01	C307	
5.4.1.2.12	TC_LPAd_AddProfile_Empty_MatchingID	C035	

Test case	Name	V3.1	Test Env.
	Only test sequence #02		
5.4.1.2.13	TC_LPAd_AddEnabledProfile_Manual_Entry	C034	
5.4.1.2.14	TC_LPAd_AddEnabledProfile Only test sequence #01	C306	
5.4.1.2.14	TC_LPAd_AddEnabledProfile Only test sequence #02	C034	
5.4.1.2.15	TC_LPAd_AddEnabledProfile_ConfirmationCode_smdpSigned2_QR	C306	
5.4.1.2.16	TC_LPAd_AddEnableProfile_ConfirmationCode_smdpSigned2_Manual_Entry	C034	
5.4.1.2.17	TC_LPAd_AddEnabledProfile_default_SM-DP+_address	C034	
5.4.1.2.18	TC_LPAd_AddEnableProfile_ with_ConfirmationCode Only test sequence #01	C306	
5.4.1.2.18	TC_LPAd_AddEnableProfile_with_ConfirmationCode Only test sequence #02	C034	
5.4.1.2.19	TC_LPAd_AddEnabledProfile_PPRs Only test sequence #1	C049	
5.4.1.2.19	TC_LPAd_AddEnabledProfile_PPRs Only test sequence #2	C050	
5.4.1.2.19	TC_LPAd_AddEnabledProfile_PPRs Only test sequence #3	C052	
5.4.1.2.20	TC_LPAd_AddEnableProfile_Empty_MatchingID Only test sequence #01	C306	
5.4.1.2.20	TC_LPAd_AddEnableProfile_Empty_MatchingID Only test sequence #02	C034	
5.4.2.2.1	TC_LPAd_ListProfiles	C007	
5.4.3.2.1	TC_LPAd_SetNickname	C027	
5.4.3.2.2	TC_LPAd_EditNickname	C027	
5.4.4.2.1	TC_LPAd_DeleteProfile_Disabled_without_PPR	C007	
5.4.4.2.2	TC_LPAd_DeleteProfile_Enabled_without_PPR	C009	
5.4.4.2.3	TC_LPAd_DeleteProfile_Error_with_PPR1	C012	
5.4.4.2.4	TC_LPAd_DeleteProfile_Error_Disabled_with_PPR2	C013	
5.4.4.2.5	TC_LPAd_DeleteProfile_Error_Enabled_with_PPR2	C014	
5.4.4.2.6	TC_LPAd_DeleteProfile_Security_Errors	C007	
5.4.4.2.7	TC_LPAd_DeleteProfiles_MEP_Enabled_without_PPR	C305	
5.4.5.2.1	TC_LPAd_EnableProfile	C009	
5.4.5.2.2	TC_LPAd_EnableProfile_ImplicitDisable	C009	

Test case	Name	V3.1	Test Env.
5.4.5.2.3	TC_LPAd_EnableProfile_Error_ProfileAlreadyEnabled	C010	
5.4.5.2.4	TC_LPAd_EnableProfile_Error_PPR1Set	C011	
5.4.5.2.6	TC_LPAd_EnableProfile_MEP	C305	
5.4.6.2.1	TC_LPAd_DisableProfile	C009	
5.4.6.2.2	TC_LPAd_DisableProfile_Error_ProfileAlreadyDisabled	C017	
5.4.6.2.3	TC_LPAd_DisableProfile_Error_PPR1Set	C018	
5.4.6.2.5	TC_LPAd_DisableProfile_MEP	C305	
5.4.7.2.1	TC_LPAd_RetrieveEID	C004	
5.4.8.2.1	TC_LPAd_eUICCMemoryReset Only test sequence #1 and test sequence #5	C007	
5.4.8.2.1	TC_LPAd_eUICCMemoryReset Only test sequence #2 and test sequence #3	C044	
5.4.8.2.1	TC_LPAd_eUICCMemoryReset Only test sequence #4	C043	
5.4.8.2.2	TC_LPAd_eUICCMemoryResetWithPINVerification	C009	
5.4.8.2.3	TC_LPAd_eUICCMemoryReset_MEP	C305	
5.4.10.2.1	TC_LPAd_Set/Edit Default SM-DP+ Address	C041	
5.4.11.2.1	TC_LPAd_DevicePowerOnProfileDiscovery_SM-DP+_address	C022	
5.4.11.2.2	TC_LPAd_DevicePowerOnProfileDiscovery_SM-DS	C022	
5.4.12.2.1	TC_LPAd_RPM_Command_Execution_EnableProfile	C303	
5.4.13.2.1	TC_LPAd_RPM_Command_Execution_DisableProfile	C303	
5.4.14.2.1	TC_LPAd_RPM_Command_Execution_DeleteProfile	C303	
5.4.15.2.1	TC_LPAd_RPM_Command_Execution_ListProfileInfo	C303	
5.4.16.2.1	TC_LPAd_RPM_Command_Execution_UpdateMetadata	C303	

Table 5: Applicability of Tests

Conditional item	Condition
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
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

Conditional item	Condition
C011	IF (O_D_LPAD AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_ALLOWS_PPR1_EUC_REQ OR O_D_ALLOWS_PPR1_EUC_NOT_REQ) AND O_D_ENPREVPPR1 AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEM M ELSE N/A
C012	IF (O_D_LPAD AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_ALLOWS_PPR1_EUC_REQ OR O_D_ALLOWS_PPR1_EUC_NOT_REQ) AND O_D_DISDELPPR1 AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEN M ELSE N/A
C013	IF (O_D_LPAD AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_ALLOWS_PPR2_EUC_REQ OR O_D_ALLOWS_PPR2_EUC_NOT_REQ) AND O_D_DELPPR2 AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEN M ELSE N/A
C014	IF (O_D_LPAD AND O_D_PIN AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_ALLOWS_PPR2_EUC_REQ OR O_D_ALLOWS_PPR2_EUC_NOT_REQ) AND O_D_DELPPR2 AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEN M ELSE N/A
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_REMOVABLE_DOWNLOAD_PPR OR O_D_ALLOWS_PPR1_EUC_REQ OR O_D_ALLOWS_PPR1_EUC_NOT_REQ) AND O_D_DISPPR1) THEN M ELSE N/A
C022	IF (O_D_LPAD AND O_D_POW_ON_PROF_DISCOVERY AND NOT 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
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
C034	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_COMBINED AND NOT O_D_QR_SCANNING) THEN M ELSE NA
C035	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_SEPARATED AND NOT O_D_QR_SCANNING) THEN M ELSE NA
C041	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_DEFAULT_DP_ADDRESS) THEN M ELSE N/A
C043	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_REQ OR O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ)) THEN M ELSE N/A
C044	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND (O_D_REMOVABLE_DOWNLOAD_PPR OR

Conditional item	Condition
	O_D_EMB_ALLOWS_PPR2_EUC_REQ OR O_D_EMB_ALLOWS_PPR2_EUC_NOT_REQ)) THEN M ELSE N/A
C045	IF ((O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) AND (O_D_EU_POSTPONED OR O_D_EU_REJECT) AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_REQ)) THEN M ELSE N/A
C046	IF ((O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) AND (O_D_EU_POSTPONED OR O_D_EU_REJECT) AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR2_EUC_REQ)) THEN M ELSE N/A
C047	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_SEPARATED AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_REQ)) THEN M ELSE NA
C048	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_SEPARATED AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR2_EUC_REQ)) THEN M ELSE NA
C049	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_COMBINED AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_REQ)) THEN M ELSE NA
C050	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_COMBINED AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR2_EUC_REQ)) THEN M ELSE NA
C051	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_SEPARATED AND O_D_ADDPREPPR1 AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_REQ OR O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ) THEN M ELSE N/A
C052	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_COMBINED AND O_D_ADDPREPPR1 AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_REQ OR O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ) THEN M ELSE N/A
Conditions applicable to SGP.23 v3.1 only	
C302	IF (O_D_EU_POSTPONED_AFTER_GETBPP AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY) THEN M ELSE N/A
C303	IF O_D_RPM THEN M ELSE N/A
C304	IF O_D_RPM AND O_D_ENTERPRISE THEN M ELSE N/A
C305	IF O_D_MEP THEN M ELSE N/A
C306	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_COMBINED AND O_D_QR_SCANNING) THEN M ELSE N/A

Conditional item	Condition
C307	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_ADD_ENABLE_SEPARATED AND O_D_QR_SCANNING) THEN M ELSE N/A
C308	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND O_D_QR_SCANNING) THEN M ELSE N/A
C309	IF (O_D_LPAD AND NOT O_D_ONLY_CELLULAR_CONNECTIVITY AND NOT O_D_QR_SCANNING) THEN M ELSE N/A

Table 6: Conditional Items Referenced by Table 6

Note: Conditions C0XX which are missing in Table 6 are present in an earlier version of SGP.23 but are not used in the current version.

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.

4.X.Y.Z Test Cases

4.X.Y.Z.1 TC_IUT_TestName1

General Initial Conditions	
Entity	**Description of the general initial condition**
Entity1	Test case - general condition 1
Entity2	Test case - general condition 2

Test Sequence #01: Short Description

Description of the aim of the test sequence N°1

Initial Conditions	
Entity	**Description of the initial condition**
Entity1	Test sequence N°1 - initial condition 1
Entity2	Test sequence N°1 - initial condition 2

Step	Direction	Sequence / Description	Expected result
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
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
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

4.X.Y.Z.2 TC_IUT_TestName2

…

The test cases TC_IUT_TestName1 and TC_IUT_TestName2 are referenced in Table 5 that allows indicating the applicability of the tests.

In the test case TC_IUT_TestName1, the requirements REQ1 and REQ2 are respectively covered by the test sequences #01 and #02.

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

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

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

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

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

The tables defining the different initial conditions are optional.

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

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

In the test sequence #01:

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

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

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

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

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.

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

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

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

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 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 VOID

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 <EUICC_CI_PK_ID_TO_BE_USED> set to the CI for signing indicated as highest priority in euiccCiPKIdListForSigning in the <R_EUICC_INFO1>
    - with the #CERT_S_SM_DPauth_SIG 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_SIG leading to the same CI as the one chosen for signing
    - Metadata of the downloaded Profile instead of #METADATA_OP_PROF1

- #GET_BPP_OK with the content of the installed Profile (no session keys used)

Before running a test sequence, and after establishing the Initial conditions, all pending Notifications (sent on the best-effort basis as soon as connectivity is available as defined in section 3.5 of SGP.22 [2]) SHALL have been acknowledged by the simulated SM-DP+(s). S_SM-DP+(s) SHALL be run with suitable addresses in order to receive and acknowledge all pending Notifications (including install, enable, disable and delete). The addresses which are required depend on the server address used for recent profile downloads (typically #TEST_DP_ADDRESS1 to receive and acknowledge PIR), and the notificationAddress values in the Metadata of recently downloaded Profiles (for otherSignedNotification). Each S_SM_DP+ SHALL use the TLS certificate corresponding to its address (CERT_S_SM_DP_TLS, CERT_S_SM_DP2_TLS, etc).

If only O_D_ADD_ENABLE_COMBINED is supported by the DUT, the user might have to perform actions in a particular manner in order to achieve the initial conditions related to enabled/disabled state of profiles (for example: disable a profile after installing, install profiles in a particular order, re-enable an initial profile after installing a subsequent profile).

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 protection mechanism is enforced upon entry to the LUI as defined in SGP.22 [2].

The way to perform Strong Confirmation SHALL be executed by the S_EndUser according to the description provided by the Device Vendor in #IUT_LPAd_Confirmation.

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

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

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

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

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

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

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

2.2.4.3 TLS Testing Rules and 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.

Unless it is defined differently in test case, the S_SM-DP or S_SM_DS is configured to respond to the “ClientHello” in the following way:

- Certificates that chain to eSIM CA SHALL be used, even if the received ClientHello message contains the "server_name" extension with the v3-specific FQDN

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

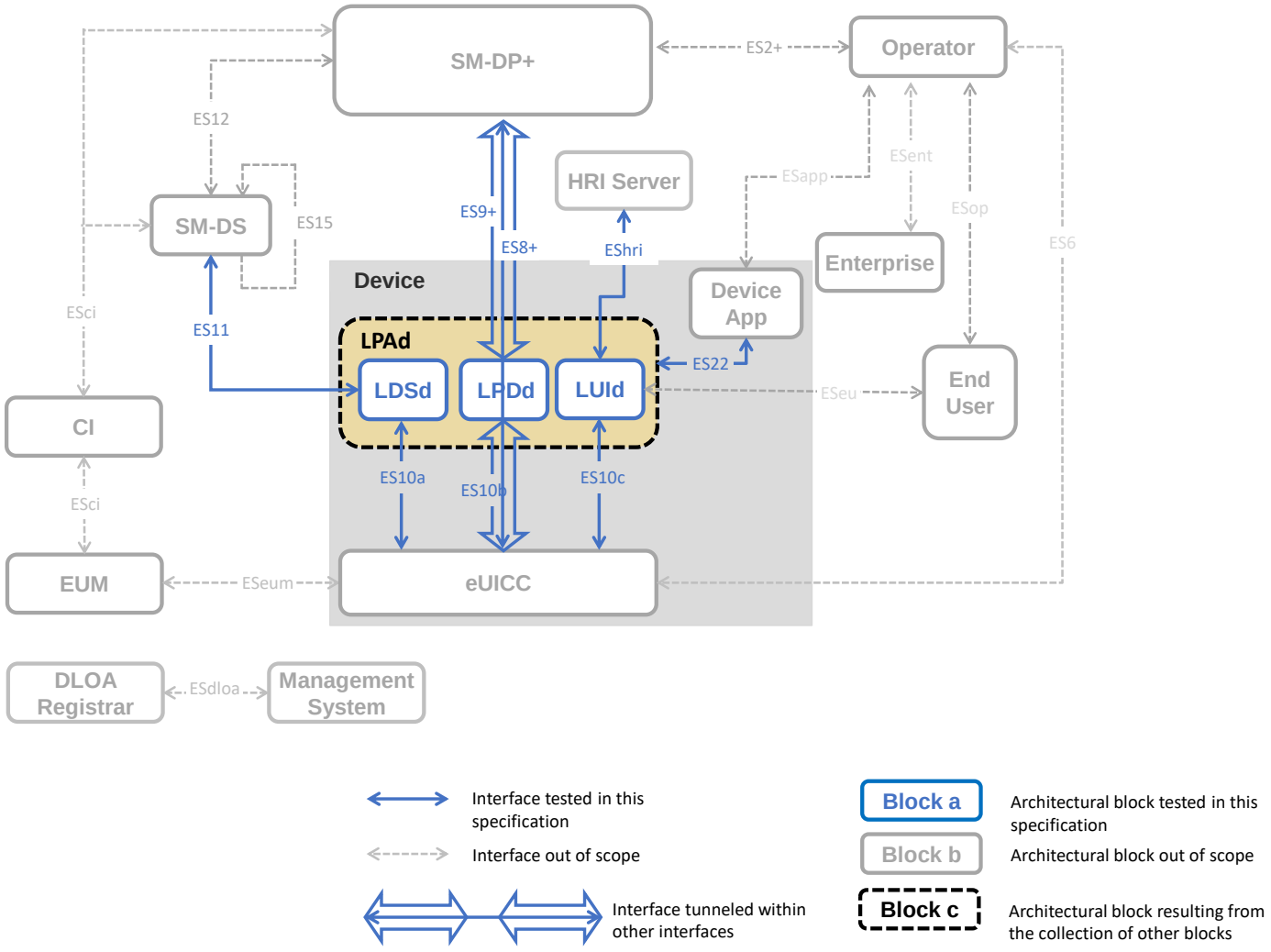
Some of the test cases or test sequences described in this Test Plan are FFS (For Future Study). This MAY mean that some clarifications are expected at the requirement level to conclude on a test method. As consequence, the corresponding test SHALL not be executed.

2.2.7 VOID

3 Testing Architecture

3.1 Testing Scope

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



Interface	Between		Description
ES2+	Operator	SM-DP+	Used by the Operator to order Profiles for specific eUICCs as well as other administrative functions. NOTE: this interface is out of scope of this specification.
ES8+	SM-DP+	eUICC	Provides a secure end-to-end channel between the SM-DP+ and the eUICC for the administration of the ISD-P and the associated Profile during download and installation. It provides Perfect Forward Secrecy.
ES9+	SM-DP+	LPD	Used to provide a secure transport between the SM-DP+ and the LPA (LPD) for the delivery of the Bound Profile Package and the delivery of Remote Profile Management Commands.
ES11	LDS	SM-DS	Used by the LDS to retrieve Event Records for the respective eUICC.

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, specific simulators SHALL be used. The simulators that have been defined are listed hereafter:

- S_SM-DP+: the SM-DP+ Simulator
- S_SM-DS: the SM-DS 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_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. 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 VOID

3.2.2 VOID

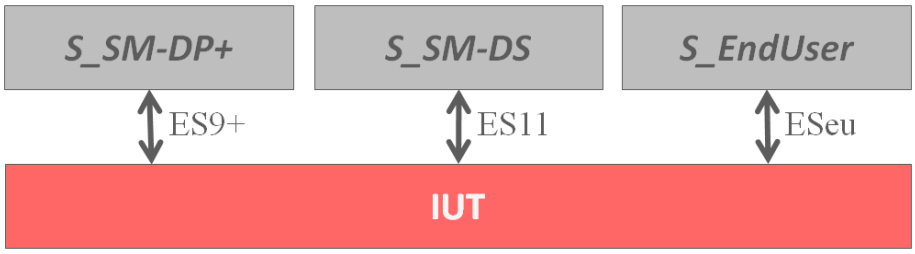
3.2.3 Device/LPAd - Test Environment

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

- The Device contains an eUICC configured with Test Certificates and Test Keys
- The Test eUICC is either soldered or removable. In case the eUICC is removable, it SHALL NOT be removed during testing
- The Test eUICC is only used for LPAd testing and SHALL not be considered as an IUT
- The Test eUICC SHALL not support LPAe
- The Test eUICC SHALL 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.

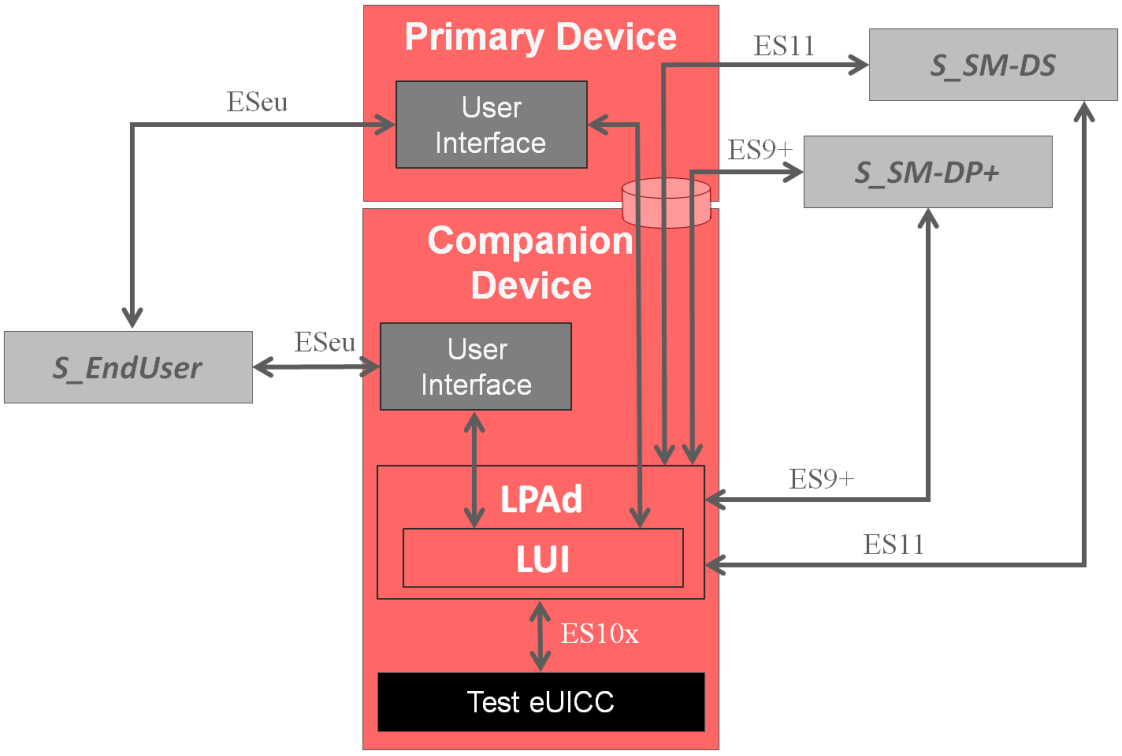
Device Vendors SHALL provide the mechanism to connect the Primary Device to the Companion Device.

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 VOID

3.2.5 VOID

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 VOID

4.3 VOID

4.4 LPAd Interfaces

4.4.1 ES10a (LPA -- eUICC): GetEuiccConfiguredAddresses

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

4.4.2 ES10a (LPA -- eUICC): SetDefaultDpAddress

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

4.4.3 ES10b (LPA -- eUICC): PrepareDownload

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

4.4.4 ES10b (LPA -- eUICC): LoadBoundProfilePackage

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

4.4.5 ES10b (LPA -- eUICC): GetEUICCChallenge

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

4.4.6 ES10b (LPA -- eUICC): GetEUICCInfo

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

4.4.7 ES10b (LPA -- eUICC): ListNotification

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

4.4.8 ES10b (LPA -- eUICC): RetrieveNotificationsList

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

4.4.9 ES10b (LPA -- eUICC): RemoveNotificationFromList

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

4.4.10 ES10b (LPA -- eUICC): LoadCRL

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

4.4.11 ES10b (LPA -- eUICC): AuthenticateServer

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

4.4.12 ES10b (LPA -- eUICC): CancelSession

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

4.4.13 ES10c (LPA -- eUICC): GetProfilesInfo

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

4.4.14 ES10c (LPA -- eUICC): EnableProfile

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

4.4.15 ES10c (LPA -- eUICC): DisableProfile

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

4.4.16 ES10c (LPA -- eUICC): DeleteProfile

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

4.4.17 ES10c (LPA -- eUICC): eUICCMemoryReset

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

4.4.18 ES10c (LPA -- eUICC): GetEID

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

4.4.19 ES10c (LPA -- eUICC): SetNickname

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

4.4.20 ES10b (LPA -- eUICC): GetRAT

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

4.4.21 ES9+ (LPA -- SM-DP+): InitiateAuthentication

4.4.21.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 2.1, 2.4a.1.3
- Section 2.6.6.2, 2.10.1
- Section 3.0.1, 3.1.3, 3.7.2, 3.7.3
- Section 5.6.1, 5.6.3, 5.7.14a
- Section 6.2
- Section 6.3
- Section 6.5.1, 6.5.1.1, 6.5.1.2, 6.5.1.3, 6.5.1.4, 6.5.2, 6.5.2.6
- Section 6.6.2.2

4.4.21.2 Test Cases

4.4.21.2.1 TC_LPAd_InitiateAuthentication_Nominal

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).
eUICC	There is no default SM-DP+ address configured.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.

Test Sequence #01 Nominal: Initiate Authentication

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE> Verify if: • <LPA_RSP_CAPABILITY> contains • `crlStaplingV3Support` set to '1' • `certChainV3Support` set to '1' • `signedSmdsResponseV3Support` set to '1'
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases

General Initial Conditions	
Entity	**Description of the general initial condition**
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.

Test Sequence #01 Error: Invalid SM-DP+ Address

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_8_1_3_8)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout

			#IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #02 Error: Unsupported Security Configuration

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_8_2_3_1)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #03 Error: Unsupported SVN

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_8_3_3_1)	LPAd aborts AddProfile procedure RQ56_008 RQ56_011
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #04 Error: Unavailable SM-DP+ Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1,

			#TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_8_4_3_7)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #05 Error: Invalid SM-DP+ Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_CERT)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #06 Error: Invalid SM-DP+ Signature

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_SIGN)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #07 Error: Invalid SM-DP+ Address sent by the SM-DP+

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_SMDP +_ADDRESS)	LPAd informs the S_EndUser and aborts the AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #08 Error: Unsupported CI Key ID

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_CI)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

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

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated, #ACTIVATION_CODE_2 is provided to the LPAd on request from the S_EndUser.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_2 (PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_OID)	LPAd informs the S_EndUser and aborts the AddProfile procedure
2	LPAd → S_SM-DP+	No Profile download action	No ES9+.InitiateAuthentication or ES9+.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

4.4.21.2.3 TC_LPAd_InitiateAuthentication_Nominal_V3

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)
eUICC	There is no default SM-DP+ address configured
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: Initiate Authentication – V3 Variant

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>

2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_FLEX)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

4.4.21.2.4 TC_LPAd_InitiateAuthentication_ErrorCases_V3

TBD

4.4.21.2.5 TC_LPAd_ **InitiateAuthentication_Norminal**_RPM

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 The eUICC supports RPM
LPAd	RPM operation is enabled in the LPA by the End User
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: Initiate Authentication for RPM – Variant A, with Polling Address and allowed GSMA CI RootCA public key identifier

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_CI_PKI_RPM is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address and the Allowed GSMA CI RootCA public key identifier (#EUICC_CI_PKI_RPM) are retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1,

			#IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

Test Sequence #02 Nominal: Initiate Authentication for RPM – Default SM-DP+ Address and allowed GSMA CI RootCA public key identifier

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with # METADATA_OP_PROF1_RPM_CONF_EN_NO_POLLING_ADDRESS is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Default SM-DP+ Address (#TEST_DP_ADDRESS1) and the Allowed GSMA CI RootCA public key identifier (#EUICC_CI_PKI_RPM) are set using ES10a.SetDefaultDpAddress and retrieved using the ES10a.GetEuiccConfiguredData.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

Test Sequence #03 Nominal: Initiate Authentication for RPM – Root SM-DS Address as Polling Address and allowed GSMA CI RootCA public key identifier

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_POL_SMDS_CI_PKI_RPM is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address and the Allowed GSMA CI RootCA public key identifier (#PK_CI_ECDSA_RPM) are retrieved from the ProfileInfo, and the Polling Address indicates the S_SM-DS address #TEST_ROOT_DS_ADDRESS.
S_SM-DS	S_SM-DP+ (#TEST_DP_ADDRESS1) performed RPM Download Event Registration to the S_SM-DS (#TEST_ROOT_DS_ADDRESS) with #EVENT_ID_1
S_SM-DP+	There is a pending RPM package order for #EVENT_ID_1 (PROFILE_OPERATIONAL1)

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES11		
1	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK_VARIANT_A)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

Test Sequence #04 Nominal: Initiate Authentication for RPM – Root SM-DS Address and no default SM-DP+ address configured

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_NO_POLLING_ADDRESS is loaded and disabled.

	eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
eUICC	S_SM_DS address #TEST_ROOT_DS_ADDRESS is configured in eUICC and no Default S_SM-DP+ is set in eUICC.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. The Root SM-DS Address (#TEST_ROOT_DS_ADDRESS) is retrieved using the ES10a.GetEuiccConfiguredData.
S_SM-DS	S_SM-DP+ (#TEST_DP_ADDRESS1) performed RPM Download Event Registration to the S_SM-DS (#TEST_ROOT_DS_ADDRESS) with #EVENT_ID_1
S_SM-DP+	There is a pending RPM package order for #EVENT_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES11		
1	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK_VARIANT_A)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

Test Sequence #05 Nominal: Initiate Authentication for RPM – previous Profile Download session with pending RPM

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	Installation of PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_POL_DP8 is triggered using the Activation Code (#ACTIVATION_CODE_1) with S_SM-DP+ address (#TEST_DP_ADDRESS1).
S_SM-DP+	There is a pending RPM package order for (PROFILE_OPERATIONAL1). During Mutual Authentication for the Profile Download S_SM-DP+ sends the AuthenticateClientOk with #S_SMDP_SIGNED2_RPM_PENDING.
eUICC	PROFILE_OPERATIONAL1 with is loaded and is disabled.

LPAd	No Update Profile operation is initiated for RPM download and it is triggered by the rpmPending received from the Profile Download session.
eUICC	eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer for RPM download.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

Test Sequence #06 Nominal: Initiate Authentication for RPM – previous RPM Download session with pending RPM

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_POL_DP8 is loaded and is disabled
LPAd	Update Profile operation for RPM download to enable PROFILE_OPERATIONAL1 is initiated.
S_SM-DP+	There is a pending RPM package, Enable profile order for (PROFILE_OPERATIONAL1). During Mutual Authentication for the RPM Download S_SM-DP+ sends the #AUTH_CLIENT_RPM_OK_AND_RPM_PENDING. S_SM-DP+ has another pending RPM package for (PROFILE_OPERATIONAL1), eg: Disable profile order.
eUICC	PROFILE_OPERATIONAL1 is enabled using the RPM download.
LPAd	No Update Profile operation is initiated again for the pending RPM download (it is triggered by the rpmPending received from the previous RPM Download session).
eUICC	eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer for RPM download.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

4.4.21.2.6 TC_LPAd_ InitiateAuthentication_ErrorCases_RPM

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 The eUICC supports RPM
LPAd	RPM operation is enabled in the LPA by the End User
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Error: InitiateAuthentication for RPM, Polling Address and not matching allowed GSMA CI Root CA public key identifier

The purpose of this test sequence is to ensure that LPAd stops the RPM procedure if there is a restriction to a single allowed GSMA CI RootCA public key identifier and it does not match the Subject Key Identifier of the GSMA Root CI corresponding to CERT.XXauth.SIG.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_CI_PKI_RAND is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected.

	Polling Address and the Allowed GSMA CI RootCA public key identifier (#EUICC_CI_PKI_RANDOM) are retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	LPAd aborts Update Profile procedure.
3	LPAd → S_SM-DP+	No retry of Update Profile operation	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #02 Error: InitiateAuthentication for RPM, Default SM-DP+ Address and not matching allowed GSMA CI Root CA public key identifier

The purpose of this test sequence is to ensure that LPAd stops the RPM procedure if there is a restriction to a single allowed GSMA CI RootCA public key identifier and it does not match the Subject Key Identifier of the GSMA Root CI corresponding to CERT.XXauth.SIG.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_NO_POLLING_ADDRESS is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Default SM-DP+ Address (#TEST_DP_ADDRESS1) and the Allowed GSMA CI RootCA public key identifier (#EUICC_CI_PKI_RANDOM) are set using ES10a.SetDefaultDpAddress and retrieved using the ES10a.GetEuiccConfiguredData.

S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	LPAd aborts Update Profile procedure.
3	LPAd → S_SM-DP+	No retry of Update Profile operation	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #03 Error: InitiateAuthentication for RPM, Root SM-DS Address as Polling Address and not matching allowed GSMA CI Root CA public key identifier

The purpose of this test sequence is to ensure that LPAd stops the RPM procedure if there is a restriction to a single allowed GSMA CI RootCA public key identifier and it does not match the Subject Key Identifier of the GSMA Root CI corresponding to CERT.XXauth.SIG.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_POL_SMDS_CI_PKI_RAND is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address and the Allowed GSMA CI RootCA public key identifier (#PK_CI_ECDSA_RANDOM) are retrieved from the ProfileInfo, and the Polling Address indicates the S_SM-DS address #TEST_ROOT_DS_ADDRESS.
S_SM-DS	S_SM-DP+ (#TEST_DP_ADDRESS1) performed RPM Download Event Registration to the S_SM-DS (#TEST_ROOT_DS_ADDRESS) with #EVENT_ID_1

S_SM-DP+	There is a pending RPM package order for #EVENT_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES11		
1	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK_VARIANT_A)	LPAd aborts Update Profile procedure.
3	LPAd → S_SM-DS	No retry of Update Profile operation	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #04 Error: InitiateAuthentication for RPM, Pending RPM in previous RSP Session and not matching GSMA CI Root CA public key indicator in AC

The purpose of this test sequence is to ensure that LPAd stops the RPM procedure if there is a restriction to a single allowed GSMA CI RootCA public key identifier and it does not match the Subject Key Identifier of the GSMA Root CI corresponding to CERT.XXauth.SIG while using the SM-DP+ address from previous RSP Session.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	Installation of PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_POL_DP8 is triggered using the Activation Code (#ACTIVATION_CODE_1) with S_SM-DP+ address (#TEST_DP_ADDRESS1).
S_SM-DP+	There is a pending RPM package order for (PROFILE_OPERATIONAL1). During Mutual Authentication for the Profile Download S_SM-DP+ sends, • #CERT_S_SM_DPauth_ECDSA in Initiate Authentication response. the AuthenticateClientOk with #S_SMDP_SIGNED2_RPM_PENDING.
eUICC	PROFILE_OPERATIONAL1 with is loaded and is disabled.
LPAd	No Update Profile operation is initiated for RPM download and it is triggered by the rpmPending received from the Profile Download session.

S_SM-DP+	During Mutual Authentication for the RPM Download S_SM-DP+ sends #CERT_S_SM_DPauth_PK_CI2_ECDSA in Initiate Authentication response.
eUICC	eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer for RPM download.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_DIFF_CERT_VARIANT_A)	LPAd aborts Update Profile procedure.
3	LPAd → S_SM-DP+	No retry of Update Profile operation	No ES9+.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

4.4.22 ES9+ (LPA -- SM-DP+): GetBoundProfilePackage

4.4.22.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]

Requirements

-
- Section 3.0.1, 3.1.3,
- Section 5.6.2,
- Section 6.5, 6.5.2.7

4.4.22.2 Test Cases

4.4.22.2.1 TC_LPAd_ES9+_GetBoundProfilePackage_Nominal

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Nominal: Get BPP using S-ENC and S-MAC without Confirmation Code

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
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_SIG
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error, see NOTE.
NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested.			

Test Sequence #02 Nominal: Get BPP using S-ENC and S-MAC with Confirmation Code

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3.

S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT_CC		
IC4	LPAd → S_EndUser	LPAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE1 is provided by manual entry.
1	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_WITH_CC)) Verify if: • <S_TRANSACTION_ID> is the same as in #R_PREP_DOWNLOAD_WITH_CC • <EUICC_SIGNATURE2> using the #PK_EUICC_SIG • <S_HASHED_CC> = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE1, <S_TRANSACTION_ID>)
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error, see NOTE.
NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested.			

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

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
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_SIG
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK_PPK)	No error, see NOTE.
NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested.			

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

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT_CC		
IC4	LPAd → S_EndUser	LPAd requests the Confirmation Code from the S_EndUser.	
1	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_WITH_CC)) Verify if: • <S_TRANSACTION_ID> is the same as in #R_PREP_DOWNLOAD_WITH_CC • <EUICC_SIGNATURE2> using the #PK_EUICC_SIG • <S_HASHED_CC> = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE1, <S_TRANSACTION_ID>)
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK_PPK)	No error, see NOTE.

NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested.

4.4.22.2.2 TC_LPAd_ES9+_GetBoundProfilePackage_Retry

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Nominal: Get BPP Retry after incorrect Confirmation Code

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT_CC		
IC4	LPAd → S_EndUser	LPAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.
IC5	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_WITH_CC)) Verify if: <S_HASHED_CC> = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE2, <S_TRANSACTION_ID>)
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_7_3_8)	Continue to step 2
2	S_SM-DP+ closes TLS session (unless ,LPAd has already closed TLS session)		
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT_CC		

6	LPAd → S_EndUser	LPAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE1 is provided by manual entry.
7	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_WITH_CC)) Verify if: • If <S_TRANSACTION_ID> is the same as in #R_PREP_DOWNLOAD_WITH_CC • <EUICC_SIGNATURE2> using the #PK_EUICC_SIG • <S_HASHED_CC> = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE1, <S_TRANSACTION_ID>)
8	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error, see NOTE 1.
NOTE: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested.			

4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.

Test Sequence #01 Error: Wrong eUICC Signature

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_6_1)	LPAd aborts AddProfile procedure NOTE: The LPAd MAY retry by restarting the Profile download and installation procedure.

Test Sequence #02 Error: BPP Not Available

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_3_7)	LPAd aborts AddProfile procedure NOTE: the LPAd MAY retry by restarting the Profile download and installation procedure.

Test Sequence #03 Error: Unknown TransactionID received by SM-DP+

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	LPAd → S_SM-DP+	Send ES9+.GetBundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_10_1_3_9)	LPAd aborts AddProfile procedure NOTE: the LPAd MAY retry by restarting the Profile download and installation procedure.

Test Sequence #04 Error: Missing Confirmation Code

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_7_2_2)	LPAd aborts AddProfile procedure NOTE: the LPAd MAY retry by restarting the Profile download and installation procedure.

Test Sequence #05 Error: VOID

Test Sequence #06 Error: Wrong Confirmation Code

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT_CC		
IC4	LPAd → S_EndUser	LPAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.
IC5	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_WITH_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_7_3_8)	LPAd aborts AddProfile procedure NOTE: The LPAd MAY retry by restarting the Profile download and installation procedure

Test Sequence #07 Error: Maximum number of Confirmation Code retries exceeded

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT_CC		
IC4	LPAd → S_EndUser	LPAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.

IC5	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_WITH_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_7_6_4)	LPAd aborts AddProfile procedure The LPAd SHALL NOT retry by restarting the Profile download and installation procedure.

4.4.22.2.4 TC_LPAd_ES9+_GetBoundProfilePackage_Retry_Reuse_oldOTPK

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 with Old Keys

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC)) Extract the <OTPK_EUICC_AKA> and reuse the same value in step 10.
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error. The LPAd MAY display any relevant part of the Profile Metadata and offers the S_EndUser to postpone the Profile installation. Request for Strong Confirmation, if not requested before and not aborted.
2	S_EndUser → LPAd	End User Postpone is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	

3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED))
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	No Error
5	S_SM-DP+ closes TLS session (unless LPAd has already closed TLS session)		
6	Restart “Add Profile” Procedure as defined in the initial conditions.		
7	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
8	PROC_ES9+_INIT_AUTH		
9	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
10	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK_OLD_KEYS)	No Error
11	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)) The <OTPK_EUICC_AKA> present in euiccSigned2 may be the same or different as in #AUTH_CLIENT_OK_OLD_KEYS.
12	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error, see NOTE 1.
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. Request for Strong Confirmation, if not requested before and not aborted.			

4.4.23 ES9+ (LPA -- SM-DP+): AuthenticateClient

4.4.23.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 2.1, 2.6.6.2, 2.10.1
- Section 3.0.1, 3.1.3, 3.1.3.2, 3.2.7, 3.7.2
- Section 4.2, 4.3
- Section 5.6, 5.6.1, 5.6.3
- Section 5.7.5

- Section 6.2
- Section 6.3

4.4.23.2 Test Cases

4.4.23.2.1 TC_LPAd_AuthenticateClient_Nominal

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Nominal: Authenticate Client without Confirmation Code

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>, <LPA_RSP_CAPABILITY>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO)) Verify: • if #R_AUTH_SERVER_MATCH_ID_DEV_INFO used with the #MATCHING_ID_1 • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • if <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK • for #DEVICE_INFO:

			- The value of the TAC corresponds to the first 8 digits of #IUT_IMEI and is represented as a string of 4 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] - if IMEI is present then its value corresponds to #IUT_IMEI and is represented as a string of 8 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] except that the last octet contains the check digit (in high nibble) and an'F' filler (in low nibble) - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease (or eutranEpcSupportedRelease if #IUT_RSP_VERSION is v2.2.2 or higher) is set to the highest release as defined in #IUT_LTE_EUTRAN_REL. – if O_D_NFC_TS26 then contactlessSupportedRelease is set to the highest release as defined in #IUT_NFC_REL. – that rspCrlSupportedVersion is not present. – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - if O_D_CAT_CLASSES then catSupportedClasses is set to the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - lpaSvn is present and set to #IUT_RSP_VERSION_HIGHEST

			- euiccFormFactorType is present and set to the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. - lpaRspCapability is present and equal to <LPA_RSP_CAPABILITY> in IC2 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, O_D_CAT_CLASSES, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC, if the option is not set, verify that the corresponding field in DeviceCapabilities is not present.
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK)	Next step of profile download procedure is performed (i.e. GetBoundProfilePackage request is sent).

Test Sequence #02 Nominal: Authenticate Client with Confirmation Code

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO)) Verify:

			• if #R_AUTH_SERVER_MATCH_ID_DEV_INFO used with the #MATCHING_ID_3 • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • if <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK • for #DEVICE_INFO: - The value of the TAC corresponds to the first 8 digits of #IUT_IMEI and is represented as a string of 4 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] - if IMEI is present then its value corresponds to #IUT_IMEI and is represented as a string of 8 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] except that the last octet contains the check digit (in high nibble) and an 'F' filler (in low nibble) - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL – if O_D_LTE then eutranSupportedRelease (or eutranEpcSupportedRelease if #IUT_RSP_VERSION is v2.2.2 or higher) is set to the highest release as defined in #IUT_LTE_EUTRAN_REL. – if O_D_NFC_TS26 then contactlessSupportedRelease is set to the highest release as defined in #IUT_NFC_REL. – that rspCrlSupportedVersion is not present. – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL,

			– if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - if O_D_CAT_CLASSES then catSupportedClasses is set to the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - lpaSvn is present and set to #IUT_RSP_VERSION_HIGHEST - euiccFormFactorType is present and set to the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. - lpaRspCapability is present and equal to <LPA_RSP_CAPABILITY> in IC2 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, O_D_CAT_CLASSES, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC, if the option is not set, verify that the corresponding field in DeviceCapabilities is not present.
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK_CC)	Next step of profile download procedure is performed (i.e. GetBoundProfilePackage request is sent).

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

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT_CC		

IC4	LPAd → S_EndUser	LPAd requests the Confirmation Code from the S_EndUser.	#CONFIRMATION_CODE2 is provided by manual entry.
IC5	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	Verify if: <S_HASHED_CC> = MTD_GENERATE_HASHED_CC(#CONFIRMATION_CODE2, <S_TRANSACTION_ID>)
IC6	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_7_3_8)	
IC7	Restart Add Profile procedure if O_D_CC_RETRY not supported		
IC8	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC9	PROC_ES9+_INIT_AUTH		
IC10	S_SM-DP+ → LPAd	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK_CC)	Next step of profile download procedure is performed (i.e. GetBoundProfilePackage request is sent).

4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases

General Initial Conditions	
Entity	**Description of the general initial condition**
S_SM-DP+	There is a pending Profile download order for MATCHING_ID_1 (PROFILE_OPERATIONAL1).
eUICC	There is no default SM-DP+ address configured.
Device	The protection of access to the LUI is disabled.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.

Test Sequence #01 Error: VOID

Test Sequence #02 Error: VOID

Test Sequence #03 Error: Invalid eUICC or EUM Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))

2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_3_6_1)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #04 Error: Expired eUICC Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_3_6_3)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #05 Error: Invalid eUICC Signature or serverChallenge

Step	Direction	Sequence / Description	Expected result	REQ
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+			
IC2	PROC_ES9+_INIT_AUTH			
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))	
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_6_1)	LPAd aborts AddProfile procedure	
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.	

Test Sequence #06 Error: Insufficient Memory

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		

1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_4_8)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #07 Error: Unknown CI Root Key

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_11_1_3_9)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #08 Error: Profile not Allowed (Not in 'released' State)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_1_2)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #09 Error: Unknown TransactionID

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_10_1_3_9)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #10 Error: Refused MatchingID

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_6_3_8)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #11 Error: Refused EID

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_1_3_8)	LPAd aborts AddProfile procedure

3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #12 Error: No Eligible Profile for this eUICC/Device

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_2_5_4_3)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #13 Error: Expired Download Order

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_8_5_4_10)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #14 Error: Maximum Number of Retries Exceeded

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT,

			MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_8_5_6_4)	LPAd aborts AddProfile procedure
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #15 Error: Invalid SM-DP+(pb) certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#AUTH_CLIENT_INV_PB_CERT)	LPAd aborts AddProfile procedure (See NOTE)
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.
NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Confirmation.			

Test Sequence #16 Error: Different OID for SM-DP+ Certificates (CERT.DPpb.SIG and CERT.DPauth.SIG not belonging to the same entity)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	AuthenticateClient	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#AUTH_CLIENT_INV_CI)	LPAd aborts AddProfile procedure (See NOTE)
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Confirmation.

Test Sequence #17 Error: Invalid SM-DP+ signature (smdpSignature2)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#AUTH_CLIENT_INV_SIGN)	LPAd aborts AddProfile procedure (See NOTE)
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.
NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Confirmation.			

Test Sequence #18 Error: Invalid TransactionID sent by SM-DP+

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient Method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#AUTH_CLIENT_INV_TRANSACTION_ID)	LPAd aborts AddProfile procedure (See NOTE)
3	LPAd → S_SM-DP+	No Profile download action	No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.
NOTE: Before the AddProfile procedure is aborted, the LPAd may request for Confirmation from the S_EndUser. In this case the S_EndUser SHALL give the Confirmation.			

4.4.23.2.3 TC_LPAd_AuthenticateClient_Nominal_V3

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
eUICC	There is no default SM-DP+ address configured
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	Variant A certificates are included in certificate chain for TLS procedures

Test Sequence #01 Nominal: Authenticate Client V3 Variant 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
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_FLEX)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify that : • #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK_VARIANT_A • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1_V3>

			present in #INITIATE_AUTH_OK_VARIANT_FLEX • for #DEVICE_INFO (deviceCapabilities) verify that: - The value of the TAC corresponds to the first 8 digits of #IUT_IMEI and is represented as a string of 4 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] - if IMEI is present then its value corresponds to #IUT_IMEI and is represented as a string of 8 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] except that the last octet contains the check digit (in high nibble) and an'F' filler (in low nibble) - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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. – that rspCrlSupportedVersion is not present. – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - if O_D_CAT_CLASSES then catSupportedClasses is set to the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - lpaSvn is present and set to #IUT_RSP_VERSION_HIGHEST

			- euiccFormFactorType , is present and set to the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. - lpaRspCapability is present and equal to <LPA_RSP_CAPABILITY> in IC2 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, O_D_CAT_CLASSES, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not set, verify that the corresponding field in DeviceCapabilities is not present.
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK)	Next step of profile download procedure is performed (i.e. GetBoundProfilePackage request is sent).

Test Sequence #02 Nominal: Authenticate Client - response with RPM pending

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_EN is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) There is another pending RPM package order and S_SM-DP+ sends AuthenticationClient Response with rpmPending.

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1,

			#IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify that: • #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_ECDSA • <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK_VARIANT_A • operationType in #CTX_PARAMS_RPM_ICCID1 is set to #OP_TYPE_RPM • matchingIdSource in #CTX_PARAMS1_RPM_ICCID1 is set to NULL • for #DEVICE_INFO (deviceCapabilities) verify that: - TAC is BCD coded as 4 octets acc. to 3GPP TS 23.003 - if IMEI is present then it is BCD coded as 8 octets acc. to 3GPP TS 23.003 - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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 . – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - lpaSvn, the highest SGP.22 version of the LPA as defined in #IUT_RSP_VERSION_HIGHEST. - catSupportedClasses, the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - euiccFormFactorType , the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. 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, O_D_CRL, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not set, verify that the corresponding field in DeviceCapabilities is not present. • for #DEVICE_INFO (lpaRspCapabilities) verify that: - lpaRspCapabilities is set to #IUT_LPA_RSP_CAPABILITY
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_OK_AND_RPM_PENDING)	Next step of RPM Package handling procedure is performed (i.e. ES9+.HandleNotification is sent). LPA shall initiate another Mutual Authentication with same S_SM-DP+ after completing to first RPM download is completed.

4.4.23.2.4 TC_LPAd_**AuthenticateClient_ErrorCases_V3**

TBD

4.4.24 ES9+ (LPA – SM-DP+): HandleNotification

4.4.24.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]

Requirements

- Section 3.0.1, 3.1.1
- Section 5.6.4
- Section 6.5.1, 6.5.2

4.4.24.2 Test Cases

4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal

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

General Initial Conditions	
Entity	**Description of the general initial condition**
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).
S_SM-DP+	S_SM-DP+(1) is configured with #TEST_DP_ADDRESS1 and #CERT_S_SM_DP_TLS. S_SM-DP+(2) is configured with #TEST_DP_ADDRESS2 and #CERT_S_SM_DP2_TLS.
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Nominal: Successful PIR and Install Notifications to the Same SM-DP+ Address

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1 for PROFILE_OPERATIONAL1.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	PROC_ES9+_GET_BPP		

	(s NOTE 1)		
1	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_OK)) • Verify the euiccSign <EUICC_SIGN_PIR> using the #PK_EUICC_SIG
2	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd. The LPAd MAY inform the End User of the success status indicated by the Profile Installation Result.
3	LPAd → S_SM-DP+(1)	Establish an HTTPs connection if previously closed	
4	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_INST1)) sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
5	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session. NOTE 2: The timeout SHALL start after the PIR is received. NOTE 3: In case the AddProfile initiation was combined with “Enable” (i.e. O_D_ADD_ENABLE_SEPARATED is not supported), any subsequent Enable Notification is not part of the test sequence.			

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

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1 for PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_EN instead of #METADATA_OP_PROF1.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		

IC4	PROC_ES9+_GET_BPP (s. NOTE 1)		
1	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_OK)) • Verify the euiccSign <EUICC_SIGN_PIR> using the #PK_EUICC_SIG
2	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd. The LPAd MAY inform the End User of the success status indicated by the Profile Installation Result.
3	S_EndUser → LPAd	If PROFILE_OPERATIONAL1 is not already enabled (see NOTE 3), initiate the Enable Profile operation for PROFILE_OPERATIONAL1.	PROFILE_OPERATIONAL1 is enabled
4	LPAd → S_SM-DP+(1)	Establish an HTTPs connection if previously closed	
5	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_EN1)) sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
6	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session.

NOTE 2: The timeout SHALL start after the initiation of the Enable Profile operation.

NOTE 3: PROFILE_OPERATIONAL1 is expected to be already enabled only in the case that the device supports only O_D_ADD_ENABLE_COMBINED.

Test Sequence #03 Nominal: Disable and Delete Notifications to the Same SM-DP+ Address

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	PROFILE_OPERATIONAL1 is in the Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL1	PROFILE_OPERATIONAL1 is disabled
2	LPAd → S_SM-DP+(1)	Establish an HTTPs connection if previously closed	
3	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS1)) sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT (see NOTE 1) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
4	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd
5	LPAd → S_SM-DP+(1)	Establish an HTTPs connection if previously closed	
6	S_EndUser → LPAd	Initiate the Delete Profile operation for PROFILE_OPERATIONAL1	Successful End User Intent verified PROFILE_OPERATIONAL1 is deleted
7	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1)) sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT (see NOTE 2) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
8	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd

NOTE 1: The timeout SHALL start after the initiation of the Disable Profile operation.

NOTE 2: The timeout SHALL start after the End User Intent verification.

Test Sequence #04 Nominal: Enable and Disable Notifications with Different SM-DP+ Addresses

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	PROFILE_OPERATIONAL2 is installed on the eUICC.
eUICC	PROFILE_OPERATIONAL1 is in the Enabled state.
eUICC	PROFILE_OPERATIONAL2 is in the Disabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL2	PROFILE_OPERATIONAL2 is enabled
2	LPAd → S_SM-DP+(1)	Establish an HTTPs connection if previously closed	
3	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS1)) sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
4	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd
5	LPAd → S_SM-DP+(2)	Establish an HTTPs connection	
6	LPAd → S_SM-DP+(2)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS2, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_EN2)) sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
7	S_SM-DP+(2) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd
NOTE 1: Steps 2,3 and 4 can be executed in parallel to the steps 5, 6 and 7. NOTE 2: The timeout SHALL start after the initiation of the Enable Profile operation.			

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

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1 for PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_INST_DIFF instead of #METADATA_OP_PROF1.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	PROC_ES9+_GET_BPP(s. NOTE 1)		
1	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_OK))
2	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd. The LPAd MAY inform the End User of the success status indicated by the Profile Installation Result.
3	LPAd → S_SM-DP+(2)	Establish an HTTPs connection	
4	LPAd → S_SM-DP+(2)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS2, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_INST_ADDRESS2)) sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
5	S_SM-DP+(2) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session.

NOTE 2: Steps 1 and 2 can be executed in parallel to the steps 3,4 and 5.

NOTE 3: The timeout SHALL start after the End User Intent verification.

Test Sequence #06 Nominal: Profile Download with PIR Failed

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1 for PROFILE_OPERATIONAL1.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		
IC4	LPAd → S_SM-DP+(1)	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
IC5	S_SM-DP+(1) → LPAd	MTD_HTTP_RESP(#GET_BPP_INV)	No error exhibited by the LPAd, s. note 1.
1	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_SECU_INVALID)) • Verify the euiccSign <EUICC_SIGN_PIR> using the #PK_EUICC_SIG
2	S_SM-DP+(1) → LPAd	#R_HTTP_204_OK	No error exhibited by the LPAd. The LPAd MAY inform the End User of the error status indicated by the Profile Installation Result.
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session.			

Test Sequence #07 Nominal: Successful PIR and Install Notifications after Connectivity Interruption

This Test Sequence is FFS.

Test Sequence #08 Nominal: No Acknowledge for Successful PIR results in No Further Notifications

The purpose of this test case is to verify that the next Notification of a group is not sent until LPA receives a successful response from the SM-DP+ for the previous Notification.

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1 for PROFILE_OPERATIONAL1.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
IC4	PROC_ES9+_GET_BPP (s. NOTE 1)		
1	LPAd → S_SM-DP+(1)	Send ES9+.HandleNotification method initiated	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_OK))
2	LPAd → S_SM-DP+(1)	No ES9+.HandleNotification method sent	No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT OR TLS Session closed independent of timeout.
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL confirm End User Intent if requested and SHALL not abort the session. NOTE 2: The timeout in Step 3 SHALL start after the End User Intent verification.			

4.4.25 ES9+ (LPA – SM-DP+): CancelSession

4.4.25.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 2.9.2.1, 2.9.2.4
- Section 2.10.1
- Section 3.0.1, 3.0.2, 3.1.3, 3.1.3.2, 3.1.5, 3.2.7, 3.7.2
- Section 5.6.5
- Section 6.5.2.10

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 and disabled on the eUICC.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_4.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_4 (associated with PROFILE_OPERATIONAL4).
S_SM-DP+	The S_SM-DP+ is configured to ignore the forbidden PPR during the eligibility check.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP with #METADATA_OP_PROF4 used in #GET_BPP_OK This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method)		
1	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_PPR_NOT_ALLOWED)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT. OR If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #02 Nominal: End User rejection

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
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.
2	S_EndUser → LPAd	End User Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_REJ)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT. OR If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #03 Nominal: Load BPP Error

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_LOAD_ERROR)	Continue to step 2 (End User Confirmation) if requested, otherwise continue with Step 3
2	LPAd → S_EndUser	Request for Confirmation if not requested before.	End User Intent successfully verified.
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_LOAD_BPP_ERROR)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #04 Nominal: End User Timeout

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method)		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation of the Profile Download.
2	S_EndUser → LPAd	No End User Rejection or Confirmation is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_TIMEOUT)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT. OR If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

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

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		

IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_LOAD_ERROR_UNKNOWN_TAG)	Continue to step 2 (End User Confirmation) if requested, otherwise continue with Step 3
2	LPAd → S_EndUser	Request for Confirmation if not requested before.	End User Intent successfully verified.
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_LOAD_BPP_ERROR)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

4.4.25.2.2 TC_LPAd_ES9+_CancelSession_EndUserPostponed_Nominal

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Nominal: End User Postponed

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
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.
2	S_EndUser → LPAd	End User Postpone is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT. OR If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

4.4.25.2.3 TC_LPAd_ES9+_CancelSession_Error

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Error: Unknown TransactionID after End User Rejection/Postpone

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		

IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID> Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method)		
IC5	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download.
IC6	S_EndUser → LPAd	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
1	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_REJ)) OR MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_10_1_3_9)	No error after receiving the HTTPs response. (See NOTE)
NOTE: The LPA MAY either stop or retry sending ES9+.CancelSession method.			

Test Sequence #02 Error: Invalid eUICC Signature after End User Rejection/Postpone

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID> Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP		

	This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method)		
IC5	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download.
IC6	S_EndUser → LPAd	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
1	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_REJ)) OR MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_6_1)	No error after receiving the HTTPs response. The LPA SHALL stop the procedure: no ES9+.CancelSession requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #03 Error: Invalid SM-DP+ OID after End User Rejection/Postpone

Initial Conditions	
Entity	**Description of the initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID> Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method)		
IC5	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download.

IC6	S_EndUser → LPAd	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
1	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_REJ)) OR MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_ERROR_8_8_3_10)	No error after receiving the HTTPs response. The LPA SHALL stop the procedure: no ES9+.CancelSession requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT..

4.4.25.2.4 TC_LPAd_ES9+_CancelSession_PPRs

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Nominal: End User rejection/postpone after PPR1 consent requested

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed and End User Consent is required for #MCC_MNC4 with gid1 and gid2 absent.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_4.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_4 (associated with PROFILE_OPERATIONAL4).

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT		

	Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP with #METADATA_OP_PROF4 used in #GET_BPP_OK This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method)		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download. Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download.
2	S_EndUser → LPAd	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_REJ)) OR MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT. OR If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #02 Nominal: End User rejection/posptone after PPR2 consent requested

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR2 is allowed and End User Consent is required for #MCC_MNC2 with gid1 and gid2 absent.

LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3_NO_CC.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL3).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP with #METADATA_OP_PROF3 used in #GET_BPP_OK This step is conditional – occurs only if ES9+.CancelSession method was not sent before (e.g. request for Confirmation was required after ES9+.AuthenticateClient method)		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download. Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download.
2	S_EndUser → LPAd	End User Postpone/Rejection (or failed confirmation) is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_REJ)) OR MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_EU_POSTPONED)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_SIG •<S_TRANSACTION_ID> is the same as in IC3
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	If Step 1 was performed directly after IC3: No ES9+.GetBoundProfilePackage requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT. OR

			If Step 1 was performed after IC4: No ES9+.HandleNotification requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

4.4.25.2.5 TC_LPAd_ES9+_CancelSession_AuthenticateClient_RPM

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 The eUICC supports RPM
LPAd	RPM operation is enabled in the LPA by the End User
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: RPM Disabled by User

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_EN is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1. RPM operation is disabled in the LPA by the End User.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
IC2	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
IC3	S_SM-DP+ → LPAd		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>,

			#R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3))
IC4	S_SM-DP+ → LPAd		LPAd initiates Cancel Session
1	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_RPM_DISABLED)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_ECDSA •<S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK_VARIANT_A
2	S_SM-DP+ → LPAd		No ES9+ Handle Notification with Load RPM Package Result is sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #02 Nominal: RPM Package download – Reject on Strong Confirmation

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_DELETE is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) with RPM Command for Delete profile.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
IC2	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1,

			#IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
IC3	S_SM-DP+ → LPAd		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3))
IC4	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_DELETE_OK)	
1	LPAd → S_EndUser	Request for User Confirmation.	LPAd initiates Confirmation Request for Strong Confirmation on the execution of RPM Command for Delete Profile,
2	S_EndUser → LPAd	End User Rejects the RPM Command within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	LPAd → S_SM-DP+	Send ES9+.CancelSession method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_RPM_EU_REJECT)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_ECDSA •<S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK_VARIANT_A
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#R_SUCCESS)	No ES9+ Handle Notification with Load RPM Package Result is sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

Test Sequence #03 Nominal: RPM Package download – Reject on Confirmation – Disable Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_DISABLE is loaded and enabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.

LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) with RPM Command for Disable profile.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
IC2	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
IC3	S_SM-DP+ → LPAd		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3))
IC4	S_SM-DP+ → LPAd		
1	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation on the execution of RPM Command for Disable Profile,
2	S_EndUser → LPAd		
3	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_OK_RPM_EU_REJECT)) Verify: •<EUICC_CANCEL_SESSION_SIGNATURE> with the #PK_EUICC_ECDSA •<S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK_VARIANT_A

4	S_SM-DP+ → LPAd		No ES9+ Handle Notification with Load RPM Package Result is sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT.

4.4.26 ES9+ (LPA – SM-DP+): HTTPS

4.4.26.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 2.1
- Section 2.6.6, 2.6.7.1
- Section 3.0.1
- Section 5.6
- Section 6
- Section 6.1

4.4.26.2 Test Cases

4.4.26.2.1 TC_LPAd_HTTPS_Nominal_Variant_O

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.

Test Sequence #01 Nominal: HTTPS Session Establishment

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>) Verify the following: • #IUT_TLS_VERSION SHALL be 1.2 or higher • <TLS_CIPHER_SUITES> SHALL contain at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and NOT contain 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).
2	S_SM-DP+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS, NO_PARAM)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)
3	S_SM-DP+ → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

Test Sequence #02 Nominal: non-reuse of session keys

The purpose of this test sequence is to verify that the LPAd is not reusing ephemeral keys from the previous session.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ Extract <CLIENT_TLS_EPHEM_KEY> Extract <SESSION_ID_CLIENT> and <S_SESSION_ID_SERVER>		
IC2	Terminate TLS session and restart “Add Profile” Procedure as define in the initial conditions.		
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLEINT>, <EXT_SHA256_ECDSA>) Verify the following: • #IUT_TLS_VERSION SHALL be 1.2 or higher • <TLS_CIPHER_SUITES> SHALL be at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and NOT contain 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+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS, NO_PARAM)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>) Verify if • <CLIENT_TLS_EPHEM_KEY> is different from the one used by LPAd in IC1
3	S_SM-DP+ → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

4.4.26.2.2 TC_LPAd_HTTPS_ErrorCases

General Initial Conditions	
Entity	**Description of the general initial condition**
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.

Test Sequence #01 Error: Invalid (SM-DP+) TLS Certificate signature

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SM-DP+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_SIG, NO_PARAM) Note: if the LPAd sends an Alert during or after any of the messages sent by the S_SM-DP+ in MTD_TLS_SERVER_HELLO_ETC, then the S_SM-DP+ might not send the messages specified in MTD_TLS_SERVER_HELLO_ETC which occur after the Alert.	LPAd MAY send a TLS Alert. LPAd aborts AddProfile procedure.
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected.

Test Sequence #02 Error: Expired TLS Certificate

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SM-DP+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_EXPIRED, NO_PARAM) Note: if the LPAd sends an Alert during or after any of the messages sent by the S_SM-DP+ in MTD_TLS_SERVER_HELLO_ETC, then the S_SM-DP+ might not send the messages specified in	LPAd MAY send a TLS Alert. LPAd aborts AddProfile procedure

		MTD_TLS_SERVER_HELLO_ETC which occur after the Alert.	
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected.

Test Sequence #03 Error: VOID

Test Sequence #04 Error: VOID

Test Sequence #05 Error: VOID

Test Sequence #06 Error: VOID

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

Step	Direction	Sequence / Description	Expected result
IC1	Power-on the Device		
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SM-DP+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS_INV_CURVE, NO_PARAM) Note: if the LPAd sends an Alert during or after any of the messages sent by the S_SM-DP+ in MTD_TLS_SERVER_HELLO_ETC, then the S_SM-DP+ might not send the messages specified in MTD_TLS_SERVER_HELLO_ETC which occur after the Alert.	LPAd MAY send a TLS Alert. LPAd aborts AddProfile procedure
3	LPDd → S_SM-DP+	TLS 1.2 close	The TLS connection is rejected.

4.4.26.2.3 TC_LPAd_HTTPS_Nominal_Variants_V3

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	There is no default SM-DP+ address configured.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.

Test Sequence #01 Nominal: HTTPS Session Establishment Variant A

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>) Verify the following: • #IUT_TLS_VERSION SHALL be 1.2 or higher • <TLS_CIPHER_SUITES> SHALL contain at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and NOT contain 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).
2	S_SM-DP+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS, #CERT_S_SM_DP_SubCA_ECDSA)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)
3	S_SM-DP+ → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

Test Sequence #02 Nominal: HTTPS Session Establishment Variant B

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>) Verify the following: • #IUT_TLS_VERSION SHALL be 1.2 or higher • <TLS_CIPHER_SUITES> SHALL contain at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and NOT contain 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).

2	S_SM-DP+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS, #CERT_S_CI_SubCA_SIG)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)
3	S_SM-DP+ → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

Test Sequence #03 Nominal: HTTPS Session Establishment Variant C

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>) Verify the following: • #IUT_TLS_VERSION SHALL be 1.2 or higher • <TLS_CIPHER_SUITES> SHALL contain at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and NOT contain 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).
2	S_SM-DP+ → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DP_TLS, #CERT_S_SM_DP_SubCAList_SIG)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)
3	S_SM-DP+ → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

4.4.27 ES11 (LPA – SM-DS): InitiateAuthentication

4.4.27.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.0.1, 3.1.3

- Section 5.8.1
- Section 6.5.2.11
- Section 6.6.2.6

4.4.27.2 Test Cases

4.4.27.2.1 TC_LPAd_ES11_InitiateAuthentication_Nominal

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
Device	The Profile Download is initiated using SM-DS (see section 2.2.4.1).
S_SM-DS	S_SM-DP+ (#TEST_DP_ADDRESS1) performed Profile download Event Registration to the S_SM-DS (#TEST_ROOT_DS_ADDRESS) with #EVENT_ID_1.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Nominal: Initiate Authentication

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
1	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS) , <LPA_RSP_CAPABILITY>) • Extract <EUICC_CHALLENGE> Verify if: • <LPA_RSP_CAPABILITY> contains • `crlStaplingV3Support` set to '1' • `certChainV3Support` set to '1' • `signedSmdsResponseV3Support` set to '1'
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK)	Next step of common mutual authentication procedure is performed (i.e. AuthenticateClient request is sent).

4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
Device	The Profile Download is initiated using SM-DS (see section 2.2.4.1).

S_SM-DS	S_SM-DP+ (#TEST_DP_ADDRESS1) performed Profile download Event Registration to the S_SM-DS (#TEST_ROOT_DS_ADDRESS) with #EVENT_ID_1 (see NOTE).
eUICC	There is no default SM-DP+ address configured.
NOTE: The S_SM_DP+ does not need to be available to the LPAd for profile download during test sequence execution, as the LPAd is not expected to receive the smdpAddress.	

Test Sequence #01 Error: Invalid SM-DS Address

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_8_9_1_3_8)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #02 Error: Unsupported Security Configuration

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_8_9_2_3_1)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #03 Error: Unsupported SVN

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		

IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR _8_9_3_3_1)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #04 Error: Unavailable SM-DS Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_ 8_9_4_3_7)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #05 Error: Invalid SM-DS Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_CERT_D S)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent

			within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #06 Error: Invalid SM-DS Signature

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_SIGN_DS)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #07 Error: Invalid SM-DS Address sent by the SM-DS

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_SMDS_ADDRESS)	LPAd informs the S_EndUser and aborts the AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

Test Sequence #08 Error: Unsupported CI Key ID

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		

IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_INV_CI_DS)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No ES11.InitiateAuthentication or ES11.AuthenticateClient requests are sent within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

4.4.28 ES11 (LPA – SM-DS): AuthenticateClient

4.4.28.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.0.1, 3.1.3, 3.1.3.2
- Section 4.2
- Section 5.8.2
- Section 6.2
- Section 6.3
- Section 6.5.1, 6.5.1.1, 6.5.1.2, 6.5.1.3, 6.5.1.4, 6.5.2, 6.5.2.12
- Section 6.6.2.7

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).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK)	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO)) Verify: • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_DS_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • if matchingId field in #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO is missing OR matchingId field is present and <MATCHING_ID> is empty • if <S_SMDS_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO is the same as in <S_SMDS_SIGNED1> present in #INITIATE_AUTH_DS_OK • for #DEVICE_INFO: - The value of the TAC corresponds to the first 8 digits of #IUT_IMEI and is represented as a string of 4 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS 29.002 [26] and 3GPP TS 23.003 [12] - if IMEI is present then its value corresponds to #IUT_IMEI and is represented as a string of 8 octets that is coded as a Telephony Binary Coded Decimal String as defined in 3GPP TS

			29.002 [26] and 3GPP TS 23.003 [12] except that the last octet contains the check digit (in high nibble) and an 'F' filler (in low nibble) - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease (or eutranEpcSupportedRelease if #IUT_RSP_VERSION is v2.2.2 or higher) is set to the highest release as defined in #IUT_LTE_EUTRAN_REL. – if O_D_NFC_TS26 then contactlessSupportedRelease is set to the highest release as defined in #IUT_NFC_REL. – that rspCrlSupportedVersion is not present , – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - if O_D_CAT_CLASSES then catSupportedClasses is set to the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - lpaSvn is present and set to #IUT_RSP_VERSION_HIGHEST - euiccFormFactorType is present and set to the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. - lpaRspCapability is present and equal to <LPA_RSP_CAPABILITY> in IC2 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, O_D_CAT_CLASSES, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC, if the option is not set, verify that the corresponding field in DeviceCapabilities is not present.
2	S_SM-DS → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_DS_OK1)	Next step of profile discovery procedure is performed (ie. ES9+.InitiateAuthentication request is sent)

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).

Step	Direction	Sequence/ Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS(1)	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>
1	S_SM-DS(1) → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK)	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO)) Verify: • if matchingId field in #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO is missing OR matchingId field is present and <MATCHING_ID> is empty
2	S_SM-DS(1) → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_DS_OK_DS ADDR1)	

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, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>
5	S_SM-DS(2) → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK_1)	MTD_HTTP_REQ(#TEST_DS_ADDRESS1 , #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO_1)) Verify: • if <MATCHING_ID> is set to #EVENT_ID_1
6	S_SM-DS(2) → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_DS_OK2)	Next step of profile discovery procedure is performed (ie. ES9+.InitiateAuthentication request is sent)

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: VOID

Test Sequence #02 Error: VOID

Test Sequence #03 Error: Invalid eUICC or EUM Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	PROC_ES11_INIT_AUTH		
1	LPAd → S_SM-DS	Send ES11.AuthenticateClient method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT,

			MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO))
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_3_6_1)	LPAd aborts AddProfile procedure
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.

Test Sequence #04 Error: Expired eUICC or EUM Certificate

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	PROC_ES11_INIT_AUTH		
1	LPAd → S_SM-DS	Send ES11.AuthenticateClient method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO))
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_3_6_3)	LPAd aborts AddProfile procedure
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.

Test Sequence #05 Error: Invalid eUICC signature or serverChallenge

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	PROC_ES11_INIT_AUTH		
1	LPAd → S_SM-DS	Send ES11.AuthenticateClient method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO))
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_8_1_6_1)	LPAd aborts AddProfile procedure
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.

Test Sequence #06 Error: Unknown TransactionID

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	PROC_ES11_INIT_AUTH		
1	LPAd → S_SM-DS	Send ES11.AuthenticateClient method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO))
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_8_10_1_3_9)	LPAd aborts AddProfile procedure
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.

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
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
IC2	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>
IC3	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK)	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO))
IC4	S_SM-DS → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_DS_OK_DS ADDR1)	No Error
IC5	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11 with #TEST_DS_ADDRESS1 and #CERT_S_SM_DS2_TLS		

IC6	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DS_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DS_ADDRESS1, <LPA_RSP_CAPABILITY>)) • Extract <EUICC_CHALLENGE>
IC7	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK_1)	MTD_HTTP_REQ(#TEST_DS_ADDRESS1 , #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO_1))
1	S_SM-DS → LPAd	MTD_HTTP_RESP(#R_ERROR_8_9_5_3_9)	LPAd aborts AddProfile procedure
2	LPAd → S_SM-DS	No Profile download action	No requests are sent on ES11 within the timeout #IUT_LPAd_SESSION_CLOSE_TIMEOUT in Annex F.

4.4.29 ES11 (LPA -- SM-DS): HTTPS

4.4.29.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 2.6.6, 2.6.7.1
- Section 4.5.2.2
- Section 5.8
- Section 6
- Section 6.1

4.4.29.2 Test Cases

4.4.29.2.1 TC_LPAd_ES11_HTTPS_Nominal_Variant_O

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
1	LPAd → S_SM-DS	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>) Verify the following: • #IUT_TLS_VERSION SHALL be 1.2 or higher • <TLS_CIPHER_SUITES> SHALL contain at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and NOT contain 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).
2	S_SM-DS → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS, NO_PARAM)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)
3	S_SM-DS → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

Test Sequence #02 Nominal: non-reuse of session keys

The purpose of this test sequence is to verify that the LPAd is not reusing ephemeral keys from the previous session.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11 Extract <CLIENT_TLS_EPHEM_KEY> Extract <SESSION_ID_CLIENT> and <S_SESSION_ID_SERVER>		
IC2	Terminate TLS session and Initiate Profile Download using SM-DS (see section 2.2.4.1).		
1	LPAd → S_SM-DS	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>) Verify the following: • #IUT_TLS_VERSION SHALL be 1.2 or higher • <TLS_CIPHER_SUITES> SHALL be at least TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 and NOT contain 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-DS → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS, NO_PARAM)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>) Verify if • <CLIENT_TLS_EPHEM_KEY> is different from the one used by LPAd in IC1
3	S_SM-DS → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

4.4.29.2.2 TC_LPAd_ES11_HTTPS_Error

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
Device	The Profile Download is initiated using SM-DS (see section 2.2.4.1).
S_SM-DS	S_SM-DP+ (#TEST_DP_ADDRESS1) performed Profile download Event Registration to the S_SM-DS (#TEST_ROOT_DS_ADDRESS) with #EVENT_ID_1.
eUICC	There is no default SM-DP+ address configured.

Test Sequence #01 Error: Invalid (SM-DS) TLS Certificate signature

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DS	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SM-DS → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_SIG, NO_PARAM) Note: if the LPAd sends an Alert during or after any of the messages sent by the S_SM-DP+ in MTD_TLS_SERVER_HELLO_ETC, then the S_SM-DP+ might not send the messages specified in MTD_TLS_SERVER_HELLO_ETC which occur after the Alert.	LPAd MAY send a TLS Alert. LPAd aborts AddProfile procedure
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected.

Test Sequence #02 Error: Expired TLS Certificate

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DS	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SM-DS → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_EXPIRED, NO_PARAM) Note: if the LPAd sends an Alert during or after any of the messages sent by the S_SM-DP+ in MTD_TLS_SERVER_HELLO_ETC, then the S_SM-DP+ might not send the messages specified in MTD_TLS_SERVER_HELLO_ETC which occur after the Alert.	LpaD may send a TLS Alert. LPAd aborts AddProfile procedure
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected.

Test Sequence #03 Error: VOID

Test Sequence #04 Error: VOID

Test Sequence #05 Error: VOID

Test Sequence #06 Error: VOID

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

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DS	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SM-DS → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SM_DS_TLS_INV_CURVE, NO_PARAM) Note: if the LPAd sends an Alert during or after any of the messages sent by the S_SM-DP+ in MTD_TLS_SERVER_HELLO_ETC, then the S_SM-DP+ might not send the	LPAd MAY send a TLS Alert. LPAd aborts AddProfile procedure

		messages specified in MTD_TLS_SERVER_HELLO_ETC which occur after the Alert.	
3	LPDd → S_SM-DS	TLS 1.2 close	The TLS connection is rejected.

4.4.30 ES9+ (LPA -- SM-DP+): AuthenticateClient – RPM Package Download

4.4.30.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]

Requirements

- 2.6.6.2, 2.10.1
- 3.0.1
- 3.2.7, 3.7.2
- 3.7.2
- 5.6.3

4.4.30.2 Test Cases

4.4.30.2.1 TC_LPAd_AuthenticateClient_Nominal_RPM

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 The eUICC supports RPM
LPAd	RPM operation is enabled in the LPA by the End User
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: Authenticate Client Variant A for RPM, Polling Address: SM-DP+

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_EN is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected.

	Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+	
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify that: • #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_ECDSA • <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK_VARIANT_A • operationType in #CTX_PARAMS_RPM_ICCID1 is set to #OP_TYPE_RPM • matchingIdSource in #CTX_PARAMS1_RPM_ICCID1 is set to NULL • for #DEVICE_INFO (deviceCapabilities) verify that: - TAC is BCD coded as 4 octets acc. to 3GPP TS 23.003 - if IMEI is present then it is BCD coded as 8 octets acc. to 3GPP TS 23.003 - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then utranSupportedRelease is set to the

			highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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 . – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - lpaSvn is present. - catSupportedClasses, the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - euiccFormFactorType , the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. 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, O_D_CRL, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not

			set, verify that the corresponding field in DeviceCapabilities is not present. • for #DEVICE_INFO (lpaRspCapabilities) verify that: - lpaRspCapabilities is set to #IUT_LPA_RSP_CAPABILITY
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_OK)	No Error LPAd initiates Simple Confirmation. S_EndUser accepts RPM Command.

Test Sequence #02 Nominal: Authenticate Client Variant B for RPM

General Initial Conditions	
Entity	Description of the initial condition
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_EN is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+	
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_B)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify: • if #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • if <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK_VARIANT_B • if operationType in #CTX_PARAMS_RPM_ICCID1 is set to #OP_TYPE_RPM • if matchingIdSource in #CTX_PARAMS1_RPM_ICCID1 is set to NULL • for #DEVICE_INFO (deviceCapabilities) verify: - TAC is BCD coded as 4 octets acc. to 3GPP TS 23.003 - if IMEI is present then it is BCD coded as 8 octets acc. to 3GPP TS 23.003 - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then

			utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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 . – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - lpaSvn is present. - catSupportedClasses, the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - euiccFormFactorType , the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. 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, O_D_CRL, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not

			set, verify that the corresponding field in DeviceCapabilities is not present. • for #DEVICE_INFO (lpaRspCapabilities) verify: - lpaRspCapabilities is set to #IUT_LPA_RSP_CAPABILITY
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_OK)	No Error LPAd initiates Simple Confirmation. S_EndUser accepts RPM Command.

Test Sequence #03 Nominal: Authenticate Client Variant C for RPM

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_EN is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+	
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_C)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify: • if #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • if <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK_VARIANT_C • if operationType in #CTX_PARAMS_RPM_ICCID1 is set to #OP_TYPE_RPM • if matchingIdSource in #CTX_PARAMS1_RPM_ICCID1 is set to NULL • for #DEVICE_INFO (deviceCapabilities) verify: - TAC is BCD coded as 4 octets acc. to 3GPP TS 23.003 - if IMEI is present then it is BCD coded as 8 octets acc. to 3GPP TS 23.003 - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then

			utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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 . – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - lpaSvn is present. - catSupportedClasses, the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - euiccFormFactorType , the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. 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, O_D_CRL, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not

			set, verify that the corresponding field in DeviceCapabilities is not present. • for #DEVICE_INFO (lpaRspCapabilities) verify: - lpaRspCapabilities is set to #IUT_LPA_RSP_CAPABILITY
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_OK)	No Error LPAd initiates Simple Confirmation. S_EndUser accepts RPM Command.

Test Sequence #04 Nominal: Authenticate Client Variant A for RPM, with PPR

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_PPR_RPM_CONF_EN is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+	
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify: • if #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • if <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK_VARIANT_A • if operationType in #CTX_PARAMS_RPM_ICCID1 is set to #OP_TYPE_RPM • if matchingIdSource in #CTX_PARAMS1_RPM_ICCID1 is set to NULL • for #DEVICE_INFO (deviceCapabilities) verify: - TAC is BCD coded as 4 octets acc. to 3GPP TS 23.003 - if IMEI is present then it is BCD coded as 8 octets acc. to 3GPP TS 23.003 - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then

			utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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 . – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - lpaSvn, the highest SGP.22 version of the LPA as defined in #LPA_RSP_VERSION_HIGHEST. - catSupportedClasses, the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - euiccFormFactorType , the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. 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, O_D_CRL,

			O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not set, verify that the corresponding field in DeviceCapabilities is not present. • for #DEVICE_INFO (lpaRspCapabilities) verify: - lpaRspCapabilities is set to #IUT_LPA_RSP_CAPABILITY
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_OK)	No Error • LPAd initiates Simple Confirmation • LPAd may display relevant information concerning the Profile and PPR(s) to the S_EndUser and requests the User Consent. It may be combined with Simple Confirmation. • S_EndUser approves RPM Command.

Test Sequence #05 Nominal: Authenticate Client Variant A for RPM – PPR Update

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_UPDATE_MD_PPR is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sent by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) with RPM Command Update Metadata for removing PPRs.

Step	Direction	Sequence / Description	Expected result
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+	
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify that: • #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_ECDSA • <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK_VARIANT_A • operationType in #CTX_PARAMS_RPM_ICCID1 is set to #OP_TYPE_RPM • matchingIdSource in #CTX_PARAMS1_RPM_ICCID1 is set to NULL • for #DEVICE_INFO (deviceCapabilities) verify that: - TAC is BCD coded as 4 octets acc. to 3GPP TS 23.003 - if IMEI is present then it is BCD coded as 8 octets acc. to 3GPP TS 23.003 - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then utranSupportedRelease is set to the

			highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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 . – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - lpaSvn is present - catSupportedClasses, the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - euiccFormFactorType , the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. 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, O_D_CRL, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not

			set, verify that the corresponding field in DeviceCapabilities is not present. • for #DEVICE_INFO (lpaRspCapabilities) verify that: - lpaRspCapabilities is set to #IUT_LPA_RSP_CAPABILITY
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_UM_PPR_OK)	LPA may optionally initiate User Consent. and S_EndUser approves RPM Command. Next step of RPM Package handling procedure is performed (i.e. ES9+.HandleNotification is sent).

4.4.30.2.2 TC_LPAd_AuthenticateClient_Nominal_RPM_Enterprise Device

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 The eUICC supports RPM and Enterprise profiles
LPAd	Variant A certificates are included in certificates chain for TLS procedures. RPM operation is enabled in the LPA by the End User

Test Sequence #01 Nominal: Authenticate Client Variant A for RPM, Update Reference Enterprise Rule

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_UM_ENT_RULES is loaded and disabled. eUICC returns CTX_PARAMS1_RPM_ICCID1 (sends by the LPAd) in euiccSigned1 in the response to ES10.AuthenticateServer.
LPAd	Update Profile operation is initiated and PROFILE_OPERATIONAL1 is selected. Polling Address is retrieved from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Update Reference Enterprise Rules

Step	Direction	Sequence / Description	Expected result
IC1		PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+	
IC2	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1, #IUT_LPA_RSP_CAPABILITY)) • Extract <EUICC_CHALLENGE>
1	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Verify: • if #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 used with the #MATCHING_ID_1 • If <S_TRANSACTION_ID> is the same as in #INITIATE_AUTH_OK • <EUICC_SIGNATURE1> using the #PK_EUICC_SIG • if <S_SMDP_CHALLENGE> present in the #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 is the same as in <S_SMDP_SIGNED1> present in #INITIATE_AUTH_OK_VARIANT_A • if operationType in #CTX_PARAMS_RPM_ICCID1 is set to #OP_TYPE_RPM • if matchingIdSource in #CTX_PARAMS1_RPM_ICCID1 is set to NULL • for #DEVICE_INFO (deviceCapabilities) verify: - TAC is BCD coded as 4 octets acc. to 3GPP TS 23.003 - if IMEI is present then it is BCD coded as 8 octets acc. to 3GPP TS 23.003 - if O_D_GSM_GERAN then gsmSupportedRelease is set to the highest release as defined in #IUT_GSM_GERAN_REL. – if O_D_UMTS_UTRAN then

			utranSupportedRelease is set to the highest release as defined in #IUT_UMTS_UTRAN_REL. – if O_D_CDMA2000_1X then cdma2000onexSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_1X_REL. – if O_D_CDMA2000_HRPD then cdma2000hrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_HRPD_REL. The value R is either 1, 2 or 3 for Rev 0, A or B respectively. – if O_D_CDMA2000_EHRPD then cdma2000ehrpdSupportedRelease is set to the highest release as defined in #IUT_CDMA2000_EHRPD_REL. – if O_D_LTE then eutranSupportedRelease 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 . – if O_D_NR_EPC then nrEpcSupportedRelease is set to the highest release as defined in #IUT_NR_EPC_REL, – if O_D_NR_5GC then nr5gcSupportedRelease is set to the highest release as defined in #IUT_NR_5GC_REL, – if O_D_EUTRAN_5GC then eutran5gcSupportedRelease is set to the highest release as defined in #IUT_EUTRAN_5GC_REL, - lpaSvn, the highest SGP.22 version of the LPA as defined in #LPA_RSP_VERSION_HIGHEST. - catSupportedClasses, the set of supported Card Application Toolkit letter classes as defined in #IUT_CAT_CLASSES. - euiccFormFactorType , the eUICC form factor as defined in #IUT_EUICC_FORM_FACTOR. 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, O_D_CRL, O_D_NR_EPC, O_D_NR_5GC or O_D_EUTRAN_5GC if the option is not set, verify that the corresponding field in DeviceCapabilities is not present.

			• for #DEVICE_INFO (lpaRspCapabilities) verify: - lpaRspCapabilities is set to #IUT_LPA_RSP_CAPABILITY
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_OK)	No Error • LPAd initiates Simple Confirmation • LPAd may display relevant information concerning the Enterprise Rules to the S_EndUser and requests the User Consent. It may be combined with Simple Confirmation. • S_EndUser approves RPM Command.

4.4.32 ES9+ (LPA – SM-DP+): HandleNotification for RPM Package Download

4.4.32.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- 2.6.6.2, 2.10.1
- 3.0.1
- 3.2.7, 3.7.2
- 3.7.2
- 5.5.3, 5.6.3, 5.7.14a
- 6.6.2.4

4.4.32.2 Test Cases

4.4.32.2.1 TC_LPAd_RPM_ HandleNotification for RPM Package Download

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
Device	RPM operation is enabled in the LPA by the End User
eUICC	There is no default SM-DP+ address configured The eUICC supports RPM
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: Successful RPR and Other Notification for RPM Enable operation to the Same SM-DP+ Address

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_NOTIF_CONF is installed and disabled.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Update Profile operation	LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(enable, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser	Request for User Confirmation.	LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser	Accepts the RPM command	PROFILE_OPERATIONAL1 is enabled
7	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(enableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)))
8	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error
9	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_RPM_EN1))

10	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error

Test Sequence #02 Nominal: Successful RPR and Other Notifications to different SM-DP+ Address (implicit disabling of the formerly enabled Profile)

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_NOTIF_CONF2 is installed and disabled.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1_AND_NOTIF_CONF (without PPR1 present) is loaded and Enabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Update Profile operation	LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(enable, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser	Request for User Confirmation.	LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser	Accepts the RPM command	PROFILE_OPERATIONAL1 is enabled
7	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS2, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_IMPLICIT_DIS2))
8	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error
9	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(enableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)

			))
10	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error
11	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS2, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_RPM_EN2))
12	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error

4.5 VOID

4.6 VOID

4.7 VOID

5 Procedure - Behaviour Testing

5.1 General Overview

5.2 VOID

5.3 VOID

5.4 Device Procedures

5.4.1 Local Profile Management - Add Profile

5.4.1.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 2.9.2.1
- Section 3.0.1, 3.1.3, 3.1.3.2
- Section 3.2, 3.2.5
- Section 4.1, 4.4
- Annex C.1, C.3

5.4.1.2 Test Cases

5.4.1.2.1 TC_LPAd_AddProfile_Manual_Entry

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

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (manual entry)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate Add Profile operation	LPAd requests the Activation Code from the End User
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1 by manual entry	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL1 is displayed in Disabled state

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
1	S_EndUser→ LPAd	Initiate Add Profile operation	LPAd requests the Activation Code from the S_End User
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_3 by manual entry	No error
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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified, if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction. NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

5.4.1.2.2 TC_LPAd_AddProfile_QRcode_scanning

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

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (QR code scanning)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).

eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate Add Profile operation	LPAd requests the Activation Code from the End User
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1 by scanning the QR code	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser→ LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.			

5.4.1.2.3 TC_LPAd_AddProfile_ActivationCode_InvalidFormat_QRcode

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	The PROFILE_OPERATIONAL1 is not installed on the eUICC.

Test Sequence #01 Error: Add a new Operational Profile by using wrongly formatted Activation Code (QR code scanning)

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Add Profile operation	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_INVALID_FORMAT by scanning the QR code	LPAd provides an error message to the EndUser

3	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is not displayed

5.4.1.2.4 TC_LPAd_AddProfile_ActivationCode_InvalidFormat_ManualEntry

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

Test Sequence #01 Error: Add a new Operational Profile by using wrongly formatted Activation Code (Manual entry)

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Add Profile operation	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_INVALID_FORMAT by manual entry	LPAd provides an error message to the EndUser
3	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is not displayed

5.4.1.2.5 TC_LPAd_AddProfile_ConfirmationCode_smdpSigned2_QR

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

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (QR code scanning) with confirmation code indicated only in smdpSigned2

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) which requires confirmation code.
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate Add Profile operation	Activation Code is requested from the End User by LPAd

2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1 by scanning the QR code	No error
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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified, if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser→ LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.

5.4.1.2.6 TC_LPAd_AddProfile_ConfirmationCode_smdpSigned2_Manual_Entry

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

Test Sequence #01 Nominal: Add a new Operational Profile by using Activation Code (manual entry) with confirmation code indicated only in smdpSigned2

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) which requires confirmation code.
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Add Profile operation	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_1 by manual entry	No error

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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified, if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.

5.4.1.2.7 TC_LPAd_AddProfile_default_SM-DP+_address

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

Test Sequence #01 Nominal: Add a new Operational Profile by using the default SM-DP+ Address

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for PROFILE_OPERATIONAL1 linked to the EID of the eUICC.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Add Profile operation See NOTE1	No error
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
3	PROC_ES9+_INIT_AUTH		

4	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID> or missing MatchingID data object		
5	PROC_ES9+_GET_BPP (see NOTE 2)		
6	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
7	PROC_ES9+_HANDLE_NOTIF		
8	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state

NOTE 1: The Profile download by default SM-DP+ address MAY be implemented in different ways (e.g. some Device MAY implement a separate LUI menu for this function, some Device MAY request first the activation code, etc.). In order to enforce that the default SM-DP+ address is used the user SHALL not enter the Activation Code in case it is requested.

NOTE 2: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

5.4.1.2.8 TC_LPAd_AddProfile _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.
1	S_EndUser → LPAd	Initiate Add Profile operation	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide#ACTIVATION_CODE_3 by scanning the QR code	
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.

7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified, if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction. NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

Test Sequence #02 Nominal: Add a new Operational Profile by using Activation Code (manual) 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.
1	S_EndUser → LPAd	Initiate Add Profile operation	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide#ACTIVATION_CODE_3 by manual entry	
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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	For LPAd supporting SGP.22 v2.2.2 or earlier: Request for Confirmation, if not requested before. (see NOTE 2)	For LPAd supporting SGP.22 v2.2.2 or earlier: End User Intent successfully verified, if not verified before. (see NOTE 2)

9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction. NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

5.4.1.2.9 TC_LPAd_AddProfile_PPRs

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

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed and End User Consent is required for #MCC_MNC4 with gid1 and gid2 absent.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_4.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_4 (associated with PROFILE_OPERATIONAL4).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP with #METADATA_OP_PROF4 used in #GET_BPP_OK		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure. Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download. (See NOTE)
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
NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt.			

Test Sequence #02 Nominal: End User Confirmation after PPR2 consent requested

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR2 is allowed and End User Consent is required for #MCC_MNC2 with gid1 and gid2 absent.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_3_NO_CC.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL3).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP with #METADATA_OP_PROF3 used in #GET_BPP_OK		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download. (See NOTE)
2	S_EndUser → LPAd	End User Confirmation is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	PROC_ES9+_HANDLE_NOTIF		
4	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL3 is displayed in Disabled state
NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt.			

Test Sequence #03 Nominal: Profile with PPR1 already present

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed for #MCC_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required).
eUICC	The PROFILE_OPERATIONAL4 with PPR1 is installed and enabled on the eUICC.
LPAd	Add Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download. End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.
2	S_EndUser → LPAd	End User Confirmation is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	PROC_ES9+_HANDLE_NOTIF		
4	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL1 is displayed in Disabled state

5.4.1.2.10 VOID

5.4.1.2.11 TC_LPAd_AddProfile_Security_Errors

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
eUICC	The PROFILE_OPERATIONAL1 is not installed on the eUICC.

Test Sequence #01 Error: Stop Add Profile Operation if No Confirmation Provided

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
IC1	S_EndUser→ LPAd	Initiate Add Profile operation	LPAd requests the Activation Code from the End User
IC2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1	No error
IC3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ (See NOTE 2)		
IC4	PROC_ES9+_INIT_AUTH		
IC5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
IC6	PROC_ES9+_GET_BPP (see NOTE 1)		
1	LPAd → S_EndUser	Request for Confirmation, if not requested before. The End User SHALL not provide Confirmation.	The LPAd stops the Add Profile procedure automatically or provide means to stop the procedure by the End User.
2	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL1 is not displayed
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction if there is a way to abort it in step 1. NOTE 2: Step IC6 is conditional – occurs only if Step 1 (Request for Confirmation) was not executed before.			

5.4.1.2.12 TC_LPAd_AddProfile_Empty_MatchingID

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

Test Sequence #01 Nominal: Add a new Operational Profile by using empty MatchingID (QR code entry)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.

S_SM-DP+	There is a pending Profile download order for PROFILE_OPERATIONAL1 linked to the EID of the eUICC.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Add Profile operation	LPAd requests the Activation Code from the End User
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_5 by scanning the QR code	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.			

Test Sequence #02 Nominal: Add a new Operational Profile by using empty MatchingID (manual entry)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for PROFILE_OPERATIONAL1 linked to the EID of the eUICC.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Add Profile operation	LPAd requests the Activation Code from the End User
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_5 by manual entry	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP		

	(see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.			

5.4.1.2.13 TC_LPAd_AddEnableProfile_Manual_Entry

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

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual entry)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate combined Add and Enable Profile operations	LPAd requests the Activation Code from the End User
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1 by manual entry	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified,if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL1 is displayed in Enabled state

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

Test Sequence #02 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual entry) with Confirmation Code

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (PROFILE_OPERATIONAL1) associated with #CONFIRMATION_CODE1.
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate combined Add and Enable Profile operations	LPAd requests the Activation Code from the S_End User
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_3 by manual entry	No error
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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction. NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

5.4.1.2.14 TC_LPAd_AddEnableProfile

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

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (QR code scanning)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1).
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate combined Add and Enable Profile operations	LPAd requests the Activation Code from the End User
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1 by scanning the QR code	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified,if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser→ LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.			

Test Sequence #02 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual)

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
1	S_EndUser→ LPAd	Initiate combined Add and Enable Profile operations	LPAd requests the Activation Code from the End User
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1 by manual entry	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified,if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser→ LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.			

5.4.1.2.15 TC_LPAd_AddEnableProfile_ConfirmationCode_smdpSigned2_QR

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

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (QR code scanning) with confirmation code indicated only in smdpSigned2

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) which requires confirmation code.
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate combined Add and Enable Profile operations	Activation Code is requested from the End User by LPAd
2	S_EndUser→ LPAd	Provide #ACTIVATION_CODE_1 by scanning the QR code	No error
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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser→ LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.

5.4.1.2.16 TC_LPAd_AddEnableProfile_ConfirmationCode_smdpSigned2_Manual_Entry

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

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual entry) with confirmation code indicated only in smdpSigned2

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) which requires confirmation code.
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate combined Add and Enable Profile operations	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_1 by manual entry	No error
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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified, if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.

5.4.1.2.17 TC_LPAd_AddEnableProfile_default_SM-DP+_address

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

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using the default SM-DP+ Address

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for PROFILE_OPERATIONAL1 linked to the EID of the eUICC.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate combined Add and Enable Profile operations See NOTE1	No error
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
3	PROC_ES9+_INIT_AUTH		
4	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID> or missing MatchingID data object		
5	PROC_ES9+_GET_BPP (see NOTE 2)		
6	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
7	PROC_ES9+_HANDLE_NOTIF		
8	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enable state

NOTE 1: The Profile download by default SM-DP+ address MAY be implemented in different ways (e.g. some Device MAY implement a separate LUI menu for this function, some Device MAY request first the activation code, etc.). In order to enforce that the default SM-DP+ address is used the user SHALL not enter the Activation Code in case it is requested.

NOTE 2: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

5.4.1.2.18 TC_LPAd_AddEnableProfile_with_ConfirmationCode

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

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using Activation Code (QR code scanning) with confirmation code

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (PROFILE_OPERATIONAL1).
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result.
1	S_EndUser → LPAd	Initiate combined Add and Enable Profile operations	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide#ACTIVATION_CODE_3 by scanning the QR code	

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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before. (see NOTE 2)	End User Intent successfully verified,if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction. NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

Test Sequence #02 Nominal: Add and Enable a new Operational Profile by using Activation Code (manual) 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.
1	S_EndUser → LPAd	Initiate combined Add and Enable Profile operations	Activation Code is requested from the End User by LPAd
2	S_EndUser → LPAd	Provide#ACTIVATION_CODE_3 by manual entry	
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.
7	PROC_ES9+_GET_BPP_CC (see NOTE 1)		

8	LPAd → S_EndUser	For LPAd supporting SGP.22 v2.2.2 or earlier: Request for Confirmation, if not requested before. (see NOTE 2)	For LPAd supporting SGP.22 v2.2.2 or earlier: End User Intent successfully verified,if not verified before. (see NOTE 2)
9	PROC_ES9+_HANDLE_NOTIF		
10	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction. NOTE 2: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

5.4.1.2.19 TC_LPAd_AddEnableProfile_PPRs

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

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed and End User Consent is required for #MCC_MNC4 with gid1 and gid2 absent.
LPAd	Add and Enable Profile operation is initiated by using #ACTIVATION_CODE_4.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_4 (associated with PROFILE_OPERATIONAL4).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP with #METADATA_OP_PROF4 used in #GET_BPP_OK		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or

			Simple Confirmation is asked on the Profile download. (See NOTE)
2	S_EndUser → LPAd	End User Confirmation is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	PROC_ES9+_HANDLE_NOTIF		
4	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL4 is displayed in Enabled state
NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt.			

Test Sequence #02 Nominal: End User Confirmation after PPR2 consent requested

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR2 is allowed and End User Consent is required for #MCC_MNC2 with gid1 and gid2 absent.
LPAd	Add and Enable Profile operation is initiated by using #ACTIVATION_CODE_3_NO_CC.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL3).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP with #METADATA_OP_PROF3 used in #GET_BPP_OK		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download either at this point or at a previous point of the procedure Either Strong Confirmation is asked by showing relevant information concerning the PPR(s); or Simple Confirmation is asked on the Profile download. (See NOTE)
2	S_EndUser → LPAd	End User Confirmation is performed within the period as defined in	

		#IUT_EU_CONFIRMATION_TIMEOUT	
3	PROC_ES9+_HANDLE_NOTIF		
4	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL3 is displayed in Enabled state
NOTE: The request for this End User consent/Confirmation for the installation of Profile Policy Rules and Profile download MAY be combined into a single prompt.			

Test Sequence #03 Nominal: Profile with PPR1 already present

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed for #MCC_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required).
eUICC	The PROFILE_OPERATIONAL4 with PPR1 is installed and enabled on the eUICC.
LPAd	Add and Enable Profile operation is initiated by using #ACTIVATION_CODE_1.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
IC2	PROC_ES9+_INIT_AUTH		
IC3	PROC_ES9+_AUTH_CLIENT Extract <S_TRANSACTION_ID>		
IC4	PROC_ES9+_GET_BPP		
1	LPAd → S_EndUser	Request for Confirmation if not requested before.	The LPA provides means for the End User Confirmation/Rejection of the Profile Download. End User advised about a Profile with PPR1 already present and the End User consent is requested if not requested before.
2	S_EndUser → LPAd	End User Confirmation is performed within the period as defined in #IUT_EU_CONFIRMATION_TIMEOUT	
3	PROC_ES9+_HANDLE_NOTIF		
4	S_EndUser → LPAd	List Profile operation is initiated	PROFILE_OPERATIONAL1 is displayed in Disabled state

5.4.1.2.20 TC_LPAd_AddEnableProfile_Empty_MatchingID

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

Test Sequence #01 Nominal: Add and Enable a new Operational Profile by using empty MatchingID (QR code entry)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for PROFILE_OPERATIONAL1 linked to the EID of the eUICC.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate combined Add and Enable Profile operations	LPAd requests the Activation Code from the End User
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_5 by scanning the QR code	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.			

Test Sequence #02 Nominal: Add and Enable a new Operational Profile by using empty MatchingID (manual entry)

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for PROFILE_OPERATIONAL1 linked to the EID of the eUICC.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate combined Add and Enable Profile operations	LPAd requests the Activation Code from the End User
2	S_EndUser → LPAd	Provide #ACTIVATION_CODE_5 by manual entry	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
4	PROC_ES9+_INIT_AUTH		
5	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID>		
6	PROC_ES9+_GET_BPP (see NOTE 1)		
7	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
8	PROC_ES9+_HANDLE_NOTIF		
9	S_EndUser → LPAd	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Enabled state
NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.			

5.4.2 Local Profile Management – ListProfiles

5.4.2.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.2, 3.2.4
- Section 4.4

5.4.2.2 Test Cases

5.4.2.2.1 TC_LPAd_ListProfiles

General Initial Conditions	
Entity	**Description of the general initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
Device	The protection of access to the LUI is disabled.

Test Sequence #01 Nominal: List the Profiles and their current state

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is Enabled.

eUICC	The PROFILE_OPERATIONAL2 is Disabled.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Request the list of Profiles	Display PROFILE_OPERATIONAL1 with Enabled state and the PROFILE_OPERATIONAL2 with Disabled state in human readable format.

5.4.3 Local Profile Management - SetNickname

5.4.3.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.2, 3.2.6

5.4.3.2 Test Cases

5.4.3.2.1 TC_LPAd_SetNickname

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

Test Sequence #01 Nominal: Add a Nickname on a Disabled Operational Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is Disabled.
eUICC	The Nickname of the PROFILE_OPERATIONAL1 is not defined .

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Select PROFILE_OPERATIONAL1. Indicates the intention to change the Profile Nickname of PROFILE_OPERATIONAL1.	LPA offers to the End User a way to enter the Nickname.
2	S_EndUser→ LPAd	Set the Profile Nickname of the PROFILE_OPERATIONAL1 to #NICKNAME2	LPAd sets the Profile Nickname (No Error)
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
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

Test Sequence #02 Nominal: Add a Nickname on an Enabled Operational Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 is Enabled.
eUICC	The Nickname of the PROFILE_OPERATIONAL2 is not defined.

Step	Direction	Sequence / Description	Expected result
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.
2	S_EndUser→ LPAd	Set the Profile Nickname of the PROFILE_OPERATIONAL2 to #NICKNAME3	LPAd sets the Profile Nickname (No Error)
3	Exit the UI menu		
4	S_EndUser→ LPAd	Perform an LUI dependent action to display the NickName ofPROFILE_OPERATIONAL2	Profile Nickname of PROFILE_OPERATIONAL2 equals to #NICKNAME3
5	Power off then power on the Device		
6	S_EndUser→ LPAd	Perform an LUI dependent action to display the NickName of PROFILE_OPERATIONAL2.	Profile Nickname of PROFILE_OPERATIONAL2 equals to #NICKNAME3

5.4.3.2.2 TC_LPAd_EditNickname

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

Test Sequence #01 Nominal: Edit the Nickname on a Disabled Operational Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is Disabled.
eUICC	The Nickname of the PROFILE_OPERATIONAL1 is equal to #NICKNAME1.

Step	Direction	Sequence / Description	Expected result
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.
2	S_EndUser→ LPAd	Set the Profile Nickname of the PROFILE_OPERATIONAL1 to #NICKNAME2	LPAd sets the Profile Nickname (No Error)
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
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

Test Sequence #02 Nominal: Edit the Nickname on an Enabled Operational Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 is Enabled.
eUICC	The Nickname of the PROFILE_OPERATIONAL2 is equal to #NICKNAME3.

Step	Direction	Sequence / Description	Expected result
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.
2	S_EndUser→ LPAd	Set the Profile Nickname of the PROFILE_OPERATIONAL2 to #NICKNAME4	LPAd sets the Profile Nickname (No Error)
3	Exit the UI menu		
4	S_EndUser→ LPAd	Perform an LUI dependent action to display the NickName of PROFILE_OPERATIONAL2	Profile Nickname equals to #NICKNAME4
5	Power off then power on the Device		
6	S_EndUser→ LPAd	Perform an LUI dependent action to display the NickName ofPROFILE_OPERATIONAL2	Profile Nickname equals to #NICKNAME4

5.4.4 Local Profile Management - Delete Profile

5.4.4.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.2, 3.2.3
- Section 3.5

5.4.4.2 Test Cases

5.4.4.2.1 TC_LPAd_DeleteProfile_Disabled_without_PPR

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

Test Sequence #01 Nominal: Deleting Disabled Profile, No PPRs

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_TEST_DP_ADDRESS1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
eUICC	The PROFILE_OPERATIONAL2 is in Disabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Delete Profile procedure is initiated for PROFILE_OPERATIONAL1	Strong Confirmation is requested by the LPAd and confirmed by the End User.
2	LPAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF1 is sent by the LPAd	The delete Notification as defined below is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG
3	S_EndUser → LPAd	Request for List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL1 is not shown.
NOTE: The timeout in Step 2 SHALL start after the End User Intent verification.			

5.4.4.2.2 TC_LPAd_DeleteProfile_Enabled_without_PPR

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

Test Sequence #01 Nominal: Deleting Enabled Profile, No PPRs

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
1	S_EndUser → LPAd	Initiate Delete Profile procedure for PROFILE_OPERATIONAL5	Strong Confirmation is requested by the LPAd and confirmed by the End User.
2	LPAd → S_SM-DP+	Send Disable Notification containing #ICCID_OP_PROF5	The disable Notification as defined below is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS5) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG See NOTE
3	LPAd → S_SM-DP+	Send Delete Notification containing #ICCID_OP_PROF5	The delete Notification as defined below is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL5) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG See NOTE
4	S_EndUser → LPAd	Request for List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL5 is not shown.
5	S_EndUser → Device	Power off then power on the Device	During Device boot up no PIN entry is requested from the End User.
NOTE: The timeout SHALL start after the End User Intent verification.			

5.4.4.2.3 TC_LPAd_DeleteProfile_Error_with_PPR1

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

Test Sequence #01 Error: Deleting Enabled Profile, PPR1 set

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed for #MCC_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required).
eUICC	The PROFILE_OPERATIONAL4 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Delete Profile procedure is initiated for PROFILE_OPERATIONAL4	Strong Confirmation is requested by the LPAd and confirmed by the End User. See NOTE 1 and NOTE 2
2	S_EndUser→ LPAd	Request for List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL4 is shown in Enabled state.
NOTE 1: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure. NOTE 2: The LPAd MAY display an error indicating that the deletion of the Profile is failed.			

5.4.4.2.4 TC_LPAd_DeleteProfile_Error_Disabled_with_PPR2

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

Test Sequence #01 Error: Deleting Disabled Profile, PPR2 set

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR2 is allowed for #MCC_MNC2 with gid1 and gid2 absent (with End User Consent either required or not required).
eUICC	The PROFILE_OPERATIONAL7 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL7 is in Disabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Delete Profile procedure is initiated for PROFILE_OPERATIONAL7	Strong Confirmation is requested by the LPAd and confirmed by the End User. See NOTE 1 and NOTE 2
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.
NOTE 1: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure. NOTE 2: The LPAd MAY display an error indicating that the deletion of the Profile is failed.			

5.4.4.2.5 TC_LPAd_DeleteProfile_Error_Enabled_with_PPR2

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

Test Sequence #01 Error: Deleting Enabled Profile, PPR2 set

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR2 is allowed for #MCC_MNC2 with gid1 and gid2 absent (with End User Consent either required or not required).
eUICC	The PROFILE_OPERATIONAL8 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL8 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate Delete Profile procedure for PROFILE_OPERATIONAL8	Strong Confirmation is requested by the LPAd and confirmed by the End User. See NOTE 2 and NOTE 3
2	LPAd → S_SM-DP+	Send Disable Notification containing #ICCID_OP_PROF8	The disable Notification as defined below is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT MTD_HANDLE_NOTIF (#PENDING_NOTIF_DIS8) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG See NOTE 1

3	S_EndUser→ LPAd	Request for List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL8 is shown in Disabled state.
4	S_EndUser→ Device	Power off then power on the Device	During Device boot up no PIN entry is requested from the End User.

NOTE 1: The timeout SHALL start after the End User Intent verification.

NOTE 2: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure.

NOTE 3: The LPAd MAY display an error indicating that the deletion of the Profile is failed.

5.4.4.2.6 TC_LPAd_DeleteProfile_Security_Errors

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

Test Sequence #01 Error: Stop Delete Profile Operation if No Confirmation Provided

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.

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

5.4.4.2.7 TC_LPAd_DeleteProfiles_MEP_Enabled_without_PPR

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

Test Sequence #01 Nominal: Deleting one of two Enabled Profiles, No PPRs

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Enabled state.
eUICC	The PROFILE_OPERATIONAL2 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate Delete Profile procedure for PROFILE_OPERATIONAL1	Strong Confirmation is requested by the LPAd and confirmed by the End User.
2	LPAd → S_SM-DP+	Send Disable Notification containing #ICCID_OP_PROF1	The disable Notification as defined below is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS1) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG See NOTE
3	LPAd → S_SM-DP+	Send Delete Notification containing #ICCID_OP_PROF1	The delete Notification as defined below is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG See NOTE
4	S_EndUser → LPAd	Request for List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL1 is not shown and PROFILE_OPERATIONAL2 is shown in Enabled state.
NOTE: The timeout (steps 2 and 3) SHALL start after the End User Intent verification.			

5.4.5 Local Profile Management - Enable Profile

5.4.5.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.2, 3.2.1, 3.2.4
- Section 3.5

5.4.5.2 Test Cases

5.4.5.2.1 TC_LPAd_EnableProfile

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
Device	The End User gets presented a list of installed (operational) Profiles with their current state.

Test Sequence #01 Nominal: Enable a formerly disabled Profile

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
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL5	PROFILE_OPERATIONAL5 is enabled
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
3	S_EndUser → Device	Enter #PO1_PIN1 to authenticate the user	Successful End User authentication for the selected application
4	S_EndUser → LPAd	Request List Profiles	PROFILE_OPERATIONAL5 is shown in Enabled state.
NOTE: The timeout SHALL start after the initiation of the Enable Profile operation.			

5.4.5.2.2 TC_LPAd_EnableProfile_ImplicitDisable

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

Test Sequence #01 Nominal: Enable a Profile with implicit disabling of the formerly enabled Profile

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
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL6	
2	LPAd → S_SM-DP+(1)	Disable Notification containing #ICCID_OP_PROF5 is sent by the LPAd	The Disable Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS5) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS1) within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_ECDSA S_SM-DP+ SHALL return #R_HTTP_204_OK
3	LPAd → S_SM-DP+(2)	Send the Enable Notification containing #ICCID_OP_PROF6	The Enable Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_EN6) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS2) within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_ECDSA S_SM-DP+ SHALL return #R_HTTP_204_OK
4	S_EndUser → Device	Enter #PO2_PIN1 to authenticate the user	Successful End User authentication for the selected application
5	S_EndUser → LPAd	Request List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL6 is shown in Enabled state.
NOTE 1: The Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel. NOTE 2: The timeout (steps 2 and 3) SHALL start after the initiation of the Enable Profile operation.			

5.4.5.2.3 TC_LPAd_EnableProfile_Error_ProfileAlreadyEnabled

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

Test Sequence #01 Error: Enable an already enabled Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL1	
2	LPAd → S_EndUser	Result of the Profile enabling	Enable Profile procedure terminates indicating an error

5.4.5.2.4 TC_LPAd_EnableProfile_Error_PPR1Set

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

Test Sequence #01 Error: Enabled Profile when a formerly enabled Profile has set PPR1

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed for #MCC_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required).
eUICC	The PROFILE_OPERATIONAL4 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
IC1	S_EndUser → LPAd	Install PROFILE_OPERATIONAL1 and ensure that it is disabled (see NOTE 1)	PROFILE_OPERATIONAL1 is installed and disabled
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL1	See NOTE 2 and NOTE 3

2	S_EndUser → LPAd	Request List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format. PROFILE_OPERATIONAL4 is shown in Enabled state.

NOTE 1: If the device supports only O_D_ADD_ENABLE_COMBINED, any attempt to automatically enable the profile is expected to fail.

NOTE 2: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure.

NOTE 3: The LPAd MAY display an error indicating that the enabling of the Profile is failed.

5.4.5.2.5 Void

5.4.5.2.6 TC_LPAd_EnableProfile_MEP

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 two Disabled Profiles

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 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
1	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL1	PROFILE_OPERATIONAL1 is enabled
2	S_EndUser → LPAd	Initiate the Enable Profile operation for PROFILE_OPERATIONAL2	PROFILE_OPERATIONAL2 is enabled
3	LPAd → S_SM-DP+	Send the Enable Notification containing #ICCID_OP_PROF1	The Enable Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_EN1) is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT

			Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG S_SM-DP+ SHALL return #R_HTTP_204_OK
4	LPAd → S_SM-DP+	Send the Enable Notification containing #ICCID_OP_PROF2	The Enable Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_EN2) 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_SIG S_SM-DP+ SHALL return #R_HTTP_204_OK
5	S_EndUser → LPAd	Request List Profiles	PROFILE_OPERATIONAL1 and PROFILE_OPERATIONAL2 are shown in Enabled state.

NOTE 1: The Notifications (steps 3 and 4) MAY be sent sequentially in either order or in parallel.

NOTE 2: The timeout (steps 3 and 4) SHALL start after the initiation of the Enable Profile operation.

5.4.6 Local Profile Management- Disable Profile

5.4.6.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.2, 3.2.2, 3.2.4
- Section 3.5

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
1	S_EndUser → LPAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL1	PROFILE_OPERATIONAL1 is disabled
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
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.
NOTE: The timeout SHALL start after the initiation of the Disable Profile operation.			

5.4.6.2.2 TC_LPAd_DisableProfile_Error_ProfileAlreadyDisabled

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

Test Sequence #01 Error: Disable an already disabled Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL1	
2	LPAd → S_EndUser	Result of the Profile disabling	The Disable Profile procedure terminates indicating a failure

5.4.6.2.3 TC_LPAd_DisableProfile_Error_PPR1Set

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

Test Sequence #01 Error: Disable an Enabled Profile with PPR1 set

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR1 is allowed for #MCC_MNC4 with gid1 and gid2 absent (with End User Consent either required or not required).
eUICC	The PROFILE_OPERATIONAL4 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL4	See NOTE 1 and NOTE 2
2	S_EndUser → LPAd	Request List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format PROFILE_OPERATIONAL4 is shown in Enabled state
NOTE 1: The LPAd MAY check the policy rules of the Profiles and give a warning to the End User. The procedure can be continued after the warning and the End User shall continue the procedure. NOTE 2: The LPAd MAY display an error indicating that the disabling of the Profile is failed.			

5.4.6.2.4 VOID

5.4.6.2.5 TC_LPAd_DisableProfile_MEP

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 two Enabled Profiles

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Enabled state.
eUICC	The PROFILE_OPERATIONAL2 is in Enabled state.
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL1	PROFILE_OPERATIONAL1 is disabled
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_SIG S_SM-DP+ SHALL return #R_HTTP_204_OK
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 and PROFILE_OPERATIONAL2 is shown in Enabled state.
4	S_EndUser → LPAd	Initiate the Disable Profile operation for PROFILE_OPERATIONAL2	PROFILE_OPERATIONAL2 is disabled
5	LPAd → S_SM-DP+	Send the Disable Notification containing #ICCID_OP_PROF2	The Disable Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS2) 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_SIG S_SM-DP+ SHALL return #R_HTTP_204_OK
6	S_EndUser → LPAd	Request List Profiles	Installed Operational Profile(s) with their current states are displayed in a human readable format. PROFILE_OPERATIONAL1 and PROFILE_OPERATIONAL2 are shown in Disabled state.
NOTE 1: The Notifications (steps 2 and 5) MAY be sent sequentially in either order or in parallel. NOTE 2: The timeout (steps 2 and 5) SHALL start after the initiation of the Enable Profile operation.			

5.4.7 Local eUICC Management - Retrieve EID Process

5.4.7.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.3.1

5.4.7.2 Test Cases

5.4.7.2.1 TC_LPAd_RetrieveEID

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

Test Sequence #01 Nominal: Retrieve EID

The purpose of this test is to check if the Device is capable to display the stored EID in as QR code or in text string format.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Request to display EID. (See NOTE)	EID is displayed.
2	LPAd → S_EndUser	Presentation of the EID	The LPA presents the #EID1 to the End User as a text string and/or as a QR code. If the EID is represented as text string, the text SHALL be identical to #EID1 If the #EID1 is shown as a QR code it SHALL be either #EID1_QR_CODE1 or #EID1_QR_CODE2 with or without blank spaces.
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]:

- Section 3.2.4
- Section 3.3.2, 3.3.4
- Section 3.5

5.4.8.2 Test Cases

5.4.8.2.1 TC_LPAd_eUICCMemoryReset

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	No proactive session is ongoing. NOTE: These test cases MAY fail due to the fact that a proactive is ongoing but it is impossible to determine that this is the case. In this instance it is recommended to repeat the test.

Device	The protection of access to the LUI is disabled.

Test Sequence #01 Nominal: eUICC Memory Reset, Operational Profile installed, no Operational Profile enabled

The purpose of this test is to check the basic functions of the eUICC Memory Reset. An installed but not enabled Operational Profile SHALL be deleted.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the eUICC Memory Reset for operational profiles	Strong Confirmation is requested by the LPAd and confirmed by the End User.
2	LPAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF1 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1) is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG The S_SM-DP+ SHALL return #R_HTTP_204_OK
3	S_EndUser → LPAd	Request List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available
NOTE: The timeout (step 2) SHALL start after the End User Intent verification.			

Test Sequence #02 Nominal: eUICC Memory Reset, Operational Profile with PPR2 installed, no Operational Profile enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed but not enabled Operational Profile with PPR2 ('Deletion of this Profile is not allowed').

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR2 is allowed for #MCC_MNC2 with gid1 and gid2 absent (with End User Consent either required, or not required).
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC with #METADATA_OP_PROF2_MEMRES1.
eUICC	The PROFILE_OPERATIONAL2 is in Disabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the eUICC Memory Reset for operational profiles	Strong Confirmation is requested by the LPAd and confirmed by the End User.
2	LPAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF2 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL2) is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG The S_SM-DP+ SHALL return #R_HTTP_204_OK
3	S_EndUser → LPAd	**Request List Profiles**	Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available
NOTE: The timeout (step 2) SHALL start after the End User Intent verification.			

Test Sequence #03 Nominal: eUICC Memory Reset, Operational Profile with PPR2 installed and enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed and enabled Operational Profile with PPR2 (''Deletion of this Profile is not allowed'').

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC's RAT is configured as follows: PPR2 is allowed for #MCC_MNC2 with gid1 and gid2 absent (with End User Consent either required, or not required).
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC with #METADATA_OP_PROF2_MEMRES1.
eUICC	The PROFILE_OPERATIONAL2 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the eUICC Memory Reset for operational profiles	Strong Confirmation is requested by the LPAd and confirmed by the End User.

2	LPAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF2 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL2) is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG The S_SM-DP+ SHALL return #R_HTTP_204_OK See NOTE 2
3	S_EndUser → LPAd	**Request List Profiles**	Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available RQ32_053 RQ33_012
NOTE 1: The timeout (step 2) SHALL start after the End User Intent verification. NOTE 2: A Disable Notification for PROFILE_OPERATIONAL2 MAY be sent before the Delete Notification. This notification SHALL NOT be checked.			

Test Sequence #04 Nominal: eUICC Memory Reset, Operational Profile with PPR1 installed and enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed and enabled Operational Profile with PPR1 (‘‘Disabling of this Profile is not allowed’’).

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Test eUICC’s RAT is configured as follows: PPR1 is allowed for #MCC_MNC4 with gid1 and gid2 absent (with End User Consent either required, or not required).
eUICC	The PROFILE_OPERATIONAL4 is installed on the eUICC with #METADATA_OP_PROF4_MEMRES1.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the eUICC Memory Reset for operational profiles	Strong Confirmation is requested by the LPAd and confirmed by the End User.

2	LPAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF4 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL4) is received by the S_SM-DP+ within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG The S_SM-DP+ SHALL return #R_HTTP_204_OK See NOTE 2
3	S_EndUser → LPAd	**Request List Profiles**	Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available
NOTE 1: The timeout (step 2) SHALL start after the End User Intent verification. NOTE 2: A Disable Notification for PROFILE_OPERATIONAL4 MAY be sent before the Delete Notification. This notification SHALL NOT be checked.			

Test Sequence #05 Nominal: eUICC Memory Reset, multiple Operational Profiles are installed, an Operational Profile is enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes all Operational Profiles installed and send the required Notifications to the appropriate SM-DP+.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Enabled state.
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 is in Disabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the eUICC Memory Reset for operational profiles	Strong Confirmation is requested by the LPAd and confirmed by the End User.
2	LPAd → S_SM-DP+(1)	Delete Notifications containing #ICCID_OP_PROF1 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS1) within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT The S_SM-DP+ SHALL return #R_HTTP_204_OK See NOTE 3

3	LPAd → S_SM-DP+(2)	Delete Notification containing #ICCID_OP_PROF2 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL2) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS2) within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG The S_SM-DP+ SHALL return #R_HTTP_204_OK
4	S_EndUser → LPAd	Request List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available
NOTE 1: The Delete Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel. NOTE 2: The timeout (steps 2 and 3) SHALL start after the End User Intent verification. NOTE 3: A Disable Notification for PROFILE_OPERATIONAL1 MAY be sent before the Delete Notification. This notification SHALL NOT be checked.			

5.4.8.2.2 TC_LPAd_eUICCMemoryResetWithPINVerification

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	No proactive session is ongoing. NOTE: these test cases may fail due to the fact that a proactive session is ongoing but it is impossible to determine that this is the case. In this instance it is recommended to repeat the test.
Device	The protection of access to the LUI is disabled.

Test Sequence #01 Nominal: eUICC Memory Reset, installed and enabled Operational Profile with PIN verification

The purpose of this test is to check if an initiated eUICC Memory Reset deletes an installed and enabled Operational Profile with PIN verification enabled.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL5 is installed on the eUICC with #METADATA_OP_PROF5.
eUICC	The PROFILE_OPERATIONAL5 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the eUICC Memory Reset for operational profiles	Strong Confirmation is requested by the LPAd and confirmed by the End User.

2	LPAd → S_SM-DP+	Delete Notification containing #ICCID_OP_PROF5 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL5) is received by the S_SM-DP+ within the timeout (#IUT_LPAd_NOTIFICATION_TIMEOUT, + #IUT_LPAd_READY_AFTER_REBOOT_TIMEOUT) Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG The S_SM-DP+ SHALL return #R_HTTP_204_OK See NOTE 3
3	Device	Power off then power on the Device If the Device does not automatically power off and power on, the S_EndUser SHALL power off and power on the Device.	During Device boot up no PIN entry is requested from the End User.
4	S_EndUser → LPAd	Request List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available
NOTE 1: The Delete Notification (step 2) can be sent at any step after having successfully initiated the eUICC Memory Reset. NOTE 2: The timeout (step 2) SHALL start after the End User Intent verification. NOTE 3: A Disable Notification for PROFILE_OPERATIONAL5 MAY be sent before the Delete Notification. This notification SHALL NOT be checked.			

Test Sequence #02 Nominal: VOID

5.4.8.2.3 TC_LPAd_eUICCMemoryReset_MEP

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, multiple Operational Profiles are installed, two Operational Profiles are enabled

The purpose of this test is to check if an initiated eUICC Memory Reset deletes all Operational Profiles installed and send the required Notifications to the appropriate SM-DP+.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Enabled state.
eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the eUICC Memory Reset for operational profiles	Strong Confirmation is requested by the LPAd and confirmed by the End User.
2	LPAd → S_SM-DP+(1)	Delete Notifications containing #ICCID_OP_PROF1 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL1) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS1) within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT The S_SM-DP+ SHALL return #R_HTTP_204_OK See NOTE 3
3	LPAd → S_SM-DP+(2)	Delete Notification containing #ICCID_OP_PROF2 is sent by the LPAd	The Delete Notification MTD_HANDLE_NOTIF(#PENDING_NOTIF_DEL2) is received by the S_SM-DP+ (configured with #TEST_DP_ADDRESS2) within the timeout #IUT_LPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_SIG The S_SM-DP+ SHALL return #R_HTTP_204_OK
4	S_EndUser → LPAd	Request List Profiles	Installed Operational Profiles with their current states are displayed in a human readable format No Operational Profile is available
NOTE 1: The Delete Notifications (steps 2 and 3) MAY be sent sequentially in either order or in parallel. NOTE 2: The timeout (steps 2 and 3) SHALL start after the End User Intent verification. NOTE 3: Disable Notifications for PROFILE_OPERATIONAL1 and PROFILE_OPERATIONAL2 MAY be sent before the Delete Notification. These notifications SHALL NOT be checked.			

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]:

- Section 3.3.4

5.4.10.2 Test Cases

5.4.10.2.1 TC_LPAd_Set/Edit Default SM-DP+ Address

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

Test Sequence #01 Nominal: Set Default SM-DP+ Address where no Default Address has been set before

The purpose of this test is to set a default SM-DP+ address on a eUICC where no SM-DP+ default address is stored.

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

Step	Direction	Sequence / Description	Expected result
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
2	S_EndUser → LPAd	If required, initiate the function to enter #TEST_DP_ADDRESS1 as the new Default SM-DP+ address or enter directly #TEST_DP_ADDRESS1 as the new Default SM-DP+.	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before. (see NOTE 1)
3	S_EndUser → LPAd	Initiate the function to retrieve the configured address	The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser The current Default SM-DP+ Address #TEST_DP_ADDRESS1 is shown
NOTE 1: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

Test Sequence #02 Nominal: Edit the Default SM-DP+ Address and store it on the eUICC

The purpose of this test is to edit an existing default SM-DP+ address on a eUICC and to ensure that the changes are stored.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Default SM-DP+ field is set to #TEST_DEFAULT_DP_ADDRESS_1.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the function to retrieve the configured address	The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser The current Default SM-DP+ Address is #TEST_DEFAULT_DP_ADDRESS_1
2	S_EndUser → LPAd	If required, initiate the function to enter #TEST_DP_ADDRESS1 as the new Default SM-DP+ address or enter directly #TEST_DP_ADDRESS1 as the new Default SM-DP+.	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before. (see NOTE 1)
3	S_EndUser → LPAd	Initiate the function to retrieve the configured address	The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser The current Default SM-DP+ Address #TEST_DP_ADDRESS1 is shown
NOTE 1: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

Test Sequence #03 Nominal: Edit the Default SM-DP+ Address and store a Default Address with an empty value

The purpose of this test is to edit an existing Default SM-DP+ address on a eUICC and to ensure that the changes are stored even if the new Default Address value is empty.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The Default SM-DP+ field is set to #TEST_DEFAULT_DP_ADDRESS_1.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd	Initiate the function to retrieve the configured address	The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser The current Default SM-DP+ Address is #TEST_DEFAULT_DP_ADDRESS_1
2	S_EndUser → LPAd	If required, initiate the function to enter “” (empty value) as the new Default SM-DP+ address or enter directly “” as the new Default SM-DP+.	Successful End User Intent verified as defined in SGP.21 [3] for Simple Confirmation, if not verified before. (see NOTE 1)
3	S_EndUser → LPAd	Initiate the function to retrieve the configured address	The LPAd retrieves the Default SM-DP+ Address and presents it to the EndUser The current Default SM-DP+ Address is empty respectively no Default SM-DP+ Address is shown
NOTE 1: The LPAd MAY skip this request for Confirmation. If so, it SHALL NOT be regarded as a failure.			

5.4.11 Device Power On – Profile Discovery

5.4.11.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- Section 3.4.4

5.4.11.2 Test Cases

5.4.11.2.1 TC_LPAd_DevicePowerOnProfileDiscovery_SM-DP+_address

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
Device	The setting of the configuration parameter for Device Power-on Profile discovery is 'Enabled'.
Device	The Device is powered off.

Test Sequence #01 Nominal: Power-on Profile discovery by using the default SM-DP+ Address

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
S_SM-DP+	There is a pending Profile download order for PROFILE_OPERATIONAL1 linked to the EID of the eUICC.

Step	Direction	Sequence / Description	Expected result
IC1	Power on the Device		
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	PROC_ES9+_INIT_AUTH		
3	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID> or missing MatchingID data object		
4	PROC_ES9+_GET_BPP (see NOTE 1)		
5	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified, if not verified before.
6	PROC_ES9+_HANDLE_NOTIF		
7	LPAd → S_EndUser	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

5.4.11.2.2 TC_LPAd_DevicePowerOnProfileDiscovery_SM-DS

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled.
Device	The setting of the configuration parameter for Device Power-on Profile discovery is 'Enabled'.
Device	The Device is powered off.

Test Sequence #01 Nominal: Power-on Profile discovery by using the SM-DS

Initial Conditions	
Entity	**Description of the initial condition**
S_SM-DS	S_SM-DP+ (#TEST_DP_ADDRESS1) performed Profile download Event Registration to the S_SM-DS (#TEST_ROOT_DS_ADDRESS) with #EVENT_ID_1.
S_SM-DP+	There is a pending Profile download order for #EVENT_ID_1 (PROFILE_OPERATIONAL1).
S_SM-DP+	The PROFILE_OPERATIONAL1 on the S_SM-DP+ is in “Released” state.
eUICC	There is no default SM-DP+ address configured.

Step	Direction	Sequence / Description	Expected result
IC1	Power-on the Device		
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES11		
2	PROC_ES11_INIT_AUTH		
3	PROC_ES11_AUTH_CLIENT with #MATCHING_ID_EMPTY as <MATCHING_ID> or missing MatchingID data object		
4	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
5	PROC_ES9+_INIT_AUTH		
6	PROC_ES9+_AUTH_CLIENT with #EVENT_ID_1 as <MATCHING_ID>		
7	PROC_ES9+_GET_BPP (see NOTE 1)		
8	LPAd → S_EndUser	Request for Confirmation, if not requested before.	End User Intent successfully verified , if not verified before.
9	PROC_ES9+_HANDLE_NOTIF		
10	LPAd → S_EndUser	Initiate List Profile operation	PROFILE_OPERATIONAL1 is displayed in Disabled state

NOTE 1: The LPAd MAY display any relevant part of the Profile Metadata and MAY offer the S_EndUser to postpone or reject the Profile installation. The S_EndUser SHALL not abort the transaction.

5.4.12 RPM Command Execution - Enable Profile

5.4.12.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- 2.6.6.2, 2.10.1
- 3.0.1
- 3.2.7, 3.7.2
- 3.7.3
- 5.5.3, 5.6.3, 5.7.14a
- 5.7.16

5.4.12.2 Test Cases

5.4.12.2.1 TC_LPAd_RPM_Command_Execution_EnableProfile

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
Device	RPM operation is enabled in the LPA by the End User
eUICC	There is no default SM-DP+ address configured The eUICC supports RPM
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: RPM Enable a formerly disabled Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_EN is installed and disabled.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrievesPolling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_V3		
4	PROC_ES9+_AUTH_CLIENT_RPM with #MATCHING_ID_1 as <MATCHING_ID>		
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation

6	S_EndUser		PROFILE_OPERATIONAL1 is enabled
7	PROC_ES9+_HANDLE_NOTIF_RPM_OK		
8	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is displayed in Enabled state
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: RPM Command - RPM Enable a Profile with implicit disabling of the formerly enabled Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_EN is installed and disabled.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1 (without PPR1 present) is loaded and Enabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrievesPolling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(enable, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation

6	S_EndUser		PROFILE_OPERATIONAL1 is enabled
7	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(enableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)))
8	S_SM-DP+ → LPAd		No error
9	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Enabled state PROFILE_OPERATIONAL2 is in Disabled state
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 #03 Error: RPM Command Result Data Error - Enable Profile, Disallowed Managing SM-DP+

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_EN_DP_OID2 is installed and disabled.
S_SM-DP+	There is a pending RPM package order with OID 1 for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) The SM-DP+ that sent the RPM Command is not included in the Managing SM-DP+ List in the Profile Metadata (rpmConfiguration) of PROFILE_OPERATIONAL1.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.

2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_V3		
4	PROC_ES9+_AUTH_CLIENT_RPM with #MATCHING_ID_1 as <MATCHING_ID> and #RPM_PKG_EN is included in <S_SMDP_SIGNED3>		
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser		The LPAd MAY inform the End User of the status indicating the error disallowedManagingDp.
7	PROC_ES9+_HANDLE_NOTIF_RPR_ERROR(MTD_RESP_RPR_FOR_SINGLE_CMND(enableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, disallowedManagingDp))		
8	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Disabled state

Test Sequence #04 Error: RPM Command Result Data Error - Enable Profile, Disallowed by Policy

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 has been loaded on the eUICC with #METADATA_OP_PROF1_RPM_CONF_EN and is disabled.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1_PPR1 has been loaded before the PROFILE_OPERATIONAL1 and is Enabled.
S_SM-DP+	There is a pending RPM package order with OID 1 for #MATCHING_ID_1 (PROFILE_OPERATIONAL1)

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrievesPolling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_V3		
4	PROC_ES9+_AUTH_CLIENT_RPM with #MATCHING_ID_1 as <MATCHING_ID> and #RPM_PKG_EN is included in <S_SMDP_SIGNED3>		
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser		The LPAd MAY inform the End User of the status indicating the error disallowedByPolicy.
7	PROC_ES9+_HANDLE_NOTIF_RPR_ERROR(MTD_RESP_RPR_FOR_SINGLE_CMND(enableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, disallowedByPolicy))		
8	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Disabled state

5.4.13 RPM Command Execution - Disable Profile

5.4.13.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- 2.6.6.2, 2.10.1
- 3.0.1
- 3.2.7, 3.7.2
- 3.7.3
- 5.5.3, 5.6.3, 5.7.14a
- 5.7.17

5.4.13.2 Test Cases

5.4.13.2.1 TC_LPAd_RPM_Command_Execution_DisableProfile

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
Device	RPM operation is enabled in the LPA by the End User
eUICC	There is no default SM-DP+ address configured The eUICC supports RPM
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: RPM Command - Disable an Enabled Profile without PPR

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_ALL is installed and Enabled.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Disabling the Profile

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(disable, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser		PROFILE_OPERATIONAL1 is disabled

7	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(disableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)))
8	S_SM-DP+ → LPAd		No error
9	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Disabled state
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 Error: RPM Command Result Data Error - Disable Profile, ICCID Not Found

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_ALL is installed and Enabled. (PPR1 is not set in the Metadata)
eUICC	The Operational Profile identified by the ICCID #ICCID_OP_PROFX is not loaded.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Disabling the Profile.
S_SM-DP+	ICCID in the RPM package is set to #ICCID_OP_PROFX.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		

3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(disable, <S_TRANSACTION_ID>, #ICCID_OP_PROFX, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser		The LPAd MAY inform the End User of the status indicating the error iccidOrAidNotFound.
7	PROC_ES9+_HANDLE_NOTIF_RPR_ERROR(MTD_RESP_RPR_FOR_SINGLE_CMND(disableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROFX, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, iccidOrAidNotFound))		
8	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Enabled state

Test Sequence #03 Error: RPM Command Result Data Error - Disable Profile, Profile is not in Enabled state

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 has been loaded on the eUICC with #METADATA_OP_PROF1_RPM_CONF_ALL and it is in Disabled state.

S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Disabling the Profile.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(disable, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser		The LPAd MAY inform the End User of the status indicating the error profileNotInEnabledState.
7	PROC_ES9+_HANDLE_NOTIF_RPR_ERROR(MTD_RESP_RPR_FOR_SINGLE_CMND(disableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, profileNotInEnabledState))		
8	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Enabled state

5.4.14 RPM Command Execution - Delete Profile

5.4.14.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- 2.6.6.2, 2.10.1
- 3.0.1
- 3.2.7, 3.7.2
- 3.7.3
- 5.5.3, 5.6.3, 5.7.14a
- 5.7.18

5.4.14.2 Test Cases

5.4.14.2.1 TC_LPAd_RPM_Command_Execution_DeleteProfile

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
Device	RPM operation is enabled in the LPA by the End User
eUICC	There is no default SM-DP+ address configured The eUICC supports RPM
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: RPM Command - Delete a Disabled Profile without PPR

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_RPM_CONF_ALL is installed and Disabled.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1 (without PPR1 present) is loaded and Enabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Deleting the Profile

Step	Direction	Sequence / Description	Expected result
1	S_EndUser → LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		

4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(delete, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Strong Confirmation
6	S_EndUser		PROFILE_OPERATIONAL1 is deleted
7	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(deleteResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)))
8	S_SM-DP+ → LPAd		No error
9	S_EndUser → LPAd		PROFILE_OPERATIONAL2 is in Enabled state
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: RPM Command - Delete a Disabled Profile without PPR2 and with PPR1

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_ALL_PPR1 is installed and Disabled.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1 is loaded and Enabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Deleting the Profile

Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(delete, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error See NOTE 1 below.
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Strong Confirmation LPAd may request End User Consent and it may combine with the Confirmation
6	S_EndUser		PROFILE_OPERATIONAL1 is deleted
7	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(

			MTD_RESP_RPR_FOR_SINGLE_CMND(deleteResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)))
8	S_SM-DP+ → LPAd		No error
9	S_EndUser → LPAd		PROFILE_OPERATIONAL2 is in Enabled state
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 #03 Error: RPM Command Result Data Error - Delete Profile, Profile is not in Disabled state

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 has been loaded on the eUICC with #METADATA_OP_PROF1_RPM_CONF_ALL and it is in Enabled state. (PPR2 is not set in the Metadata)
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Disabling the Profile.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(delete, <S_TRANSACTION_ID>,	No error

		#ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM) **)**	
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser		The LPAd MAY inform the End User of the status indicating the error profileNotInDisabledState.
7	PROC_ES9+_HANDLE_NOTIF_RPR_ERROR(MTD_RESP_RPR_FOR_SINGLE_CMND(deleteResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, profileNotInDisabledState))		
8	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Enabled state

Test Sequence #04 Error: RPM Command Result Data Error - Delete Profile, disallowed by Policy

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 has been loaded with #METADATA_OP_PROF1_RPM_CONF_DE_PPR2 and is Disabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Disabling the Profile.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		

4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(delete, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_EndUser		LPAd initiates Confirmation Request for Simple Confirmation
6	S_EndUser		The LPAd MAY inform the End User of the status indicating the error disallowedByPolicy.
7	PROC_ES9+_HANDLE_NOTIF_RPR_ERROR(MTD_RESP_RPR_FOR_SINGLE_CMND(disableResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, disallowedByPolicy))		
8	S_EndUser → LPAd		PROFILE_OPERATIONAL1 is in Disable state

5.4.15 RPM Command Execution – List Profile Info

5.4.15.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- 2.6.6.2, 2.10.1
- 3.0.1
- 3.2.7, 3.7.2
- 3.7.3

- 5.5.3, 5.6.3, 5.7.14a
- 5.7.15

5.4.15.2 Test Cases

5.4.15.2.1 TC_LPAd_RPM_Command_Execution_ListProfileInfo

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
Device	RPM operation is enabled in the LPA by the End User
eUICC	There is no default SM-DP+ address configured
S_SM-DP+	The eUICC supports RPM

Test Sequence #01 Nominal: RPM Command - ListProfileInfo_by ICCID

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_ALL is installed and Enabled.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_ALL_OWNER2 is loaded and Disabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for ListProfileInfo the Profile

Step	Direction	Sequence / Description	Expected result
1	S_End User→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO (<S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, NO_PARAM))	No error No user Confirmation

5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(listProfileInfoResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, #PROFILE_INFO1, NO_PARAM, NO_PARAM)))
6	S_SM-DP+ → LPAd		No error

Test Sequence #02 Nominal: RPM Command - ListProfileInfo_with ICCID and tagList PPR present

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_ALL_PPR1 is installed and Enabled.
eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_ALL_OWNER2 is loaded and Disabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for ListProfileInfo the Profile

Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_End User→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		

3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO (<S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, '99'H))	No error No user Confirmation
5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(listProfileInfoResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, #PROFILES_INFO_TAGLIST_PPR1, NO_PARAM, NO_PARAM)))
6	S_SM-DP+ → LPAd		No error

Test Sequence #03 Nominal: RPM Command - ListProfileInfo_with Profile Owner ID and tagList with multiple Tags present

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_ALL_PPR1 is installed and Enabled.

eUICC	The PROFILE_OPERATIONAL2 with #METADATA_OP_PROF2_RPM_CONF_ALL_OWNER2 is loaded and Disabled on the eUICC.
eUICC	The PROFILE_OPERATIONAL3 with #METADATA_OP_PROF3_RPM_CONF_ALL is loaded and Disabled on the eUICC.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for ListProfileInfo the Profile

Step	Direction	Sequence / Description	Expected result
1	S_End User→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO (<S_TRANSACTION_ID>, NO_PARAM, <S_SM_DP+_SIGNATURE3>, #S_PROFILE_OWNER_OID, 'BB9ABC5A'H))	No error No user Confirmation
5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(listProfileInfoResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, { #PROFILES_INFO_MULT_TAGS_PROFILE1,

			#PROFILES_INFO_MULT_TAGS_PROFILE3}, NO_PARAM, NO_PARAM)))
6	S_SM-DP+ → LPAd		No error

Test Sequence #04 Error: RPM Command - ListProfileInfo_ICCID specified, SM-DP+ OID not in Managing SM-DP+ list

The purpose of this Test Sequence is to ensure RPM Command ListProfileInfo is not executed if the SM-DP+ that sent the RPM Command is not included in the Managing SM-DP+ List in the Profile Metadata (rpmConfiguration) and LPA sends the Notification with error 'disallowedManaging' to the SM-DP+.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with #METADATA_OP_PROF1_RPM_CONF_ALL_DP_OID2 is installed and Enabled.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for RPM commands ListProfileInfo (TagList with rpmConfiguration).

Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO (<S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, NO_PARAM, '5A'H	No error No user Confirmation

		))	
5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(listProfileInfoResult, <S_TRANSACTION_ID>, NO_PARAM, #ICCID_OP_PROF1, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, disallowedManaging)))
6	S_SM-DP+ → LPAd		No error

Test Sequence #05 Error: RPM Command_ListProfileInfo_- Profile Owner ID specified, allowed CI public key identifier does not match

The purpose of this test sequence is to ensure RPM Command ListProfileInfo is not executed if Subject Key Identifier of the CI corresponding to CERT.DPauth.ECDSA attached to the ongoing session does not match with the Allowed CI Public Key Identifier in the Profile Metadata (rpmConfiguration) and LPA sends the Notification with error ‘ciPKMismatch’ to the SM-DP+.

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with # METADATA_OP_PROF1_RPM_CONF_ALL_CI_PKI_RAND is installed and Enabled.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for ListProfileInfo (with TagList with multiple tags)

Step	Direction	Sequence / Description	Expected result
1	S_End User→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO (<S_TRANSACTION_ID>, NO_PARAM, <S_SM_DP+_SIGNATURE3>, #S_PROFILE_OWNER_OID, 'BB9ABC5A'H))	No error No user Confirmation
5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(listProfileInfoResult, <S_TRANSACTION_ID>, NO_PARAM, #ICCID_OP_PROF1, 1, -- error response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, ciPKMismatch)))
6	S_SM-DP+ → LPAd		No error

5.4.16 RPM Command Execution – Update Metadata

5.4.16.1 Conformance Requirements

References

GSMA RSP Technical Specification [2]:

- 2.6.6.2, 2.10.1
- 3.0.1
- 3.2.7, 3.7.2
- 3.7.3
- 5.5.3, 5.6.3, 5.7.14a

5.4.16.2 Test Cases

5.4.16.2.1 TC_LPAd_RPM_Command_Execution_UpdateMetadata

General Initial Conditions	
Entity	**Description of the general initial condition**
Device	The protection of access to the LUI is disabled
Device	RPM operation is enabled in the LPA by the End User
eUICC	There is no default SM-DP+ address configured The eUICC supports RPM
S_SM-DP+	Variant A certificates are included in certificates chain for TLS procedures.

Test Sequence #01 Nominal: RPM Command - UpdateMetadata – Remove PPR1

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	PROFILE_OPERATIONAL1 with # METADATA_OP_PROF1_RPM_CONF_UPDATE_MD_PPR is installed and Enabled.
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Update Metadata.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		

4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(updateMetadata, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, profilePolicyRules {ppr2}, NO_PARAM, NO_PARAM))	No error See NOTE 1 below.
5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(updateMetadataResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)))
6	S_SM-DP+ → LPAd		No error
NOTE 1: The LPAd MAY ask for the S_End_User consent by showing relevant information concerning the Profile and PPR(s). This information MAY include the consequences of the Profile Policy Rule			

Test Sequence #02 Nominal: RPM Command - UpdateMetadata – RPM Configuration

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 has been installed on the eUICC with #METADATA_OP_PROF1_RPM_CONF_ALL_PPR1
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Update Metadata.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(updateMetadata, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, rpmConfiguration #RPM_CONFIG_OP_PROF1, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(updateMetadataResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)))
6	S_SM-DP+ → LPAd		No error

Test Sequence #03 Nominal: RPM Command UpdateMetadata – Multiple Tags

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 has been installed on the eUICC with #METADATA_OP_PROF1_RPM_CONF_ALL_PPR1
S_SM-DP+	There is a pending RPM package order for #MATCHING_ID_1 (PROFILE_OPERATIONAL1) for Update Metadata.

Step	Direction	Sequence / Description	Expected result
1	S_EndUser→ LPAd		LPAd retrieves Polling Address from the ProfileInfo, and the Polling Address indicates the SM-DP+ address #TEST_DP_ADDRESS1.
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A on ES9+		
3	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3		
4	S_SM-DP+ → LPAd	MTD_HTTP_RESP(MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND(updateMetadata, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, <S_SM_DP+_SIGNATURE3>, #MULTIPLE_TAGS_OP_PROF1, NO_PARAM, NO_PARAM))	No error
5	LPAd → S_SM-DP+		MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(MTD_RESP_RPR_FOR_SINGLE_CMND(updateMetadataResult, <S_TRANSACTION_ID>, #ICCID_OP_PROF1, 0, -- OK response #NOTIF_METADATA_PROF1_DP1_RPR, #S_SM_DP+_OID, NO_PARAM, NO_PARAM, NO_PARAM)

			))
6	S_SM-DP+ → LPAd		No error

6 VOID

7 VOID

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$$1 ACTIVATION_CODE_3.png as defined in Annex H
ACTIVATION_CODE_3_NO_CC	1$#TEST_DP_ADDRESS1$#MATCHING_ID_3 ACTIVATION_CODE_3_NO_CC.png as defined in Annex H
ACTIVATION_CODE_4	1$#TEST_DP_ADDRESS1$#MATCHING_ID_4 ACTIVATION_CODE_4.png as defined in Annex H
ACTIVATION_CODE_5	1$#TEST_DP_ADDRESS1$#MATCHING_ID_EMPTY ACTIVATION_CODE_5.png as defined in Annex H
ACTIVATION_CODE_INVALID_FORMAT	1#TEST_DP_ADDRESS1$#MATCHING_ID_1 ACTIVATION_CODE_INVALID_FORMAT.png as defined in Annex H
ACTIVATION_CODE_WITH_CI_PK_IND_RAND	1#TEST_DP_ADDRESS1$#MATCHING_ID_1$#S_SM_DP+_OID$$#PK_CI_ECDSA_RANDOM 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

Name	Content
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.
CTX_PARAMS1_MATCH_ID_DEV_INFO_V3 (CtxParams1)	ctxParamsForCommonAuthentication : { matchingId <MATCHING_ID>, deviceInfo #DEVICE_INFO, operationType { profileDownload }, OPTIONAL matchingIdSource activationCode : NULL, vendorSpecificExtension #VENDOR_SPECIFIC_EXT -- OPTIONAL }
CTX_PARAMS1_RPM_ICCID1	ctxParamsForCommonAuthentication : { #DEVICE_INFO, operationType { rpm }, iccid #ICCID_OP_PROF1, matchingIdSource none NULL, vendorSpecificExtension #VENDOR_SPECIFIC_EXT -- OPTIONAL }
DEVICE_INFO	deviceInfo { tac ..., deviceCapabilities { ... }, imei #IUT_IMEI -- Optional preferredLanguages ..., -- OPTIONAL deviceTestMode ..., -- OPTIONAL lpaRspCapability <LPA_RSP_CAPABILITY> }-- Check only that the field is present and has a valid TLV asn.1 structure unless it is defined differently in the individual test cases.
EF_UST1	0x0A 2E 14 8C E7 32 04 00 00 00 00 00 00 -- NOTE: Service n°17 (GID1) and n°18 (GID2) not available
EF_UST2	0x0A 2E 17 8C E7 32 04 00 00 00 00 00 00 -- NOTE: Service n°17 (GID1) and n°18 (GID2) available
EID1	0x89 04 90 32 12 34 51 23 45 12 34 56 78 90 12 35
EID1_QR_CODE1	QR code which decodes as: EID:89049032123451234512345678901235
EID1_QR_CODE2	QR code which decodes as: EID:89 04 90 32 12 34 51 23 45 12 34 56 78 90 12 35

Name	Content
EID2	0x89 04 90 32 11 23 41 23 40 12 34 56 78 90 13 75
ENT_CONF_REF_ENT_RULE	{ enterpriseOid #S_ENTERPRISE_OID, enterpriseName #ENTERPRISE_NAME1, enterpriseRules { referenceEnterpriseRule, onlyThisProfileCanBeEnabled, onlyEnterpriseProfilesCanBeInstalled } }
EUICC_CI_PK_ID_LIST_FOR_SIGNING_1	#CI_PKI_ID1, #CI_PKI_ID2
EUICC_CI_PK_ID_LIST_FOR_SIGNING_2	#CI_PKI_ID3, #CI_PKI_ID4
EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1	#CI_PKI_ID1, #CI_PKI_ID2
EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_2	#CI_PKI_ID3, #CI_PKI_ID4
EUICC_INFO1_8_8_2_3_1	euiccInfo1_8_8_2_3_1 EUICCInfo1 ::= { svn #RSP_SVN, euiccCiPKIdListForVerification { #EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1 }, euiccCiPKIdListForSigning { #EUICC_CI_PK_ID_LIST_FOR_SIGNING_2 } }
EUICC_INFO1_8_8_3_3_1_HIGHER	euiccInfo1_8_8_3_3_1 EUICCInfo1 ::= { svn #RSP_SVN_HIGHER, euiccCiPKIdListForVerification { #EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1 }, euiccCiPKIdListForSigning { #EUICC_CI_PK_ID_LIST_FOR_SIGNING_1 } }
EUICC_INFO1_8_8_3_3_1_LOWER	euiccInfo1_8_8_3_3_1 EUICCInfo1 ::= { svn #RSP_SVN_LOWER, euiccCiPKIdListForVerification { #EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1 }, euiccCiPKIdListForSigning { #EUICC_CI_PK_ID_LIST_FOR_SIGNING_1 } }

Name	Content
EUICC_INFO1_8_8_4_3_7	euiccInfo1_8_8_4_3_7 EUICCInfo1 ::= { svn #RSP_SVN, euiccCiPKIdListForVerification { #EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_2 }, euiccCiPKIdListForSigning { #EUICC_CI_PK_ID_LIST_FOR_SIGNING_1 } }
EUICC_SIGNED1	{ transactionId <S_TRANSACTION_ID>, serverAddress #TEST_DP_ADDRESS1, serverChallenge <S_SMDP_CHALLENGE>, euiccInfo2 #R_EUICC_INFO2, -- check only that the field is present and has a valid TLV asn.1 structure ctxParams1 #CTX_PARAMS1 }
EVENT_ID_1	07399-BGH7E-T8779
EVENT_ID_2	07399-BGH7E-T8778
EXT_SHA256_RSA	TLS extension data for "supported_signature_algorithms" set as: o HashAlgorithm sha256 (04) and o SignatureAlgorithm rsa (01).
FUNCTION_CALL_ID_1	0000-0000-0000-0001
FUNCTION_CALL_ID_2	0000-0000-0000-0002
GID1	0x47 53 4D 41
GID2	0x52 53 50 FF
HOST_ID	0x47 53 4D 41 20 53 4D 2D 58 58 -- NOTE: 'GSMA SM-XX' in ASCII
ICCID_OP_PROF1	0x98 92 09 01 21 43 65 87 09 F5
ICCID_OP_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

Name	Content
ICCID_OP_PROFX	0x98 92 09 01 43 65 87 09 FF FF
ICCID_UNKNOWN	0x98 92 01 0A 21 43 65 87 09 F8
ICON_JPG	ICON_JPG.jpg as defined in Annex H
ICON_OP_PROF1	profile_O1.png as defined in Annex H
ICON_OP_PROF1_2_SEG	profile_O1_2_SEG.png as defined in Annex H
ICON_OP_PROF2	profile_O2.png as defined in Annex H
ICON_OP_PROF3	profile_O3.png as defined in Annex H
ICON_OP_PROF4	profile_O4.png as defined in Annex H
ICON_OP_PROF5	profile_O5.png as defined in Annex H
ICON_OP_PROF6	profile_O6.png as defined in Annex H
ICON_OP_PROF7	profile_O7.png as defined in Annex H
ICON_OP_PROF8	profile_O8.png as defined in Annex H
IMSI_OP_PROF1	0x08 29 99 18 11 32 54 76 98
IMSI_OP_PROF2	0x08 29 99 28 11 32 54 76 97
IMSI_OP_PROF3	0x08 29 99 28 11 32 54 76 96
IMSI_OP_PROF4	0x08 29 99 48 43 65 87 09 21
IMSI_OP_PROF5	0x08 29 99 18 11 32 54 76 98
IMSI_OP_PROF6	0x08 29 99 28 11 32 54 76 97
IMSI_OP_PROF7	0x08 29 99 28 43 65 87 09 21
IMSI_OP_PROF8	0x08 29 99 28 43 65 87 09 21
IMSI_OP_PROF9	0x08 29 99 98 43 65 87 09 21
INSTALLED_PROFILES	0x00
INVALID_KEY_TYPE	0x80
INVALID_REMOTE_OP_ID	8
ISD_R_AID	0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 01 00
KEY_LENGTH	0x10
KEY_TYPE	0x88
MATCHING_ID_1	04386-AGYFT-A74Y8-3F815
MATCHING_ID_2	04386-AGYFT-A74Y8-3F816

Name	Content
MATCHING_ID_3	04386-AGYFT-A74Y8-3F817
MATCHING_ID_4	04386-AGYFT-A74Y8-3F818
MCC_MNC_WILDCARD	0x92 F9 EE
MCC_MNC1	0x92 F9 18
MCC_MNC2	0x92 F9 28
MCC_MNC4	0x92 F9 48
MCC_MNC9	0x92 F9 98
MIN_TLS_CIPHER_SUITES	The minimum TLS cipher suites proposed by the Client: o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
MNO_SCP80_AUTH_KEY	0x11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 10
MNO_SCP80_DATA_ENC_KEY	0x99 AA BB CC DD EE FF 10 11 22 33 44 55 66 77 88
MNO_SCP80_ENC_KEY	0x66 77 88 99 AA BB CC DD 11 22 33 44 55 EE FF 10
NAME_OP_PROF_LONG	Operational Profile Name with long name of sixty four characters NOTE: the exact text above SHOULD be used, as it is exactly 64 characters long.
NAME_OP_PROF1	Operational Profile Name 1
NAME_OP_PROF1_NON_ASCII	Operational Profile Name UTF-8 encoding: 0x4F 70 65 72 61 74 69 6F 6E 61 6C 20 50 72 6F 66 69 6C 65 20 4E 61 6D 65 20 E4 BD A0 E5 A5 BD
NAME_OP_PROF2	Operational Profile Name 2
NAME_OP_PROF3	Operational Profile Name 3
NAME_OP_PROF4	Operational Profile Name 4
NAME_OP_PROF5	Operational Profile Name 5
NAME_OP_PROF6	Operational Profile Name 6
NAME_OP_PROF7	Operational Profile Name 7
NAME_OP_PROF8	Operational Profile Name 8
NAME_OP_PROF9	Operational Profile Name 9
NICKNAME1	Nickname 1
NICKNAME2	Nickname 2
NICKNAME3	Nickname 3

Name	Content
NICKNAME4	Nickname 4
OWNER_OP_PROF1	{ mccMnc #MCC_MNC1 }
OWNER_OP_PROF2	{ mccMnc #MCC_MNC2 }
PATH_AUTH_CLIENT	/gsma/rsp2/es9plus/authenticateClient
PATH_CANCEL_ORDER	/gsma/rsp2/es2plus/cancelOrder
PATH_CANCEL_SESSION	/gsma/rsp2/es9plus/cancelSession
PATH_CONFIRM_ORDER	/gsma/rsp2/es2plus/confirmOrder
PATH_DELETE_EVENT	/gsma/rsp2/es12/deleteEvent
PATH_DOWNLOAD_ORDER	/gsma/rsp2/es2plus/downloadOrder
PATH_GET_BPP	/gsma/rsp2/es9plus/getBoundProfilePackage
PATH_HANDLE_NOTIF	/gsma/rsp2/es9plus/handleNotification
PATH_INITIATE_AUTH	/gsma/rsp2/es9plus/initiateAuthentication
PATH_REGISTER_EVENT	/gsma/rsp2/es12/registerEvent
PO1_PIN1	0x32 34 36 38 FF FF FF FF
PO2_PIN1	0x33 35 37 39 FF FF FF FF
PPK_ENC_INV_SIZE	0x01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 0D 0E 0F 10 0D 0E 0F 10
PPK_INIT_MAC_INV_SIZE	0x05 0A 04 0B 03 0C 02 0D 01 0E 00 0F 09 01 08 02 09 01 08 02 09 01 08 02
PPK_MAC_INV_SIZE	0x01 0E 00 0F 09 01 08 02 05 0A 04 0B 03 0C 02 0D 03 0C 02 0D 03 0C 02 0D
PROFILE_STATUS_AVAILABLE	Available
PROP_TLS_CIPHER_SUITES	The TLS cipher suites proposed by the Client: o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 o TLS_RSA_WITH_AES_128_CBC_SHA o TLS_RSA_WITH_AES_256_CBC_SHA256 o TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
REMOTE_OP_ID_INSTALL	1
RPM_PKG_DELETE	rpmPackage { { continueOnFailure NULL, rpmCommandDetails delete : { iccid #ICCID_OP_PROF1 } } }

Name	Content
RPM_PKG_DISABLE	rpmPackage { { continueOnFailure NULL, rpmCommandDetails disable : { iccid #ICCID_OP_PROF1 } }
RPM_PKG_EN	rpmPackage { { continueOnFailure NULL, rpmCommandDetails enable : { iccid #ICCID_OP_PROF1 } }
RPM_PKG_UM_PPR	rpmPackage { { continueOnFailure NULL, rpmCommandDetails updateMetadata : { iccid #ICCID_OP_PROF1, updateMetadataRequest { profilePolicyRules {ppr2}} } }
RPM_PKG_UM_REF_ENT_RULE	rpmPackage { { continueOnFailure NULL, rpmCommandDetails updateMetadata : { iccid #ICCID_OP_PROF1, updateMetadataRequest { enterpriseConfiguration #ENT_CONF_REF_ENT_RULE } } }
RSP_SVN_LOWEST	eUICC lowest svn encoded as the value part of an ASN.1 VersionType (e.g. 0x02 00 00)
RSP_SVN_HIGHEST	eUICC highest svn encoded as the value part of an ASN.1 VersionType (e.g. 0x03 01 00)
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_CAP_EXT	deviceCapExt DeviceCapExt ::= { unknownServiceSupport2 } using the following definition of DeviceCapExt:

Name	Content
	DeviceCapExt ::= INTEGER { unknownServiceSupport1 (0), unknownServiceSupport2 (1) }
S_DEVICE_INFO	deviceInfo { tac #S_TAC, deviceCapabilities { gsmSupportedRelease '050000'H, utranSupportedRelease '080000'H, cdma2000onexSupportedRelease '010000'H, cdma2000hrpdSupportedRelease '010000'H, cdma2000ehrpdSupportedRelease '020000'H, eutranSupportedRelease '020000'H, contactlessSupportedRelease '090000'H, rspCrlSupportedVersion #RSP_SVN_H } }
S_DEVICE_INFO_EXT	deviceInfo DeviceInfo { tac #S_TAC, deviceCapabilities { gsmSupportedRelease '050000'H, utranSupportedRelease '080000'H, cdma2000onexSupportedRelease '010000'H, cdma2000hrpdSupportedRelease '010000'H, cdma2000ehrpdSupportedRelease '020000'H, eutranEpcSupportedRelease '020000'H, contactlessSupportedRelease '090000'H, rspCrlSupportedVersion, #RSP_SVN_H nrEpcSupportedRelease '0F0000'H, nr5gcSupportedRelease '0F0000'H, eutran5gcSupportedRelease '0F0000'H, unknownServiceSupport #S_DEVICE_CAP_EXT } } Note: the definition of DeviceInfo used above is equivalent to the definition in SGP.22 v2.3 (specific version of [2]) with the additional of a further field called “unknownServiceSupport” of type DeviceCapExt (see #S_DEVICE_CAP_EXT) after the “eutran5gcSupportedRelease” field.
S_DEVICE_INFO_IMEI	deviceInfo { tac #S_TAC, deviceCapabilities { gsmSupportedRelease '050000'H, utranSupportedRelease '080000'H, cdma2000onexSupportedRelease '01000'H, eutranSupportedRelease '020000'H }, imei #S_IMEI }

Name	Content
S_EUICC_CHALLENGE	0x01 02 03 04 05 06 07 08 01 02 03 04 05 06 07 08
S_EUICC_CHALLENGE_2	0x21 22 23 24 25 26 27 28 21 22 23 24 25 26 27 28
S_EUICC_INFO1	euiccInfo1 EUICCInfo1 ::= { svn #RSP_SVN, euiccCiPKIdListForVerification { #EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1 }, euiccCiPKIdListForSigning { #EUICC_CI_PK_ID_LIST_FOR_SIGNING_1 } }
S_EUICC_INFO2	euiccInfo2 EUICCInfo2 ::= { profileVersion #PROFILE_VERSION, svn #RSP_SVN_H, euiccFirmwareVer #EUICC_FIRMWARE_VER, extCardResource #S_EXT_CARD_RESOURCE, uiccCapability #UICC_CAPABILITY, rspCapability #RSP_CAPABILITY, euiccCiPKIdListForVerification {#EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1}, euiccCiPKIdListForSigning {#EUICC_CI_PK_ID_LIST_FOR_SIGNING_1}, ppVersion #PP_VERSION, sasAcreditationNumber #SAS_ACREDITATION_NUMBER }
S_EUICC_INFO2_UICC_EXT	euiccInfo2 EUICCInfo2 ::= { profileVersion #PROFILE_VERSION, svn #RSP_SVN_H, euiccFirmwareVer #EUICC_FIRMWARE_VER, extCardResource #S_EXT_CARD_RESOURCE, uiccCapability #UICC_CAPABILITY_EXT, rspCapability #RSP_CAPABILITY, euiccCiPKIdListForVerification {#EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1}, euiccCiPKIdListForSigning {#EUICC_CI_PK_ID_LIST_FOR_SIGNING_1}, ppVersion #PP_VERSION, sasAcreditationNumber #SAS_ACREDITATION_NUMBER }
S_EUICC_INFO2_DEV_EXT	euiccInfo2 EUICCInfo2 ::= { profileVersion #PROFILE_VERSION, svn #RSP_SVN_H, euiccFirmwareVer

Name	Content
	#EUICC_FIRMWARE_VER, extCardResource #S_EXT_CARD_RESOURCE, uiccCapability #UICC_CAPABILITY, rspCapability #RSP_CAPABILITY_EXT, euiccCiPKIdListForVerification {#EUICC_CI_PK_ID_LIST_FOR_VERIFICATION_1}, euiccCiPKIdListForSigning {#EUICC_CI_PK_ID_LIST_FOR_SIGNING_1}, ppVersion #PP_VERSION, sasAcreditationNumber #SAS_ACREDITATION_NUMBER }
S_EXT_SHA256_ECDSA	TLS extension data for "supported_signature_algorithms" set as: o HashAlgorithm sha256 (04) and o SignatureAlgorithm ecdsa (03).
S_IMEI	0x00 00 00 00 11 11 11 11
S_MNO_F_REQ_ID	"S_MNO"
S_PROFILE_OWNER_OID	2.888.9
S_PROFILE_OWNER_OID2	2.888.99
S_SAH_SHA256_ECDSA	Signature And Hash Algorithm extension sent in the CertificateRequest message set as: o HashAlgorithm sha256 (04) and o SignatureAlgorithm ecsda(3).
S_SESSION_ID_EMPTY	Empty TLS session ID to identify a new session, with the Length set as 'zero'.
S_SM_DP_SVN	0x03 00 00
S_SM_DP+_F_REQ_ID	"S_SM_DP_PLUS"
S_SM_DP+_OID	2.999.10
S_SM_DP+_OID2	2.999.12
S_SM_DP+_OID4	2.999.14
S_SM_DP+_OID8	2.999.18
S_SM_DS_F_REQ_ID	"S_SM_DS"
S_SM_DS_OID	2.999.15
S_TAC	0x00 00 00 00
S_TLS_CIPHER_SUITE	TLS cipher suite selected as follows: o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
SERVER_ADDRESS	FQDN of the SERVER Under Test which can be one of the following depending on the entity under test:

Name	Content
	• #IUT_SM_DP_ADDRESS • #IUT_SM_DS_ADDRESS_ES11
SIMA_RESULT_OK	`simaresp EUICCResponse ::= {` `  peStatus {` `  {status ok}` `  }` `}`
SP_NAME_LONG	SP Name as thirty two characters NOTE: the exact text above SHOULD be used, as it is exactly 32 characters long.
SP_NAME_NON_ASCII	SP Name UTF-8 encoding: `0x53 50 20 4E 61 6D 65 20 E3 83 AB`
SP_NAME1	SP Name 1
SP_NAME2	SP Name 2
SP_NAME3	SP Name 3
SP_NAME4	SP Name 4
SP_NAME8	SP Name 8
SP_NAME9	SP Name 9
SSD_AID	`0xA0 00 00 05 59 10 10 01 02 73 64 56 61 6C 75 65`
TEST_ALT_DS_ADDRESS	testaltsmds.example.com
TEST_DEFAULT_DP_ADDRESS_1	testdefaultsmdpplus1.example.com
TEST_DP_ADDRESS1	testsmdpplus1.example.com
TEST_DP_ADDRESS2	testsmdpplus2.example.com
TEST_DP_ADDRESS3	testsmdpplus3.example.com
TEST_DP_ADDRESS4	testsmdpplus4.example.com
TEST_DP_ADDRESS8	testsmdpplus8.example.com
TEST_DS_ADDRESS1	testsmds1.example.com
TEST_HRI_ADDRESS3	testhriserver3.example.com
TEST_PCMP_ADDRESS1	testpcmp1.example.com
TEST_ROOT_DS_ADDRESS	testrootsmds.example.com
TLS_VERSION_1_1	1.1
TLS_VERSION_1_2	1.2 The minimum TLS Version supported by the Server.
UNKNOWN_BPP_SEGMENT	`0xC9 05 01 02 03 04 05`
UNKNOWN_SERVER_ADDRESS	unknownserver.example.com

Name	Content
UNSUP_TLS_CIPHER_SUITES	The TLS cipher suites proposed by the Client: o TLS_RSA_WITH_AES_128_CBC_SHA o TLS_RSA_WITH_AES_256_CBC_SHA256
UPP_OP_PROF1	The Unprotected Profile Package related to the PROFILE_OPERATIONAL1 (see Annex E).
UPP_OP_PROF2	The Unprotected Profile Package related to the PROFILE_OPERATIONAL2 (see Annex E).
UPP_OP_PROF3	The Unprotected Profile Package related to the PROFILE_OPERATIONAL3 (see Annex E).
UPP_OP_PROF4	The Unprotected Profile Package related to the PROFILE_OPERATIONAL4 (see Annex E).
UPP_OP_PROF9	The Unprotected Profile Package related to the PROFILE_OPERATIONAL9 (see Annex E).
USIM_AID	0xA0 00 00 00 87 10 02 FF 33 FF 01 89 00 00 01 00
VENDOR_SPECIFIC_EXT	Additional extensions to include the OID of the vendor and the vendorSpecificData defined by vendor, coded as ASN.1 SEQUENCE_OF_SEQUENCE (VendorSpecificExtension).

A.2 Test Certificates and Test Keys

All ECC certificates and keys described below are based on descriptions and security requirements (such as algorithms to be used) from SGP.22[2].

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 bundled with SGP.26 [25].

Name	Description
CERT_CI_SIG	Certificate of the CI for its Public SIG Key
CERT_CI_SubCA_SIG	The intermediate SubCA Certificate through which #CERT_EUM_SIG is chained to the **#**CERT_CI_SIG in Variants A and C. This certificate contains the same Extension for subjectAltName value as in #CERT_CI_SIG of that same GSMA CI.
CERT_EUICC_SIG	Certificate of the eUICC for its Public key CERT.EUICC.SIG in the X.509 format signed by the EUM with SK.EUM.SIG
CERT_EUM_SIG	Certificate of the EUM for its Public ECDSA key CERT.EUM.ECDSA in the X.509 format signed by the requested CI with SK.CI.ECDSA.
CERT_S_SERVER_TLS	CERT.SERVER.TLS certificate of the S_SERVER, based on NIST or Brainpool for this version of the specification,

	where the Certificate MAY be one of the following depending on the role of the simulator: • #CERT_S_SM_DP_TLS on ES9+ • #CERT_S_SM_DS_TLS on ES11 or ES12
CERT_S_SM_DPauth_PK_CI2_SIG	Certificate of the S_SM-DP+ for its Public key used for SM-DP+ authentication. This certificate contains the OID #S_SM_DP+_OID. The Subject Key Identifier of the GSMA CI RootCA Certificate corresponding to this SM-DPauth certificate is #PK_CI2_SIG.
CERT_S_SM_DS_SubCA_SIG	The intermediate SubCA Certificate through which #CERT_S_SM_DSauth_SIG and the #CERT_S_SM_DSpb_SIG and the #CERT_S_SM_DS_TLS are chained to the #CERT_CI_SIG in Variant A or to the #CERT_CI_SubCA_SIG in Variant C. This certificate contains the Extension for subjectAltName as OID #S_SM_DS_OID.
CERT_S_SM_DP_SubCA_SIG	The intermediate SubCA Certificate through which #CERT_S_SM_DPauth_SIG and the #CERT_S_SM_DPpb_SIG and the #CERT_S_SM_DP_TLS are chained to the #CERT_CI_SIG in Variant A or to the #CERT_CI_SubCA_SIG in Variant C. This certificate contains the Extension for subjectAltName as OID #S_SM_DP+_OID.
CERT_S_SM_DP_TLS	CERT.DP.TLS certificate of the S_SM-DP+, based on the same CI as defined in #IUT_LPAd_CI.
CERT_S_SM_DP2_TLS	CERT.DP.TLS certificate of the S_SM-DP+, based on the same CI as defined in #IUT_LPAd_CI. Contains different SM-DP+ hostname (FQDN) as #CERT_S_SM_DP2_TLS.
CERT_S_SM_DP4_TLS	CERT.DP.TLS certificate of the S_SM-DP+, based on the same CI as defined in #IUT_LPAd_CI. Contains the SM-DP+ hostname (FQDN) #TEST_DP_ADDRESS4 and OID value #S_SM_DP+_OID4.
CERT_S_SM_DP8_TLS	CERT.DP.TLS certificate of the S_SM-DP+, based on the same CI as defined in #IUT_LPAd_CI. Contains the SM-DP+ hostname (FQDN) #TEST_DP_ADDRESS8 and OID value #S_SM_DP+_OID8.
CERT_S_SM_DP_TLS_EXPIRED	Expired CERT.DP.TLS certificate of the S_SM-DP+ with a valid signature, correctly formatted as X.509 certificate.
CERT_S_SM_DP_TLS_INV_CURVE	CERT.DP.TLS certificate of the S_SM-DP+, based on the different CI as defined in #IUT_LPAd_CI, not based on the curves defined in SGP.22[2].
CERT_S_SM_DP_TLS_INV_SIG	Invalid CERT.DP.TLS certificate of the S_SM-DP+ with an invalid signature with the same tag and length as a valid signature, correctly formatted as X.509 certificate.

CERT_S_SM_DPauth_SIG	Certificate of the S_SM-DP+ for its Public SIG key used for SM-DP+ authentication. This certificate contains the OID #S_SM_DP+_OID.
CERT_S_SM_DP2auth_SIG	Certificate of the S_SM-DP+ for its Public 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 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_SIG 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 curves defined in SGP.22[2].
CERT_S_SM_DSauth_INV_CURVE	Certificate of the S_SM-DS for its Public key used for Authentication. This certificate contains the OID #S_SM_DS_OID and a public key based on a curve different from the curves defined in SGP.22[2].
CERT_S_SM_DPpb_SIG	Certificate of the S_SM-DP+ for its Public 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 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_SIG 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 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 curves defined in SGP.22[2].
CERT_S_SM_DP2pb_SIG	Certificate of the S_SM-DP+ for its Public 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_CURVE	CERT.DP.TLS certificate of the S_SM-DP+, based on the different CI as defined in #IUT_LPAd_CI, not based on the curves defined in SGP.22[2].

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_SIG	Certificate of the S_SM-DS for its Public 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 key used for SM-DS authentication. This certificate contains an invalid signature, (i.e. not generated with the #SK_CI_SIG but with the same tag and length as a valid signature)
CERT_S_SM_DP_SubCAList_SIG **(CertificateChain)**	#CERT_S_SM_DP_SubCA_SIG CERT_CI_SubCA_SIG
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`
CRL_LIST	The list of CRLs needed to verify the revocation status of each Certificate that contains a `cRLDistributionPoints` extension in the returned Certificate chain.
PK_S_SM_DP_TLS	Public key of CERT.DP.TLS of the S_SM-DP+.
PK_S_SM_DPauth_SIG	Public Key of the S_SM-DP+, contained within #CERT_S_SM_DPauth_SIG
PK_S_SM_DPpb_SIG	Public Key of the S_SM-DP+, contained within #CERT_S_SM_DPpb_SIG
PK_S_SM_DS_TLS	Public key of CERT_S_DS_TLS of the S_SM-DS.
PK_SM_DPauth_SIG	Public Key of the SM-DP+, contained within #CERT_SM_DPauth_SIG
PK_SM_DPpb_SIG	Public Key of the SM-DP+, contained within #CERT_SM_DPpb_SIG
PK_SM_DSauth_SIG	Public Key of the SM-DS, contained within #CERT_SM_DSauth_SIG
SK_CI_SIG	Private Key of the CI
SK_EUICC_SIG	Private key of the eUICC for creating signatures
SK_S_SM_DPauth_SIG	Private Key of the of S_SM-DP+ for creating signatures for SM-DP+ authentication
SK_S_SM_DSauth_SIG	Private Key of the of S_SM-DS for creating signatures for SM-DS authentication
SK_S_SM_DPpb_SIG	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_SEG_A0	Bound Profile Package TLV segment containing the tag and length fields of the firstSequenceOf87 TLV plus the first 0x87 TLV containing the ConfigureISDP command
BPP_SEG_A1	Bound Profile Package following TLV segment array, as defined in SGP.22 [2] – section 2.5.5: • array first element containing the Tag and length fields of the sequenceOf88 TLV • array following elements containing each of the '88' TLVs containing the StoreMetadata command
BPP_SEG_A2	Bound Profile Package TLV segment containing the Tag and length fields of the secondSequenceOf87 TLV plus the first '87' TLV, containing the ReplaceSessionKeys command
BPP_SEG_A3	Bound Profile Package following TLV segment array, as defined in SGP.22 [2] – section 2.5.5: • array first element containing the tag and length fields of the sequenceOf86 TLV • array following elements containing each of the '86' TLVs containing the Protected Profile Package (PPP)
BPP_SEG_INIT	Bound Profile Package TLV segment containing the tag and length fields of the BoundProfilePackage TLV plus the initialiseSecureChannelRequest command
C_APDUS_SCRIPT	List of Command APDUs formatted as an expanded structure with definite length coding as defined in ETSI TS 102 226 [14].
CC	SCP80 cryptographic checksum as defined in ETSI TS 102 225 [13] (8 bytes long).
CHANNEL_NUMBER	The logical channel number newly opened in the eUICC. If no logical channel is opened, the value is set to 0x00 (i.e. Basic Channel).
CI_PKI_RANDOM	Random Subject Key Identifier of the PK CI different from all the PK CI Identifiers defined in SGP.26 [25]. This random value has the same length as the ones defined in SGP.26 [25].
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.SIG signed over euiccCancelSessionSigned coded as ASN.1 OCTET STRING.

Variable	Description
EUICC_CANCEL_SESSION_SIGNATURE_INVALID	eUICC signature randomly generated and coded as an ASN.1 OCTET STRING not equal to <EUICC_CANCEL_SESSION_SIGNATURE> but with the same length as a valid signature
EUICC_CHALLENGE	Random eUICC challenge, coded as asn.1 OCTET STRING, 16 bytes.
EUICC_CI_PK_ID_LIST_FOR_SIGNING	List of CI Public Key Identifiers supported on the eUICC for signature creation, coded as ASN.1 sequence of SubjectKeyIdentifier. The CI Public Key Identifier is from the list of possible CI Public Key Identifier. This possible CI Public Key Identifiers as supported by the eUICC will be defined later on.
EUICC_CI_PK_ID_LIST_FOR_SIGNING_V3	List of CI Public Key Identifiers supported on the eUICC for signature creation that can be verified by a certificate chain Variant A, B or C, coded as ASN.1 sequence of SubjectKeyIdentifier. The CI Public Key Identifiers are from the list of possible identifiers as defined in SGP.26 [25].
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_NEXT_CERT	Next Certificate in the eUICC Certificates Chain. eg : If Variant A Certificates are used it is the intermediate certificate CERT_EUM_SubCA_ECDSA
EUICC_OTHER_CERTS	Other Certificates in the eUICC Certificates Chain. eg : If Variant A Certificates are used it is the CERT_EUM_ECDSA.
EUICC_RSP_CAPABILITY	RspCapability of the eUICC, coded as ASN.1 BIT STRING (4 bits) to be used for indication of additionalProfile, loadcrlSupport, rpmSupport, testProfileSupport, deviceInfoExtensibilitySupport, osUpdateSupport, lpaApiSupport, enterpriseProfilesSupport, lpaProxySupport, crlStaplingV3Support, certChainV3VerificationSupport, serviceProviderMessageSupport, signedEventRecordsV3Support
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_SIGN_RPR	The eUICC signature of the Load RPM Package Result (RPR). The input data used to generate the <EUICC_SIGN_RPR> is the loadRpmPackageResultDataSigned and smdpSignature3. euiccSignRPR shall be created using the SK.EUICC.ECDSA and verified using the PK.EUICC.ECDSA.

Variable	Description
EUICC_SIGNATURE1	The eUICC signature 1 (euiccSignature1) computed using #SK_EUICC_ECDSA across the euiccSigned1 present in the AuthenticateServerResponse structure, coded as ASN.1 OCTET STRING.
EUICC_SIGNATURE1_INVALID	eUICC signature randomly generated and coded as an ASN.1 OCTET STRING not equal to <EUICC_SIGNATURE1>
EUICC_SIGNATURE2	The eUICC signature 2 (euiccSignature2) computed using the #SK_EUICC_ECDSA across the following data objects: • euiccSigned2 • smdpSignature2 present in the PrepareDownloadRequest structure
EUICC_SIGNATURE2_INVALID	eUICC signature randomly generated and coded as an ASN.1 OCTET STRING not equal to <EUICC_SIGNATURE2>
EVENT_ID	An EventID value in String format, generated by the SM-DP+ during Event Record registration.
EVENT_ID_D	An EventID value in String format, generated by the Alternative SM-DS during cascaded Event Record deletion on ES15
EVENT_ID_R	The EventID value in String format generated by the Alternative SM-DS during cascaded Event Record registration on ES15.
EXT_CARD_RESOURCE	Extended Card Resource Information according to ETSI TS 102 226 [14], coded as ASN.1 OCTET STRING. 'Number of installed application' value field is '00'.
EXT_SHA256_ECDSA	TLS extension data for "supported_signature_algorithms" set as a minimum of HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03).
FORWARDING_INDICATOR_ANY	Any boolean value (TRUE/FALSE)
FREE_MEM_OP_PROF_INSTALLED	Non-volatile memory (tag 0x82) available in the eUICC when two or more PROFILE_OPERATIONAL are installed
FREE_MEM_OP_PROF1_DELETED	Non-volatile memory (tag 0x82) available in the eUICC after PROFILE_OPERATIONAL1 deletion
FREE_MEM_OP_PROF1_INSTALLED	Non-volatile memory (tag 0x82) available in the eUICC when only PROFILE_OPERATIONAL1 is installed
FREE_MEMORY_NO_PROFILE	Non-volatile memory (tag 0x82) available in the eUICC when there is no Profile installed
FUNCTION_CALL_ID	The function call ID generated by the entity that calls the function
FUNCTION_REQ_ID	The function requester ID
INVALID_SM_DP_OID	SM-DP+ OID (as defined in section 1.3) not equal to #IUT_SM_DP_OID
INVALID_TRANSACTION_ID	A Transaction Identifier generated by the S_SM-DP+ or the S_SM-DS that SHALL be different from <S_TRANSACTION_ID> if exists. Otherwise, a random value is generated.
ISD_P_AID	The ISD-P AID newly created in the eUICC. This AID value is in the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to

Variable	Description
	0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1].
ISD_P_AID1	The ISD-P AID created in the eUICC for the PROFILE_OPERATIONAL1. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1].
ISD_P_AID2	The ISD-P AID created in the eUICC for the PROFILE_OPERATIONAL2. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1].
ISD_P_AID3	The ISD-P AID created in the eUICC for the PROFILE_OPERATIONAL3. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1].
ISD_P_AID4	The ISD-P AID created in the eUICC for the PROFILE_OPERATIONAL4. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1].
ISD_P_AIDX	An invalid ISD-P AID not present on the eUICC. This AID value is in the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00.
L	Exact length of the corresponding tag or of the remaining data.
LPA_RSP_CAPABILITY	`LpaRspCapability`, coded as ASN.1 BIT STRING to be used for indication of RSP capabilities of the Device.
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.

Variable	Description
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_NO_PROF1_RPR	The Sequence Number of the Notification RPM Package Result related to the PROFILE_OPERATIONAL1.
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.
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_AKA_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>
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.

Variable	Description
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)
R_EUICC_INFO1	The eUICC information EUICCInfo1 coded as an ASN.1 SEQUENCE, as defined in SGP.22 v3.1[2].
R_EUICC_INFO2	The eUICC information EUICCInfo2 coded as an ASN.1 SEQUENCE, as defined in SGP.22 v3.1[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.
REASON_CODE_ANY	Any Reason Code, as defined in SGP.22 [2] – section 5.2.6.2
REPLACE_S_KEYS_REQ_ENC	An element of secondSequenceOf87, consisting of #REPLACE_S_KEYS_REQ protected with <S_ENC> and <S_MAC> and encapsulated in a TLV with tag 0x87, up to a maximum size of 1020 bytes including the tag and length field.
RSP_SERVER_ADDRESS	RSP Server address in FQDN format where the operation corresponding to the Event can be processed.
S_ENC	SCP03T Encryption Session key (128 bits length) resulting from the key agreement with eUICC.
S_HASHED_CC	Hashed Confirmation Code generated by the LPA. When generated by the S_LPAd, the S_LPAd SHALL use #CONFIRMATION_CODE1 in the calculation unless otherwise specified.
S_HASHED_CC_ERROR	A random generated hash value of the Confirmation Code not equal to S_HASHED_CC.
S_INIT_MAC	SCP03T Initial MAC chaining value (128 bits length) resulting from the key agreement with eUICC.
S_MAC	SCP03T MACing Session key (128 bits length) resulting from the key agreement with eUICC.
S_MAC_CHAIN	Current MAC chaining value used for SCP03t BPP protection.

Variable	Description
S_SEL_TLS_CIPHER_SUITE	TLS cipher suite selected by the Server set as follows: o `TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256` if present in <TLS_CIPHER_SUITES>, otherwise o `TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256`.
S_SERVER_RSP_CAPABILITY_NONE	RspCapability of the S_SERVER is coded as ASN.1 BIT STRING (2 bits) to be used for indication of crlStaplingV3Support, eventListSigningV3Support and set this as no bits are set (NONE).
S_SESSION_CONTEXT	`{` `  serverSvn #S_SM_DP_SVN,` `  crlStaplingV3Used FALSE` `}`
S_SESSION_ID_SERVER	Random value of the TLS session_id in ServerHello which is different from <SESSION_ID_CLIENT>. This value is non-empty.
S_SM_DP+_SIGN	The S_SM-DP+ signature (smdpSign), computed using the #SK_S_SM_DPpb_ECDSA across the following data objects: • remoteOpId • transactionId • controlRefTemplate • smdpOtpk • euiccOtpk, as provided earlier in the prepareDownloadResponse data object
S_SM_DP+_SIGNATURE2	The ASN.1 OCTET STRING encoded SM-DP+ signature 2 (field smdpSignature2) computed using the private key related to the server certificate (field smdpCertificate) present in the PrepareDownloadRequest structure. This signature SHALL be generated across the following data objects: • smdpSigned2 • euiccSignature1 present in the AuthenticateServerResponse structure
S_SM_DP+_SIGNATURE3	The ASN.1 OCTET STRING encoded SM-DP+ signature 3 (field smdpSignature3) computed using the private key (Compute the `smdpSignature3` over the concatenated data object of `smdpSigned3` and `euiccSignature1` using the SK.DPauth.SIG related to the server certificate (field smdpCertificate) present in the InitiateAuthenticationOk Response structure. This signature SHALL be generated across the following data objects: • smdpSigned3 euiccSignature1 present in the AuthenticateServerResponse structure
S_SMDP_CHALLENGE	The SM-DP+ Challenge (serverChallenge) randomly chosen by the simulated SM-DP+ to be signed later by the eUICC for the eUICC authentication, coded as ASN.1 OCTET STRING of 16 bytes.
S_SMDP_SIGNATURE_INV	<S_SMDP_SIGNATURE1> NOT computed with the #SK_S_SM_DPauth_ECDSA but with the same length as a valid signature
S_SMDP_SIGNATURE1	The ASN.1 OCTET STRING encoded SM-DP+ signature (field serverSignature1) computed using the private key related to the

Variable	Description
	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>, sessionContext { serverSvn #IUT_RSP_VERSION_HIGHEST, crlStaplingV3Used TRUE }, serverRspCapability { crlStaplingV3Support, cancelForEmptySpnPnSupport } }
S_SMDP_SIGNED1_V3 (ServerSigned1)	{ transactionId <S_TRANSACTION_ID>, euiccChallenge <EUICC_CHALLENGE>, serverAddress #TEST_DP_ADDRESS1, serverChallenge <S_SMDP_CHALLENGE>, sessionContext { serverSvn #IUT_RSP_VERSION_HIGHEST, crlStaplingV3Used TRUE, euiccCiPKIdToBeUsedV3 <EUICC_CI_PK_ID_TO_BE_USED_V3> }, serverRspCapability { crlStaplingV3Support, cancelForEmptySpnPnSupport } } NOTE : select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING_V3>
S_SMDP_SIGNED2_OLD_KEYS	smdpSigned2 SmdpSigned2 ::= { transactionId <S_TRANSACTION_ID>, ccRequiredFlag FALSE, bppEuiccOtpk <OTPK_EUICC_ECKA> }
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

Variable	Description
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>, sessionContext { serverSvn #IUT_RSP_VERSION_HIGHEST, crlStaplingV3Used TRUE }, serverRspCapability { crlStaplingV3Support, eventListSigningV3Support } }
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>, sessionContext { serverSvn #IUT_RSP_VERSION_HIGHEST, crlStaplingV3Used TRUE }, serverRspCapability { crlStaplingV3Support, eventListSigningV3Support } }

Variable	Description
S_SMDS_SIGNED1_V3 (ServerSigned1)	{ transactionId <S_TRANSACTION_ID>, euiccChallenge <EUICC_CHALLENGE>, serverAddress #TEST_ROOT_DS_ADDRESS, serverChallenge <S_SMDS_CHALLENGE>, sessionContext { serverSvn #IUT_RSP_VERSION_HIGHEST, crlStaplingV3Used TRUE, euiccCiPKIdToBeUsedV3 <EUICC_CI_PK_ID_TO_BE_USED_V3> }, serverRspCapability { crlStaplingV3Support, eventListSigningV3Support } } NOTE : select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING_V3>
S_TRANSACTION_ID	The TransactionID (Unique Transaction Identifier) generated by the (S_)SM-DP+, or (S_)SM-DS which is used to uniquely identify the RSP session and to correlate the multiple ESXX request messages that belong to the same RSP session. This value (binary value) can start from 0x01 and can be increased by 1 each time a Profile is downloaded in the eUICC. 1-16 bytes (ASN.1 OCTET STRING).
SAH_SHA256_ECDSA	Signature And Hash Algorithm extension sent in the CertificateRequest message set as a minimum of: •HashAlgorithm sha256 (04) and •SignatureAlgorithm ecdsa (03).
SEL_TLS_CIPHER_SUITE	TLS cipher suite selected by the Server
SEQ_NUMBER	Sequence Number related to a Notification Metadata generated by the eUICC.
SERVER_CHALLENGE	Random value generated by the SM-XX server under test coded as ASN.1 OCTET STRING of 16 bytes which can be one of the following depending on the entity under test: • <SMDP_CHALLENGE> • <SMDS_CHALLENGE>
SERVER_CHALLENGE_2	Random value generated by the SM-XX server under test coded as ASN.1 OCTET STRING of 16 bytes which can be one of the following depending on the entity under test: • <SMDP_CHALLENGE_2> • <SMDS_CHALLENGE_2>
SERVER_FINISHED	The first protected message with the negotiated algorithms, keys, and secrets. It is the Hash of the concatenation of all the data from all messages in this handshake up to, but not including, this message i.e. all handshake messages starting at ClientHello up to, but not including, this Finished message itself. NOTE: ChangeCipherSpec messages, alerts, and any other record type are not handshake messages and are not included in the hash

Variable	Description
	computations. Also, HelloRequest messages are omitted from handshake hashes.
SERVER_NAME_V3	The v3-specific FQDN SHALL be the concatenation of the string "rsp3-" and the "base" UTF-8 encoded FQDN. v3-specific FQDN is computed from the "base" FQDN known from the RSP Server (e.g. an address retrieved from the eUICC, an address read from an Activation code)
SERVER_SIGNATURE1	Server signature (serverSignature1) which can be one of the following depending on the entity under test: • SM-DP+ signature (serverSignature1) generated over #SERVER_SIGNED1 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING • SM-DS signature (serverSignature1) generated over #SERVER_SIGNED1 using SK.DSauth.ECDSA, coded as ASN.1 OCTET STRING
SERVER_SIGNATURE1_2	SERVER signature (serverSignature1) which can be one of the following depending on the entity under test: • SM-DP signature (serverSignature1) generated over #SERVER_SIGNED1_2 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING • SM-DS signature (serverSignature1) generated over #SERVER_SIGNED1_2 using SK.DSauth.ECDSA, coded as ASN.1 OCTET STRING
SERVER_TLS_EPHEM_KEY	Server's ephemeral key and associated information.
SESSION_ID_CLIENT	Random or empty value of the TLS session_id in ClientHello.
SESSION_ID_RANDOM	Random value of the TLS session.
SHS	Shared Secret resulting from the key agreement with eUICC.
SM_DP+_SIGN	The SM-DP+ signature in ES8+/InitialiseSecureChannelRequest/smdpSign.
SMDP_CHALLENGE	Random value generated by the SM-DP+ coded as ASN.1 OCTET STRING of 16 bytes.
SMDP_CHALLENGE_2	Random value generated by the SM-DP+ coded as ASN.1 OCTET STRING of 16 bytes.
SMDP_CHALLENGE_INVALID	SM-DP+ Challenge randomly generated by the simulated SM-DP+ coded as ASN.1 OCTET STRING of 16 bytes not equal to <SMDP_CHALLENGE>.
SMDP_METADATA_SEG_MAC	An element of sequenceOf88, consisting of a segment of maximum size 1008 bytes protected with <S_MAC> and encapsulated in a TLV with tag 0x88, length <L>, up to a maximum size of 1020 bytes including the tag and length field.

Variable	Description
SMDP_SIGNATURE2	SM-DP+ signature (smdpSignature2) generated over smdpSigned2 using SK.DPauth.ECDSA, coded as ASN.1 OCTET STRING
SMDS_CHALLENGE	Random value generated by the SM-DS coded as ASN.1 OCTET STRING of 16 bytes.
SMDS_CHALLENGE_2	Random value generated by the SM-DS coded as ASN.1 OCTET STRING of 16 bytes.
SMDS_CHALLENGE_INVALID	SM-DS Challenge randomly generated by the simulated SM-DS coded as ASN.1 OCTET STRING of 16 bytes not equal to <SMDS_CHALLENGE>.
STORE_DATA_BLOCK_NUM	The STORE DATA block number coded sequentially from 0x00 to 0xFF. If the value 0xFF has been reached and more STORE DATA commands are needed to complete the transfer, the numbering restarts and the next STORE DATA block number is set to 0x00.
SUBJECT_CODE_ANY	Any Subject Code, as defined in SGP.22 [2] – section 5.2.6.1
TBS_EUICC_NOTIF_SIG	The eUICC signature generated over tbsOtherNotification. NotificationMetadata, coded as ASN.1 OCTET STRING.
TLS_CIPHER_SUITES	TLS cipher suite list supported by LPAd or the Client (SM-DP+ or SM-DS) under test.
TRANSACTION_ID_2	A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session as OCTET STRING of up to 16 bytes.
TRANSACTION_ID_AC	A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session used by the AuthenticateClient function as OCTET STRING of up to 16 bytes.
TRANSACTION_ID_GBPP	A unique Transaction ID generated by an SM-DP+ within the scope and lifetime of each SM-DP+ to uniquely identify the ongoing RSP session used by the GetBoundProfilePackage function as OCTET STRING of up to 16 bytes.
TRANSACTION_ID_IA	A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or an SM-DS to uniquely identify the ongoing RSP session used by the InitiateAuthentication function as OCTET STRING of up to 16 bytes.
TRANSACTION_ID_ISC	A unique Transaction ID generated by an SM-DP+ within the scope and lifetime of each SM-DP+ to uniquely identify the ongoing RSP session used by the InitialiseSecureChannelRequest function as OCTET STRING of up to 16 bytes.
TRANSACTION_ID_SIGNED	A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session as OCTET STRING of up to 16 bytes signed as part of #SERVER_SIGNED1

Variable	Description
TRANSACTION_ID_SIGNED_2	A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session as OCTET STRING of up to 16 bytes signed as part of #SERVER_SIGNED1
TRANSACTION_ID_SIGNED_AC	A unique Transaction ID generated by an SM-DP+ or an SM-DS within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session used by the AuthenticateClient function as OCTET STRING of up to 16 bytes.
TRANSACTION_ID_SIGNED_IA	A unique Transaction ID generated by an SM-DP+ or an SM-SD within the scope and lifetime of each SM-DP+ or SM-DS to uniquely identify the ongoing RSP session used by the InitiateAuthentication function as OCTET STRING of up to 16 bytes.
TRE_PROPERTIES	The value of the treProperties field in EUICCInfo2.
TRE_REFERENCE	The value of the treProductReference field in EUICCInfo2.
UPP_OP_PROF1_SEG	A segment of the #UPP_OP_PROF1, with a maximum size of 1007 bytes.
UPP_OP_PROF2_SEG	A segment of the #UPP_OP_PROF2, with a maximum size of 1007 bytes.

Annex C Methods And Procedures

This section describes methods and procedures used in the interfaces compliance test cases. They are part of test cases and SHALL not be executed in standalone mode.

C.1 Methods

If the method is used in the “expected result” column, all parameters SHALL be verified by the simulated entity (test tool). If the method is used in the “Sequence / Description” column, the command SHALL be generated by the simulated entity.

Method	MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND
Description	This Method can be used for creating Authenticate Client response with LoadRpmPackageRequest containing an rpmPackage with a single RPM command.
Parameter(s)	• paramComandDetailsChoice: The rpmCommandDetails choice (Mandatory) • paramTransactionId: The Transaction Id (Mandatory) • paramIccidValue: The ICCID within RPM Command (Mandatory) • paramSmdpSignature3: The SM-DP+ Signature3 (Mandatory) • paramUpdateMetadataRequest: The UpdateMetadataRequest if required • paramDpiRpmValue: The dpiRpm within RPM Command if required • paramRpmPending if required Parameters paramUpdateMetadataRequest and paramDpiRpmValue shall be passed and can be empty depend on the paramComandDetailsChoice.
Details	`{` `"header" : {` `"functionExecutionStatus" : {` `"status" : "Executed-Success"` `}` `},` `"transactionId" : paramTransactionId,` `"smdpSigned3" : {` `transactionId paramTransactionId,` `rpmPackage {` `MTD_RPM_PKG_REQ_FOR_SINGLE_CMND(` `paramComandDetailsChoice,` `paramIccidValue,` `paramUpdateMetadataRequest,` `paramDpiRpmValue,` `paramRpmPending` `)` `}` `},` `"smdpSignature3" : paramSmdpSignature3` `}`

Method	MTD_AUTH_CLIENT_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO
Description	This Method can be used for creating Authenticate Client response with LoadRpmPackageRequest containing an rpmPackage with a single RPM command ListProfileInfo and Search Criteria as ICCID.

Parameter(s)	• paramTransactionId: The Transaction Id (Mandatory) • paramIccidValue: The ICCID within RPM Command (Mandatory) • paramSmdpSignature3: The SM-DP+ Signature3 (Mandatory) • paramProfileOwnerOidValue: The ProfileOwnerOid for seachCriteria in ListProfileInfo if required • paramTagListValue: The TagList if required Parameters paramTagListValue and paramProfileOwnerOidValue shall be passed and can be empty.
Details	`{` `"header" : {` `"functionExecutionStatus" : {` `"status" : "Executed-Success"` `}` `},` `"transactionId" : paramTransactionId,` `"smdpSigned3" : {` `transactionId paramTransactionId,` `rpmPackage {` `MTD_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO(` `paramIccidValue,` `paramTagListValue,` `paramProfileOwnerOidValue` `)` `}` `},` `"smdpSignature3" : paramSmdpSignature3` `}`

Method	MTD_AUTHENTICATE_CLIENT
Description	Generates or verifies the JSON formatted AuthenticateClient request
Parameter(s)	• paramTransactionId: random 16 byte identifier encoded as String Hexadecimal. • paramAuthenticateServerResponse: server authentication response structured as ASN.1 encoded as base 64.
Details	JSON body `{` `"transactionId" : paramTransactionId,` `"authenticateServerResponse" :` `paramAuthenticateServerResponse` `}`

Method	MTD_CANCEL_ORDER
Description	Sends and checks the JSON formatted CancelOrder request
Parameter(s)	• paramFunctionRequesterId • paramFunctionCallId • paramIccid: identification of the targeted profile (mandatory) • paramEID: EID of the targeted eUICC (conditional) • paramMatchingId: matching ID generated by the Operator (conditional) • paramProfileStatus: final Profile status indicator (mandatory)

Details	JSON requestHeader { "header" : { "functionRequesterIdentifier" : "paramFunctionRequesterId", "functionCallIdentifier" : "paramFunctionCallId" } JSON body { "iccid" : paramIccid "eid" : paramEID, "matchingId" : paramMatchingId "profileStatus" : paramProfileStatus } } Note: if some of the value of the parameters above are not provided, those parameters are not included as part of JSON body

Method	MTD_CANCEL_SESSION
Description	Sends or verifies the JSON formatted CancelSession request
Parameter(s)	• paramTransactionId: random 16 byte identifier. • paramCancelSessionResponse: eUICC information structured as ASN.1 encoded as base 64.
Details	JSON body { "transactionId" : paramTransactionId, "cancelSessionResponse" : paramCancelSessionResponse }

Method	MTD_CHECK_SMS_POR
Description	Check the content of the SMS POR containing the response of the ES6.UpdateMetadata request
Parameter(s)	paramExpectedSW: the expected Status Word of the last STORE DATA command
Details	Parse and retrieve the SCP80 response packet from the SMS. SCP80 response status code SHALL be equal to 0x00 – POR OK. The additional data from the response packet SHALL be formatted as an expanded structure with definite length as defined in ETSI TS 102 226 [14] and contains the following TLV: AB <L> 80 <L> <NB_EXECUTED_C_APDUS> -- *Number of executed C-APDUs* 23 <L> 00 90 00 -- *R-APDU of the INSTALL FOR PERSONALIZATION command*

	23 <L> paramExpectedSW -- *SW of the last STORE DATA command executed* <NB_EXECUTED_C_APDUS> SHALL be equal to the number of executed C-APDUs (i.e. one INSTALL FOR PERSONALIZATION + n STORE DATA command(s))

Method	MTD_ CONFIRM_ORDER
Description	Sends and checks the JSON formatted ConfirmOrder request
Parameter(s)	• paramFunctionRequesterId • paramFunctionCallId • paramIccid: identification of the targeted profile (mandatory) • paramEID: EID of the targeted eUICC (conditional) • paramMatchingId: matching ID generated by the Operator (optional) • paramConfirmationCode: confirmation code provided by the Operator (optional) • paramSmdsAddress: SM-DS to be used for event registration (conditional) • paramReleaseFlag: boolean indicating if the profile shall be released (mandatory)
Details	JSON requestHeader `{` `"header" : {` `"functionRequesterIdentifier" : "paramFunctionRequesterId",` `"functionCallIdentifier" : "paramFunctionCallId"` `}` JSON body `{` `"iccid" : paramIccid` `"eid" : paramEID,` `"matchingId" : paramMatchingId` `"confirmationCode" : paramConfirmationCode` `"smdsAddress" : paramSmdsAddress "releaseFlag" : paramReleaseFlag` `}` `}` Note: if some of the value of the parameters above are not provided, those parameters are not included as part of JSON body

Method	MTD_DELETE_EVENT
Description	Sends and checks the JSON formatted DeleteEvent request
Parameter(s)	• paramFunctionRequesterId: identification of the function requester. • paramFunctionCallId: identification of the function call. • paramEID: EID of the targeted eUICC • paramEventId: unique Identification of the Event to be registered
Details	JSON requestHeader `{`

	"header" : { "functionRequesterIdentifier" : "paramFunctionRequesterId", "functionCallIdentifier" : "paramFunctionCallId" } JSON body { "eid" : paramEID, "eventId" : paramEventId } }

Method	MTD_DISABLE_PROFILE
Description	Generate the ASN.1 DisableProfileRequest structure according to the input parameters.
Parameter(s)	• paramIccidValue: The ICCID of the Profile to Disable (optional) • paramIsdpAidValue: The ISD-P AID of the Profile to Disable (optional) • paramRefreshFlag: Boolean, TRUE if refreshFlag SHALL be set, FALSE otherwise Either paramIccidValue or paramIsdpAidValue is passed as a parameter.
Details	IF paramIccidValue is provided Then req DisableProfileRequest::= { profileIdentifier iccid : paramIccidValue, refreshFlag paramRefreshFlag } Else req DisableProfileRequest::= { profileIdentifier isdpAid : paramIsdpAidValue, refreshFlag paramRefreshFlag } End if

Method	MTD_ DOWNLOAD_ORDER
Description	Sends and checks the JSON formatted DownloadOrder request
Parameter(s)	• paramFunctionRequesterId • paramFunctionCallId • paramEID: EID of the targeted eUICC□□optional□ • paramIccid: identification of the targeted profile (conditional) • paramProfileType: identification of the targeted profile type (conditional)
Details	JSON requestHeader { "header" : { "functionRequesterIdentifier" :"paramFunctionRequesterID", "functionCallIdentifier" :"paramFunctionCallID" }

	JSON body `{` `"eid" : paramEID,` `"iccid" : paramIccid` `"profileType" : paramProfileType` `}` `}` Note: if some of the value of the parameters above are not provided, those parameters are not included as part of JSON body

Method	MTD_ENABLE_PROFILE
Description	Generate the ASN.1 EnableProfileRequest structure according to the input parameters.
Parameter(s)	• paramIccidValue: The ICCID of the Profile to Disable (optional) • paramIsdpAidValue: The ISD-P AID of the Profile to Disable (optional) • paramRefreshFlag: Boolean, TRUE if refreshFlag SHALL be set, FALSE otherwise Either paramIccidValue or paramIsdpAidValue is passed as a parameter.
Details	IF paramIccidValue is provided Then `req EnableProfileRequest ::= {` `profileIdentifier iccid : paramIccidValue,` `refreshFlag paramRefreshFlag` `}` Else `req EnableProfileRequest ::= {` `profileIdentifier isdpAid : paramIsdpAidValue,` `refreshFlag paramRefreshFlag` `}` End if

Method	MTD_DELETE_PROFILE
Description	Generate the ASN.1 DeleteProfileRequest structure according to the input parameters.
Parameter(s)	• paramIccidValue: The ICCID of the Profile to Delete (optional) • paramIsdpAidValue: The ISD-P AID of the Profile to Delete (optional) Either paramIccidValue or paramIsdpAidValue is passed as a parameter.
Details	IF paramIccidValue is provided Then `req DeleteProfileRequest ::= iccid : paramIccidValue` Else `req DeleteProfileRequest ::= isdpAid : paramIsdpAidValue` End if

Method	MTD_GET_PROFILE_INFO
Description	Generate the ASN.1 ProfileInfoListRequest according to the input parameters.
Parameter(s)	• paramIccidValue: The ICCID of the Profile • paramIsdpAidValue: The ISD-P AID of the Profile Either paramIccidValue or paramIsdpAidValue is passed as a parameter.
Details	IF paramIccidValue is provided Then `req ProfileInfoListRequest::= {` `  searchCriteria iccid: paramIccidValue` `}` Else `req ProfileInfoListRequest::= {` `  searchCriteria isdpAid: paramIsdpAidValue` `}` End If

Method	MTD_GENERATE_HASHED_CC
Description	Generate an Hashed Confirmation Code based on the Confirmation Code and the Transaction ID given in parameter.
Parameter(s)	• paramConfirmationCode: The Confirmation Code (plain) • paramTransactionId: The Transaction ID (plain)
Details	Generate a SHA-256 of the paramConfirmationCode. Concatenate the obtained hash value with the paramTransactionId. Generate and return a SHA-256 of these two concatenated elements.

Method	MTD_GET_BPP
Description	Generates or verifies the JSON formatted GetBoundProfilePackage request
Parameter(s)	• paramTransactionId: random 16 byte identifier. • paramPrepareDownloadResponse structured as ASN.1 encoded as base 64.
Details	JSON body `{` `"transactionId" : paramTransactionId,` `"prepareDownloadResponse" : paramPrepareDownloadResponse` `}`

Method	MTD_HANDLE_NOTIF
Description	Generates or verifies the JSON formatted HandleNotification request
Parameter(s)	paramPendingNotification: PendingNotification data object
Details	JSON body `{` `"pendingNotification" : paramPendingNotification`

	}

Method	MTD_HTTP_REQ
Description	Sends or verifies a secured HTTP request message delivering a JSON object payload using a network to an off-card entity.
Parameter(s)	• paramServerAddress: Target Server address • paramFunctionPath: Function path • paramRequestMessage: JSON Request message
Details	HTTP POST paramFunctionPath HTTP/1.1 Host: paramServerAddress User-Agent: See NOTE 1 X-Admin-Protocol:gsma/rsp/v<2.1.0> Content-Type: application/json;charset=UTF-8 Content-Length: <L> paramRequestMessage NOTE 1: The "User-Agent" field may contain additional information after a semicolon. The additional information shall not be checked. The HTTP POST request may contain additional header fields. These shall not be checked.

Method	MTD_HTTP_RESP
Description	Sends or verifies a secured HTTP response message delivering a JSON object payload using a network to an off-card entity.
Parameter(s)	• paramResponseMessage: JSON Response message
Details	HTTP/1.1 200 (OK) X-Admin-Protocol: gsma/rsp/v<2.1.0> Content-Type: application/json;charset=UTF-8 Content-Length: <L> paramResponseMessage The HTTP response may contain additional header fields. These shall not be checked.

Method	MTD_INITIATE_AUTHENTICATION
Description	Generates or verifies the JSON formatted Initiate Authentication request on ES9+ or ES11 as applicable.
Parameter(s)	• paramEUICCChallenge: random 16 byte challenge coded as base 64 • paramEUICCInfo1: eUICC information structured coded as base 64 • paramServerAddress: FQDN of the Server. • paramLpaRspCapability: value of lpaRspCapability, supported by LPA

Details	JSON body `{` `"euiccChallenge" : paramEUICCChallenge,` `"euiccInfo1" : paramEUICCInfo1,` `"smdpAddress" : paramServerAddress` `"lpaRspCapability " : paramLpaRspCapability` `}`

Method	MTD_REGISTER_EVENT
Description	Send or checks the JSON formatted RegisterEvent request
Parameter(s)	• paramFunctionRequesterId: identification of the function requester. • paramFunctionCallId: identification of the function call. • paramEID: EID of the targeted eUICC • paramRspServerAddress: Address of the Server sending the RegisterEvent formatted as FQDN • paramEventId: unique Identification of the Event to be registered • paramForwardingIndicator: TRUE if registration has to be made to the Root SM-DS; FALSE if this is not to be made to the Root SM-DS
Details	JSON requestHeader `{` `"header" : {` `"functionRequesterIdentifier" : "paramFunctionRequesterId",` `"functionCallIdentifier" : "paramFunctionCallId"` `}` JSON body `{` `"eid" : paramEID,` `"rspServerAddress" : paramRspServerAddress,` `"eventId" : paramEventId,` `"forwardingIndicator" : paramForwardingIndicator` `}` `}`

Method	MTD_REMOVE_NOTIF
Description	Constructs the command data for RemoveNotificationFromList
Parameter(s)	• paramSeqNumber: the sequence number to be removed
Details	`request NotificationSentRequest ::= {` `seqNumber paramSeqNumber` `}`

Method	MTD_RESP_RPR_FOR_SINGLE_CMND
Description	This Method can be used for verifying the rpmPackageResult resulted from a single RPM command.

Parameter(s)	• paramRpmCommandResultDataChoice: The rpmCommandDetails choice (Mandatory) • paramTransactionId: The Transaction Id (Mandatory) • paramIccidValue: The ICCID within the RPM Command received • paramRprErrorMask: The RPR error mask as defined below. (Mandatory) • paramNotificationMetadata: The notification Metadata. • paramSmdpOid: The SM-DP+ OID. • paramProfileInfoValue: The Profile Info response • paramPcmpAddressValue: The pcmp Address. • paramRpmCommandResultError: The Rpm Command Result error paramRprErrorMask is defined as, 0 – OK result (RPM command successful) 1 - RpmCommandResultDataError is present 2 - RpmProcessingTerminated is present 3 - LoadRpmPackageErrorCode is present 4 - LoadRpmPackageErrorCodeNotSigned is present Parameters paramNotificationMetadata and paramSmdpOid are mandatory for all the cases except for paramRprErrorMask is 4 (loadRpmPackageErrorCodeNotSigned). Parameters paramProfileInfoValue and paramPcmpAddressValue shall be passed and can be empty depend on the paramRpmCommandResultDataChoice for paramRprErrorMask is 0 (the successful responses). Parameter paramRpmCommandResultError is mandatory if paramRprErrorMask > 0 (all error cases) and can be empty for paramRprErrorMask = 0

Details	If paramRprErrorMask = 0 Then -- *OK response* If paramRpmCommandResultDataChoice = enableResult or disableResult or deleteResult or updateMetadataResult Then `response LoadRpmPackageResult ::=` `loadRpmPackageResultSigned : {` `loadRpmPackageResultDataSigned {` `transactionId paramTransactionId,` `notificationMetadata paramNotificationMetadata,` `smdpOid paramSmdpOid,` `finalResult rpmPackageExecutionResult : {` `{` `iccid paramIccidValue,` `rpmCommandResultData` `paramRpmCommandResultDataChoice : {` `paramRpmCommandResultDataChoice ok` `}` `}` `}` `},` `euiccSignRPR <EUICC_SIGN_RPR>` `}` Else if paramRpmCommandResultDataChoice = listProfileInfoResult Then `response LoadRpmPackageResult ::=` `loadRpmPackageResultSigned : {` `loadRpmPackageResultDataSigned {` `transactionId paramTransactionId,` `notificationMetadata paramNotificationMetadata,` `smdpOid paramSmdpOid,` `finalResult rpmPackageExecutionResult : {` `{` `rpmCommandResultData listProfileInfoResult : {` `profileInfoListOk : {paramProfileInfoValue}` `}` `}` `}` `},` `euiccSignRPR <EUICC_SIGN_RPR>` `}` Else if paramRpmCommandResultDataChoice = contactPcmpResult Then `response LoadRpmPackageResult ::=` `loadRpmPackageResultSigned : {` `loadRpmPackageResultDataSigned {` `transactionId paramTransactionId,` `notificationMetadata paramNotificationMetadata,` `smdpOid paramSmdpOid,` `finalResult rpmPackageExecutionResult : {` `{` `iccid paramIccidValue,` `rpmCommandResultData contactPcmpResult : {` `contactPcmpResponseOk : {` `pcmpAddress paramPcmpAddressValue }` `}` `}` `}` `},` `euiccSignRPR <EUICC_SIGN_RPR>` `}` Else If paramRprErrorMask = 1 Then -- *rpmCommandResultDataError is present*

```
If paramRpmCommandResultDataChoice = listProfileInfoResult Then
    response LoadRpmPackageResult ::=
    loadRpmPackageResultSigned : {
      loadRpmPackageResultDataSigned {
        transactionId paramTransactionId,
        notificationMetadata paramNotificationMetadata,
        smdpOid paramSmdpOid,
        finalResult rpmPackageExecutionResult : {
          {
             rpmCommandResultData listProfileInfoResult : {
                profileInfoListError :
{ paramRpmCommandResultError }
             }
          }
        }
      },
      euiccSignRPR <EUICC_SIGN_RPR>
    }
Else if paramRpmCommandResultDataChoice = contactPcmpResult Then
    response LoadRpmPackageResult ::=
    loadRpmPackageResultSigned : {
      loadRpmPackageResultDataSigned {
        transactionId paramTransactionId,
        notificationMetadata paramNotificationMetadata,
        smdpOid paramSmdpOid,
        finalResult rpmPackageExecutionResult : {
          {
             rpmCommandResultData contactPcmpResult: {
                contactPcmpResponseError:
{ paramRpmCommandResultError }
             }
          }
        }
      },
      euiccSignRPR <EUICC_SIGN_RPR>
    }
Else
    response LoadRpmPackageResult ::=
    loadRpmPackageResultSigned : {
       loadRpmPackageResultDataSigned {
            transactionId paramTransactionId,
            notificationMetadata paramNotificationMetadata,
            smdpOid paramSmdpOid,
            finalResult rpmPackageExecutionResult : {
               {
                  iccid paramIccidValue,
                  rpmCommandResultData
              paramRpmCommandResultDataChoice : {
                        paramRpmCommandResultDataChoice
              paramRpmCommandResultError
                  }
               }
            }
        },
        euiccSignRPR <EUICC_SIGN_RPR>
    }
Else If paramRprErrorMask = 2 Then  --  rpmProcessingTerminated is present
    response LoadRpmPackageResult ::=
    loadRpmPackageResultSigned : {
```

	loadRpmPackageResultDataSigned { transactionId paramTransactionId, notificationMetadata paramNotificationMetadata, smdpOid paramSmdpOid, finalResult rpmPackageExecutionResult : { { rpmCommandResultData rpmProcessingTerminated : { paramRpmCommandResultError } } } }, euiccSignRPR <EUICC_SIGN_RPR> } Else If paramRprErrorMask = 3 Then -- *loadRpmPackageErrorCode is present* response LoadRpmPackageResult ::= loadRpmPackageResultSigned : { loadRpmPackageResultDataSigned { transactionId paramTransactionId, notificationMetadata paramNotificationMetadata, smdpOid paramSmdpOid, finalResult loadRpmPackageErrorCode : { paramRpmCommandResultError } }, euiccSignRPR <EUICC_SIGN_RPR> } Else If paramRprErrorMask = 4 Then -- *loadRpmPackageErrorCodeNotSigned is present* response LoadRpmPackageResult ::= loadRpmPackageResultNotSigned : { transactionId paramTransactionId, loadRpmPackageErrorCodeNotSigned paramRpmCommandResultError } End if

Method	MTD_RETRIEVE_NOTIF_SEQ_NUM
Description	Constructs the command data for RetrieveNotificationsList filtered by sequence number
Parameter(s)	• paramSeqNumber: the sequence number to be retrieved
Details	request RetrieveNotificationsListRequest ::= { searchCriteria seqNumber paramSeqNumber }

Method	MTD_RPM_PKG_REQ_FOR_SINGLE_CMND
Description	This Method can be used for creating rpmPackage with single RPM command.
Parameter(s)	• paramComandDetailsChoice: The rpmCommandDetails choice (Mandatory) • paramIccidValue: The ICCID within RPM Command (Mandatory) • paramUpdateMetadataRequest: The UpdateMetadataRequest if required • paramDpiRpmValue: The dpiRpm within RPM Command if required • paramRpmPending if required

	Parameters paramUpdateMetadataRequest and paramDpiRpmValue shall be passed or shall be empty depend on the paramComandDetailsChoice. NULL shall be passed as parameter paramRpmPending if RPM command is triggered due to a rpmPending indication from a previous session. Otherwise it is not required.
Details	If paramComandDetailsChoice = enable or disable or delete Then { continueOnFailure NULL, rpmCommandDetails paramComandDetailsChoice : { iccid paramIccidValue } } Else if paramComandDetailsChoice = updateMetadata Then { continueOnFailure NULL, rpmCommandDetails updateMetadata : { iccid paramIccidValue updateMetadataRequest {paramUpdateMetadataRequest} } } Else if paramComandDetailsChoice = contactPcmp Then If paramDpiRpmValue is not present Then { continueOnFailure NULL, rpmCommandDetails contactPcmp : { iccid paramIccidValue } } Else { continueOnFailure NULL, rpmCommandDetails contactPcmp : { iccid paramIccidValue, dpiRpm paramDpiRpmValue } } End if

Method	MTD_RPM_PKG_REQ_FOR_SINGLE_CMND_LIST_PROFILE_INFO
Description	This Method can be used for creating rpmPackage with single RPM command ListProfileInfo and Search Criteria as ICCID.
Parameter(s)	• paramIccidValue: The ICCID within RPM Command (Mandatory) • paramTagListValue The TagList if required • paramProfileOwnerOidValue: The ProfileOwnerOid for seachCriteria in ListProfileInfo if required Parameters paramTagListValue and paramProfileOwnerOidValue shall be passed and can be empty.

Details	If paramTagListValue is not present Then If paramProfileOwnerOidValue is not present Then { continueOnFailure NULL, rpmCommandDetails listProfileInfo : { searchCriteria iccid : {paramIccidValue} } } Else { continueOnFailure NULL, rpmCommandDetails listProfileInfo : { searchCriteria profileOwnerOid : {paramProfileOwnerOidValue} } } Else If paramProfileOwnerOidValue is not present Then { continueOnFailure NULL, rpmCommandDetails listProfileInfo : { searchCriteria iccid : {paramIccidValue}, tagList { paramTagListValue} } } Else if { continueOnFailure NULL, rpmCommandDetails listProfileInfo : { searchCriteria profileOwnerOid : {paramProfileOwnerOidValue}, tagList { paramTagListValue} } } End if

Method	MTD_SELECT
Description	Generates the SELECT command as defined in GlobalPlatform Card Specification [6].
Parameter(s)	• paramAID: the AID to select
Details	- CLA = 0x or 4x (x = <CHANNEL_NUMBER>) - INS = A4 - P1 = 04 - P2 = 00 - LC = <L> - paramAID - LE = 00

Method	MTD_SEND_SMS_PP
Description	Generate and send an envelope SMS-PP download to the MNO-SD
Parameter(s)	• paramApdusList: the list of APDUs (plain) to send

Details	Generate and send the following envelope: `80 C2 00 00 <L>` `  D1 <L>` `  02 02 83 81 -- Device identity Tag` `  06 07 91 33 86 09 40 00 F0 -- Address Tag (TON/NPI/..)` `  0B <L> -- SMS TPDU` `  44 -- SMS-DELIVER` `  05 85 02 13 F2 -- TP-Originating-Address` `  7F -- TP-Protocol-Identifier` `  F6 -- TP-Data-Coding-Scheme` `  71 30 12 41 55 74 40 -- TP-Service-Centre-Time-Stamp` `  <L> -- TP-User-Data-Length` `  02 -- User-Data-Header-Length` `  70 -- IEIa` `  00 -- IEIDLa` `  <L> -- Command Packet Length (2 bytes)` `  <L> -- Command Header Length (1 byte)` `  12 21 -- SPI` `  00 -- KIC` `  15 -- KID (SCP80 Keyset version 0x01 in Triple DES)` `  B2 01 00 -- MNO-SD TAR` `  <MNO_SCP80_COUNTER>` `  00 -- Padding Counter` `  <CC> -- Cryptographic checksum` `  <C_APDUS_SCRIPT> -- Command APDUs script` <C_APDUS_SCRIPT> SHALL contain the paramApdusList (i.e. each APDU is named <APDU1>; <APDU2>; …; <APDUn> here after) formatted as an expanded structure with definite length as defined in ETSI TS 102 226 [14]: `AA <L>` `  22 <L> <APDU1>` `  22 <L> <APDU2>` `  …` `  22 <L> <APDUn>` The Cryptographic checksum <CC> SHALL be generated in Triple DES (outer-CBC mode using two different keys) with the #MNO_SCP80_AUTH_KEY as defined in ETSI TS 102 225 [13]. If the command packet length is higher than 140 bytes, it SHALL be sent over several envelopes: SMS concatenation as defined in 3GPP TS 23.040 [22] SHALL be used.

Method	MTD_STORE_DATA
Description	Generates the STORE DATA command (Case 4) as defined in GlobalPlatform Card Specification [6].
Parameter(s)	• paramCommandData: the command data
Details	- CLA = 8x or Cx (x = <CHANNEL_NUMBER>)

	- INS = E2 - P1 = 91 - P2 = 00 - LC = <L> - paramCommandData - LE = 00

Method	MTD_STORE_DATA_Case3
Description	Generates the STORE DATA command (Case3) as defined in GlobalPlatform Card Specification [6].
Parameter(s)	• paramCommandData: the command data
Details	- CLA = 8x or Cx (x = <CHANNEL_NUMBER>) - INS = E2 - P1 = 90 - P2 = 00 - LC = <L> - paramCommandData

Method	MTD_STORE_DATA_SCRIPT
Description	Generate (multiple) STORE DATA command(s) by breaking the data into smaller components (if needed) for transmission.
Parameter(s)	• paramTLVDataToTransmit: TLVs array or single TLV to transfer to the eUICC • paramCase4Command (optional parameter, default value = TRUE): TRUE if the APDU is a Case 4 command, FALSE if the APDU is a Case 3 command
Details	For each element of paramTLVDataToTransmit If the size of the element is greater than 255 bytes, split the element in several blocks of 255 bytes. The last block MAY be shorter. Each block is named <DATA_SUB_PART> here after. If the element is up to 255 bytes, <DATA_SUB_PART> contains the value of the element. The bit b1 of P1 in the STORE DATA commands is named <B1_P1> here after and is defined as below: If paramCase4Command = TRUE Then <B1_P1> = 1 Else <B1_P1> = 0 End If Set <STORE_DATA_BLOCK_NUM> to 0 For each <DATA_SUB_PART>

	If <DATA_SUB_PART> is an intermediate part, generate the following STORE DATA: - CLA = 8x or Cx (x = <CHANNEL_NUMBER>) - INS = E2 - P1 = 1x (x = <B1_P1>) - P2 = <STORE_DATA_BLOCK_NUM> - LC = <L> - <DATA_SUB_PART> - LE = 00 -- present only if paramCase4Command = TRUE If <DATA_SUB_PART> is the last part, generate the following STORE DATA: - CLA = 8x or Cx (x = <CHANNEL_NUMBER>) - INS = E2 - P1 = 9x (x = <B1_P1>) - P2 = <STORE_DATA_BLOCK_NUM> - LC = <L> - <DATA_SUB_PART> - LE = 00 -- present only if paramCase4Command = TRUE Increase the <STORE_DATA_BLOCK_NUM> by 1 End End

Method	MTD_TEST_ES8+_GET_BPP_PPK
Description	Tests the received boundProfilePackage element according to #R_GET_BPP_RESP_OP1_PPK
Parameter(s)	• paramResponse the response to GetBoundProfilePackage • paramS_MAC the 128 bit SCP03t MACing Session key • paramS_ENC the 128 bit SCP03t Encryption Session key • paramPPK_MAC the 128 bit Profile Protection MACing Key • paramPPK_ENC the 128 bit Profile Protection Encryption Key • paramMetaData the ASN.1 StoreMetadataRequest element associated to a RSP profile
Details	Parse paramResponse into #R_GET_BPP_RESP_OP1_PPK and perform the following tests: • Verify that each element in firstSequenceOf87, sequenceOf88, secondSequenceOf87 and sequenceOf86 has a total length (including tag and length fields) of 1020 or less • Verify the integrity of each element in firstSequenceOf87, sequenceOf88 and secondSequenceOf87 using paramS_MAC • Verify that <TRANSACTION_ID_ISC> in #INIT_SC_PROF1 matches <S_TRANSACTION_ID> • Verify the validity of smdpSign <SM_DP+_SIGN> in #INIT_SC_PROF1 using #PK_SM_DPpb_ECDSA • Retrieve #CONF_ISDP_PROF1_SMDP from <CONF_ISDP_PROF1_ENC> using paramS_ENC and validate the content of #CONF_ISDP_PROF1_SMDP • Construct the complete metadata element from the <SMDP_METADATA_SEG_MAC> segment(s) and verify that it matches paramMetaData • Retrieve #REPLACE_S_KEYS_REQ from <REPLACE_S_KEYS_REQ_ENC> using paramS_ENC and validate the content of #REPLACE_S_KEYS_REQ

	• Verify that the lengths of paramPPK_ENC and paramPPK_MAC in #REPLACE_S_KEYS_REQ are each 16 bytes • Verify the integrity of each <PPP_OP_PROF1_SEG_PPK> element using paramPPK_MAC • Retrieve the <UPP_OP_PROF1_SEG> segment(s) from the <PPP_OP_PROF1_SEG_PPK> segment(s) using paramPPK_ENC, construct the complete Profile from the <UPP_OP_PROF1_SEG> segment(s), then verify that the complete Profile matches #UPP_OP_PROF1

Method	MTD_TEST_ES8+_GET_BPP_SK
Description	Tests the received boundProfilePackage element according to #R_GET_BPP_RESP_OP1_SK
Parameter(s)	• paramResponse the response to GetBoundProfilePackage • paramS_MAC the 128 bit SCP03t MACing Session key • paramS_ENC the 128 bit SCP03t Encryption Session key • paramMetaData the ASN.1 StoreMetadataRequest element associated to a RSP profile
Details	Parse paramResponse into #R_GET_BPP_RESP_OP1_SK and perform the following tests: • Verify that each element in firstSequenceOf87, sequenceOf88 and sequenceOf86 has a total length (including tag and length fields) of 1020 or less • Verify the integrity of each element in firstSequenceOf87, sequenceOf88 and sequenceOf86 using paramSMAC • Verify that <TRANSACTION_ID_ISC> in #INIT_SC_PROF1 matches <S_TRANSACTION_ID> • Verify the validity of smdpSign <SM_DP+_SIGN> in #INIT_SC_PROF1 using #PK_SM_DPpb_ECDSA • Retrieve #CONF_ISDP_PROF1_SMDP from <CONF_ISDP_PROF1_ENC> using paramS_ENC and validate the content of #CONF_ISDP_PROF1_SMDP • Construct the complete metadata element from the <SMDP_METADATA_SEG_MAC> segment(s) and verify that it matches paramMetaData • Retrieve the <UPP_OP_PROF1_SEG> segment(s) from the <PPP_OP_PROF1_SEG_SK> segment(s) using paramS_ENC, then construct the complete Profile from the <UPP_OP_PROF1_SEG> segment(s), then verify that the complete Profile matches #UPP_OP_PROF1

Method	MTD_TLS_CLIENT_KEY_EXCH_ETC
Description	Finalizes the Transport Layer Security (TLS) handshake in Server authentication mode on ES9+, or ES11 (Client side).
Parameter(s)	• paramClientKeyExchange: ClientKeyExchange message
Details	Sends the session key information in TLS ClientKeyExchange message, ChangeCipherSpec and Finished message.

Method	MTD_TLS_CLIENT_HELLO
Description	Sends or checks the Client Hello message used to initiate the Transport Layer Security (TLS) handshake in Server authentication or Mutual authentication mode on ES9+, ES11, ES12 or ES15.

Parameter(s)	• paramTLSversion: TLS protocol version • paramAlgs: cipher suite types supported • paramSessionID: Session ID • paramExts: Extensions data for “supported_signature_algorithms”, “trusted_ca_keys”, “server_name” (as defined in RFC 6066) or other
Details	Sends or receives a TLS ClientHello message according to the parameters defined above. In addition, the following parameters will be set: • The list of compression algorithms supported by the client is not explicitly defined, but by default it will be set to NULL. • The random of 4 bytes representing time since epoch on client host and 28 random bytes is not explicitly defined but it SHALL be generated by the test tool TLS implementation NOTE1: The Supported Elliptic Curves Extension and the Supported Point Formats Extension extensions MAY be sent by the Client. NOTE2: Only the extensions explicitly specified in paramExts while using the method SHALL be verified. Other extensions MAY be present.

Method	MTD_TLS_SERVER_HELLO_ETC
Description	Send or Receives to the Client Hello in the Transport Layer Security (TLS) handshake in Server authentication mode on ES9+, or ES11.
Parameter(s)	• paramTLSversion: TLS protocol version • paramAlgs: cipher suite selected • paramSessionID: Session ID • paramCertificate: TLS server certificate for authentication • paramOtherCertsInChain (optional): TLS other server certificate(s) in the chain for auth. •
Details	Sends or Receives a TLS ServerHello, Server Certificate, ServerKeyExchange and ServerHelloDone message in this order according to the parameters defined above. NOTE 1: The random of 4 bytes representing time since epoch on client host and 28 random bytes is not explicitly defined in the Server Hello message but it SHALL be generated by the Server under test. NOTE 2: Server_TLS_Ephemeral_Key, the value SHALL be set as defined in [24] for ServerKeyExchange. No verification required.

C.2 Procedures

	Procedure	PROC_ES11_AUTH_CLIENT	
	Description	Authenticate Server procedure and Event Retrieval from SM-SD.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	LPAd → S_SM-DS	Send ES11.AuthenticateClient method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO))

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

	Procedure	PROC_ES11_VERIFY_EVENT_RETRIEVAL_EVENT_ID	
	Description	Performs Common Mutual Authentication on ES11 from S_LPAd to SM-DS under test supplying <EVENT_ID_R> and verifies that the pending Event #EVENT_ENTRY_1 is retrieved.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → SM-DS	PROC_TLS_INITIALIZATION_SERVER_AUTH	
2	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS_ES11))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHING_ID_EVENT_ID_R))	MTD_HTTP_RESP(#R_AUTH_CLIENT_DS_EVENT_ENTRY_1_OK)

	Procedure	PROC_ES11_VERIFY_EVENT_RETRIEVAL_EVENT_ID_ERROR	
	Description	Performs Common Mutual Authentication on ES11 from S_LPAd to SM-DS under test supplying <EVENT_ID_R> and verifies that the pending Event #EVENT_ENTRY_1 is not available.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → SM-DS	PROC_TLS_INITIALIZATION_SERVER_AUTH	
2	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS_ES11))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHING_ID_EVENT_ID_R))	MTD_HTTP_RESP(#R_ERROR_8_9_5_3_9)

	Procedure	PROC_ES11_VERIFY_EVENT_RETRIEVAL_NO_EVENT_ID	
	Description	Performs Common Mutual Authentication on ES11 from S_LPAd to SM-DS under test supplying no MatchingId and verifies that the pending Event #EVENT_ENTRY_1 is retrieved.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → SM-DS	PROC_TLS_INITIALIZATION_SERVER_AUTH	
2	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS_ES11))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHING_ID_OMITTED))	MTD_HTTP_RESP(#R_AUTH_CLIENT_DS_EVENT_ENTRY_1_OK)

	Procedure	PROC_ES11_VERIFY_EVENT_RETRIEVAL_NO_EVENT_ID_ERROR	
	Description	Performs Common Mutual Authentication on ES11 from S_LPAd to SM-DS under test supplying no MatchingId and verifies that no events are available.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → SM-DS	PROC_TLS_INITIALIZATION_SERVER_AUTH	
2	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DS_ADDRESS_ES11))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DS	MTD_HTTP_REQ(#IUT_SM_DS_ADDRESS_ES11, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_MATCHING_ID_OMITTED))	MTD_HTTP_RESP(#R_AUTH_CLIENT_DS_EVENT_ENTRY_EMPTY_OK)

	Procedure	PROC_ES11_INIT_AUTH	
	Description	Initiate Authentication procedure with SM-DS.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	LPAd → S_SM-DS	Send ES11.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_ROOT_DS_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_ROOT_DS_ADDRESS, <LPA_RSP_CAPABILITY>))
2	S_SM-DS → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_DS_OK)	No error

	Procedure	PROC_EUICC_INITIALIZATION_SEQUENCE	
	Description	Initialize communication between the S_Device and the eUICC.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_Device → eUICC	RESET	ATR present
2	S_Device → eUICC	[SELECT_MF]	FCP Template present SW=0x9000
3	S_Device → eUICC	[TERMINAL_CAPABILITY_LPAd]	SW=0x9000
4	S_Device → eUICC	[TERMINAL_PROFILE]	Toolkit initialization THEN SW=0x9000

	Procedure	PROC_EUICC_INITIALIZATION_SEQUENCE_eUICCProfileStateChanged	
	Description	Initialize communication between the S_Device and the eUICC.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_Device → eUICC	RESET	ATR returned by eUICC
2	S_Device → eUICC	[SELECT_MF]	FCP Template present SW=0x9000
3	S_Device → eUICC	[TERMINAL_CAPABILITY_LPAd]	SW=0x9000
4	S_Device → eUICC	[TERMINAL_PROFILE_eUICCProfileStateChanged]	Toolkit initialization THEN SW=0x9000

	Procedure	PROC_OPEN_LOGICAL_CHANNEL_AND_SELECT_ISDR	
	Description	The LPAd opens a logical channel and selects the ISD-R.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → eUICC	[MANAGE_CHANNEL_OPEN]	Extract the <CHANNEL_NUMBER> from response data SW=0x9000
2	S_LPAd → eUICC	MTD_SELECT(#ISD_R_AID)	SW=0x9000

	Procedure	PROC_ES9+_AUTH_CLIENT	
	Description	Authenticate Server procedure without Confirmation Code. #R_AUTH_SERVER_MATCH_ID_DEV_INFO is used with the correct MatchingID defined by the Add Profile initiation procedure (Activation Code content or Empty MatchingID).	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK)	No error

	Procedure	PROC_ES9+_AUTH_CLIENT_CC	
	Description	Authenticate Server procedure (via Activation Code) with Confirmation Code. #R_AUTH_SERVER_MATCH_ID_DEV_INFO and #AUTH_SERVER_RESP_ACT_CODE_UC_OK are used with the correct MatchingID defined by the Add Profile initiation procedure (Activation Code content or Empty MatchingID).	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_OK_CC)	No error

	Procedure	PROC_ES9+_GET_BPP
	Description	Get BPP procedure without Confirmation Code.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_NO_CC))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error

	Procedure	PROC_ES9+_GET_BPP_CC
	Description	Get BPP procedure with Confirmation Code.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send ES9+.GetBoundProfilePackage method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #R_PREP_DOWNLOAD_WITH_CC))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error

	Procedure	PROC_ES9+_HANDLE_NOTIF
	Description	Handle Notification procedure.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_PIR_OK)) See NOTE 2
2	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error

NOTE 1: Other Notifications MAY be sent within the same HTTPS session.

NOTE 2: The values of notificationAddress, iccid and smdpOid used in #R_PIR_OK MAY vary depending on the context (ICCID of the downloaded profile, used SM-DP+ address and certificate).

	Procedure	PROC_ES9+_HANDLE_NOTIF_RPR_ERROR	
	Description	Handle Notification procedure for LoadRpmPackageResult with Errors.	
	Parameter(s)	paramRpmResultError : Parameter with rpmPackageResult containing rpmCommandResultDataError or rpmProcessingTerminated or loadRpmPackageErrorCode or loadRpmPackageErrorCodeNotSigned	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(paramRpmResultError))
2	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ES9+_HANDLE_NOTIF_RPM_OK	
	Description	Handle Notification procedure for LoadRpmPackageResult.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	LPAd → S_SM-DP+	Send ES9+.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#R_RPM_OK))
2	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ES9+_AUTH_CLIENT_FAIL_DEF_DP_USE_CASE_INVALID_MATCHING_ID	
	Description	AuthenticateClient fails due to an Invalid Matching ID.	
Step	**Direction**	**Sequence / Description**	**Expected result**
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(	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)

		#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_ACT_CODE_UC_OK))	MTD_HTTP_RESP(#R_ERROR_8_2_6_3_8)

	Procedure	PROC_ES9+_AUTH_CLIENT_RPM	
	Description	Authenticate Server procedure without Confirmation Code. #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3 and #AUTH_SERVER_RESP_RPM_OK are used with the correct MatchingID for the pending RPM package order.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP (#AUTH_CLIENT_RPM_OK)	No error

	Procedure	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CANCEL_SESSION_SK	
	Description	End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the Bound Profile Package.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	PROC_TLS_INITIALIZATION_SERVER_AUTH		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)

3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_SK)
5	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))	MTD_HTTP_RESP(#R_SUCCESS) Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode.

	Procedure	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CANCEL_SESSION_PPK	
	Description	End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys, profile protection keys and the generation of the Bound Profile Package.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	PROC_TLS_INITIALIZATION_SERVER_AUTH		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)
5	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS,	MTD_HTTP_RESP(#R_SUCCESS)

		#PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))	Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode.

	Procedure	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CC_CANCEL_SESSION_PPK
	Description	End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys, profile protection keys and the generation of the Bound Profile Package when a Confirmation Code is required.

Step	Direction	Sequence / Description	Expected result
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK_CC)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP_CC))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)
5	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))	MTD_HTTP_RESP(#R_SUCCESS) Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode.

	Procedure	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC_EN
	Description	Performs Common Mutual Authentication and then delivers the Bound Profile Package to the LPAd for enable metadata notifications.

Step	Direction	Sequence / Description	Expected result
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		

2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK_EN)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)

Procedure	PROC_ES9+_PROF_DOWNLOAD_ACT_CODE_USE_CASE_CANCEL_SESSION	
Description	End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the Bound Profile Package.	

Step	Direction	Sequence / Description	Expected result
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_ACT_CODE_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)
5	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))	MTD_HTTP_RESP(#R_SUCCESS) Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode.

	Procedure	PROC_ES9+_PROF_DOWNLOAD_SM_DS_USE_CASE_CANCEL_SESSION	
	Description	End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the bound profile package.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	PROC_ES9+_TLS_INITIALIZATION_SERVER_AUTH		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION (#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_SMDS_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)
5	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))	MTD_HTTP_RESP(#R_SUCCESS) Cancel Session request accepted by SM-DP+ and ongoing RSP session shall enter retry mode.

	Procedure	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC	
	Description	Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case without a confirmation code.	
Step	**Direction**	**Sequence / Description**	**Expected result**
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,	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK)

		#PATH_AUTH_CLIENT,MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	

	Procedure	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_CC
	Description	Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case with a confirmation code.

Step	Direction	Sequence / Description	Expected result
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK_CC)

	Procedure	PROC_ES9+_INIT_AUTH
	Description	Initiate Authentication procedure.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(<EUICC_CHALLENGE>, #R_EUICC_INFO1, #TEST_DP_ADDRESS1))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK)	No error

	Procedure	PROC_ES9+_INIT_AUTH_AND_AUTH_CLIENT_REQ_V3
	Description	Initiate Authentication procedure.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION (<EUICC_CHALLENGE>, #R_EUICC_INFO1, <LPA_RSP_CAPABILITY>))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	No error
3	LPAd → S_SM-DP+	Send ES9+.AuthenticateClient method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3)) Next step of common mutual authentication procedure is performed by the S_SM-DP+.

	Procedure	PROC_ES9+_INIT_AUTH_V3
	Description	Initiate Authentication procedure.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SM-DP+	Send ES9+.InitiateAuthentication method	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION (<EUICC_CHALLENGE>, #R_EUICC_INFO1, <LPA_RSP_CAPABILITY>))
2	S_SM-DP+ → LPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK_VARIANT_A)	No error, Next step of common mutual authentication procedure is performed.

	Procedure	PROC_ES9+_VERIFY_CMA_PD_DEF_SMDP_ADDRESS_NO_CC_FAIL	
	Description	Verifies that Common Mutual Authentication for the Profile Download Default SM_DP+ use case without a confirmation code fails due to the profile being in the 'Installed' or 'Error' state.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_ERROR_8_1_1_3_8)

	Procedure	PROC_VERIFY_SESSION_IS_CANCELLED	
	Description	Verify that the RSP session identified by the TransactionID <S_TRANSACTION_ID> has been cancelled by the eUICC (i.e. Common Mutual Authentication and Profile Download procedures SHALL be rejected as long as no GetEUICCChallenge has been requested).	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT(#PREP_DOWNLOAD_NO_CC)	#R_PREP_DOWN_NO_SESSION SW=0x9000 The transactionId returned in the response SHALL not be checked (any value SHALL be accepted)
2	S_LPAd → eUICC	MTD_STORE_DATA_SCRIPT(#AUTHENTICATE_SMDP)	#R_AUTH_SERVER_NO_SESSION SW = 0x9000 The transactionId returned in the response SHALL not be checked (any value SHALL be accepted)

	Procedure	PROC_ES9+_PROF_DOWNLOAD_DEF_DP_USE_CASE_CC_CANCEL_SESSION_SK
	Description	End User cancels ongoing Profile Download after the generation of the one-time ECKA key pair, session keys and the generation of the Bound Profile Package when a Confirmation Code is required.

Step	Direction	Sequence / Description	Expected result
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK_CC)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP_CC))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_SK)
5	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_CANCEL_SESSION, MTD_CANCEL_SESSION(<S_TRANSACTION_ID>, #CS_RESP_OK_POSTPONED))	MTD_HTTP_RESP(#R_SUCCESS) Cancel Session request accepted by SM-DP+ and ongoing RSP session SHALL enter retry mode.

	Procedure	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_CC_RETRY
	Description	Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case without a confirmation code.

Step	Direction	Sequence / Description	Expected result
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_HTTP_RESP(#R_AUTH_CLIENT_RETRY_OK)

		MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	

	Procedure	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_INVALID_CC	
	Description	Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case with an invalid confirmation code provided in the GetBoundProfilePackage.	
Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_CC		
1	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP_8_2_7_3_8))	MTD_HTTP_RESP(#R_ERROR_8_2_7_3_8)

	Procedure	PROC_ES9+_CMA_PD_DEF_SMDP_ADDRESS_UC_NO_CC_RETRY	
	Description	Performs Common Mutual Authentication for the Profile Download Default SM_DP+ use case without a confirmation code.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_RETRY_OK)

	Procedure	PROC_TLS_INITIALIZATION_SERVER_AUTH
	Description	Establishes the Transport Layer Security (TLS) v1.2 connection between the Client (S_)LPAd and (S_)SERVER using Server authentication mode on ES9+ or ES11 with Variant O certificate.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SERVER	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SERVER → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SERVER_TLS, NO_PARAM)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)
3	S_SERVER → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

	Procedure	PROC_TLS_INITIALIZATION_SERVER_AUTH_VARIANT_A
	Description	Establishes the Transport Layer Security (TLS) v3 connection between the Client LPAd and S_SERVER using Server authentication mode on ES9+ or ES11 as defined in SGP.22 v3 [2] for Variant A.

Step	Direction	Sequence / Description	Expected result
1	LPAd → S_SERVER	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, #SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
2	S_SERVER → LPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <SESSION_ID_RANDOM>, #CERT_S_SERVER_TLS, #CERT_S_SM_DP_SubCA_SIG)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)
3	S_SERVER → LPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established

	Procedure	PROC_ES9+_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC_NO_CC
	Description	Performs Common Mutual Authentication and then delivers the Bound Profile Package to the LPAd.

Step	Direction	Sequence / Description	Expected result
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH,	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)

		MTD_INITIATE_AUTHENTICATION(#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_OK)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)

	Procedure	PROC_ES9+_VERIFY_PROFILE_DOWNLOAD_DEF_SMDP_ADDRESS_UC	
	Description	Verifies that Common Mutual Authentication occurs successfully and that the Bound Profile Package is generated and successfully delivered to the LPAd.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_AUTH_CLIENT_OK) OR MTD_HTTP_RESP(#R_AUTH_CLIENT_RETRY_OK)
4	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_GET_BPP, MTD_GET_BPP(<S_TRANSACTION_ID>, #PREP_DOWNLOAD_RESP))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)

	Procedure	PROC_ES9+_VERIFY_PROFILE _NOT_RELEASED_EMPTY_MID	
	Description	Performs Common Mutual Authentication on ES9+ from S_LPAd to SM-DP+ under test supplying an empty MatchingId and verifies that there is no pending profile in Released state.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → SM-DP+	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+	
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT, MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_OK))	MTD_HTTP_RESP(#R_ERROR_8_2_1_2)

	Procedure	PROC_ES9+_VERIFY_PROFILE_RELEASED_EMPTY_MID_WITH_CC	
	Description	Performs Common Mutual Authentication on ES9+ from S_LPAd to SM-DP+ under test supplying an empty MatchingId and verifies that there is at least one pending profile in Released state.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_LPAd → SM-DP+	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+	
2	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_INITIATE_AUTH, MTD_INITIATE_AUTHENTICATION(#S_EUICC_CHALLENGE, #S_EUICC_INFO1, #IUT_SM_DP_ADDRESS))	MTD_HTTP_RESP(#R_INITIATE_AUTH_OK)
3	S_LPAd → SM-DP+	MTD_HTTP_REQ(#IUT_SM_DP_ADDRESS, #PATH_AUTH_CLIENT,	MTD_HTTP_RESP(# R_AUTH_CLIENT_OK_CC)

		MTD_AUTHENTICATE_CLIENT(<S_TRANSACTION_ID>, #AUTH_SERVER_RESP_DEF_DP_UC_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_CC))	MTD_HTTP_RESP(#R_GET_BPP_RESP_OP1_PPK)

Annex D Commands And Responses

D.1 ES8+ Requests And Responses

D.1.1 ES8+ Requests

Name	Content
CONF_ISDP_EMPTY	`req ConfigureISDPRequest ::={}`
CONF_ISDP_MAX_LENGTH	`req ConfigureISDPRequest ::={` `  dpProprietaryData { -- size=128 bytes` `  dpOid #S_SM_DP+_OID,` `additionalSmdpData` `#ADDITIONAL_SMDP_DATA_MAX_LENGTH` `  }` `}` `-- NOTE: Instead of` `DpProprietaryData ::= SEQUENCE {` `  dpOid OBJECT IDENTIFIER` `  -- additional data objects defined by the` `  -- SM-DP+ MAY follow` `}` `-- the following structure is used to test the` `-- DpProprietaryData size:` `DpProprietaryData ::= SEQUENCE {` `  dpOid OBJECT IDENTIFIER,` `  additionalSmdpData OCTET STRING OPTIONAL` `}`

CONF_ISDP_PROF1	req ConfigureISDPRequest ::={ dpProprietaryData { dpOid #S_SM_DP+_OID } }
CONF_ISDP_PROF1_SMDP	req ConfigureISDPRequest ::={ dpProprietaryData { dpOid #IUT_SM_DP_OID }-- optional }
CONF_ISDP_SIZE_EXCEEDED	req ConfigureISDPRequest ::={ dpProprietaryData { -- *size=129 bytes* dpOid #S_SM_DP+_OID, additionalSmdpData #ADDITIONAL_SMDP_DATA_EXCEEDED_MAX } } -- NOTE: Instead of DpProprietaryData ::= SEQUENCE { dpOid OBJECT IDENTIFIER -- additional data objects defined by the -- SM-DP+ MAY follow } -- the following structure is used to test the -- DpProprietaryData size: DpProprietaryData ::= SEQUENCE { dpOid OBJECT IDENTIFIER, additionalSmdpData OCTET STRING OPTIONAL }

FULL_METADATA	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 } }, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr1} }
INIT_SC_INVALID_CRT	req InitialiseSecureChannelRequest ::={ remoteOpId #REMOTE_OP_ID_INSTALL, transactionId <S_TRANSACTION_ID>, controlRefTemplate { keyType #INVALID_KEY_TYPE, keyLen #KEY_LENGTH, hostId #HOST_ID }, smdpOtpk <OTPK_S_SM_DP+_ECKA>, smdpSign <S_SM_DP+_SIGN> }
INIT_SC_INVALID_OP_ID	req InitialiseSecureChannelRequest ::={ remoteOpId #INVALID_REMOTE_OP_ID, transactionId <S_TRANSACTION_ID>, controlRefTemplate { keyType #KEY_TYPE, keyLen #KEY_LENGTH, hostId #HOST_ID }, smdpOtpk <OTPK_S_SM_DP+_ECKA>, smdpSign <S_SM_DP+_SIGN> }

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

METADATA_NO_CLASS	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 } } }
METADATA_OP_PROF1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 } }, profileOwner { mccMnc #MCC_MNC1 } }

METADATA_OP_PROF1_EN	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationLocalEnable }, notificationAddress #TEST_DP_ADDRESS1 } }, profileOwner { mccMnc #MCC_MNC1 } }
METADATA_OP_PROF1_ENT_RULES_RPM_CONF_EN	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID }, enterpriseConfiguration { enterpriseOid #S_ENTERPRISE_OID, enterpriseName #ENTERPRISE_NAME1, enterpriseRules { referenceEnterpriseRule, onlyEnterpriseProfilesCanBeEnabled, onlyEnterpriseProfilesCanBeInstalled } } }

METADATA_OP_PROF1_INST_DIFF	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS2 } }, profileOwner { mccMnc #MCC_MNC1 } }
METADATA_OP_PROF1_PPR_RPM_CONF_EN	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr1, ppr2}, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_RPM_CONF_ALL	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable, disable, delete, listProfileInfo, contactPcmp}, tagList '99'H -- *Tag for PPR* } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_ALL_CI_PKI_RAND	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType { enable, disable, delete, listProfileInfo, contactPcmp} tagList '99BB9ABC5A'H -- Tag for PPR, rpmConfiguration, hriServerAddress, lprConfiguration and ICCID } }, pollingAddress #TEST_DP_ADDRESS1, allowedCiPKId <CI_PKI_RANDOM>, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_RPM_CONF_ALL_DP_OID2	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr1}, hriServerAddress { #TEST_HRI_ADDRESS1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID2, rpmType {enable, disable, delete, listProfileInfo, contactPcmp} tagList '99BB9ABC5A'H -- Tag for PPR, rpmConfiguration, hriServerAddress, lprConfiguration and ICCID } }, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_ALL_PPR1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr1}, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable, disable, delete, listProfileInfo, contactPcmp}, tagList '99'H -- Tag for PPR } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_DE_PPR2	metadataReq StoreMetadataRequest:: = { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner {

	mccMnc #MCC_MNC1 }, profilePolicyRules {ppr2}, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {delete} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_DELETE	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {delete} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_DISABLE	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {disable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_RPM_CONF_EN	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_EN_CI_PKI_RPM	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable}, } }, pollingAddress #TEST_DP_ADDRESS1, allowedCiPKId #PK_CI_ECDSA_RPM, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_RPM_CONF_EN_CI_PKI_RAND	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable}, } }, pollingAddress #TEST_DP_ADDRESS1, allowedCiPKId #PK_CI_ECDSA_RANDOM, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_EN_DP_OID2	metadataReq StoreMetadataRequest:: = { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID2, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_EN_NO_POLLING_ADDRESS	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} }

	}, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_EN_NOTIF_CONF	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationRpmEnable, notificationRpmDisable, notificationRpmDelete }, notificationAddress #TEST_DP_ADDRESS1 } }, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_RPM_CONF_EN_NOTIF_CONF2	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationRpmEnable, notificationRpmDisable, notificationRpmDelete }, notificationAddress #TEST_DP_ADDRESS2 } }, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_EN_POL_DP8	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable}, } }, pollingAddress #TEST_DP_ADDRESS8, allowedCiPKId #PK_CI_ECDSA_RANDOM, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_RPM_CONF_EN_POL_SMDS_CI_PKI_RPM	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable}, } }, pollingAddress #TEST_ROOT_DS_ADDRESS, allowedCiPKId #PK_CI_ECDSA_RPM, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF1_RPM_CONF_EN_POL_SMDS_CI_PKI_RAND	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable}, } }, pollingAddress #TEST_ROOT_DS_ADDRESS, allowedCiPKId #PK_CI_ECDSA_RANDOM, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_RPM_CONF_UPDATE_MD_PPR	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr1, ppr2}, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, tagList '99'H -- *Tag for PPR* } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } } }
METADATA_OP_PROF1_RPM_UM_ENT_RULES	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, tagList 'BD'H -- *Tag for Enterprise Configuration* } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID }, enterpriseConfiguration { enterpriseOid #S_ENTERPRISE_OID, enterpriseName #ENTERPRISE_NAME1, enterpriseRules { onlyEnterpriseProfilesCanBeEnabled, onlyEnterpriseProfilesCanBeInstalled } } }

METADATA_OP_PROF2_MEMRES1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS2 } }, profileOwner { mccMnc #MCC_MNC2 }, profilePolicyRules { ppr2 } }
METADATA_OP_PROF4_MEMRES1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF4, serviceProviderName #SP_NAME4, profileName #NAME_OP_PROF4, iconType png, icon #ICON_OP_PROF4, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS4 } }, profileOwner { mccMnc #MCC_MNC4 }, profilePolicyRules { ppr1 } }

METADATA_OP_PROF2	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS2 } }, profileOwner { mccMnc #MCC_MNC2 } }
METADATA_OP_PROF2_NO_INSTALL	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS2 } }, profileOwner { mccMnc #MCC_MNC2 } }

METADATA_OP_PROF2_RPM_CONF_ALL_OWNER2	metadataReq StoreMetadataRequest::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, profileOwner { mccMnc #MCC_MNC2 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable, disable, delete, listProfileInfo, contactPcmp}, tagList '99'H -- *Tag for PPR* } }, pollingAddress #TEST_DP_ADDRESS2, profileOwnerOid #S_PROFILE_OWNER_OID2 } }
METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, profileOwner { mccMnc #MCC_MNC2 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1_AND_NOTIF_CONF	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationLocalDisable, notificationRpmEnable, notificationRpmDisable, notificationRpmDelete }, notificationAddress #TEST_DP_ADDRESS2 } }, profileOwner { mccMnc #MCC_MNC2 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS2, profileOwnerOid #S_PROFILE_OWNER_OID } }
METADATA_OP_PROF2_RPM_CONF_EN_OWNER_OID1_PPR1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, profileOwner { mccMnc #MCC_MNC2 }, profilePolicyRules {ppr1}, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable} } }, pollingAddress #TEST_DP_ADDRESS1, profileOwnerOid #S_PROFILE_OWNER_OID } }

METADATA_OP_PROF1_NO_INSTALL	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 } }, profileOwner { mccMnc #MCC_MNC1 } }
METADATA_OP_PROF2_TEST_DP_ADDRESS1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF2, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF2, iconType png, icon #ICON_OP_PROF2, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 } }, profileOwner { mccMnc #MCC_MNC2 } }
METADATA_OP_PROF3	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF3, serviceProviderName #SP_NAME3, profileName #NAME_OP_PROF3, iconType png, icon #ICON_OP_PROF3, profileClass operational, profileOwner { mccMnc #MCC_MNC2 }, profilePolicyRules { ppr2 } }

METADATA_OP_PROF3_RPM_CONF_ALL	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF3, serviceProviderName #SP_NAME3, profileName #NAME_OP_PROF3, iconType png, icon #ICON_OP_PROF3, profileClass operational, profileOwner { mccMnc #MCC_MNC3 }, hriServerAddress { #TEST_HRI_ADDRESS3 }, rpmConfiguration { managingDpList { { managingDP #S_SM_DP+_OID, rpmType {enable, disable, delete, listProfileInfo, contactPcmp} tagList '99BB9ABC5A'H -- Tag for PPR, rpmConfiguration, hriServerAddress, lprConfiguration and ICCID } }, pollingAddress #TEST_DP_ADDRESS3, profileOwnerOid #S_PROFILE_OWNER_OID }, lprConfiguration { pcmpAddress #TEST_PCMP_ADDRESS3 } }
METADATA_OP_PROF4	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF4, serviceProviderName #SP_NAME4, profileName #NAME_OP_PROF4, iconType png, icon #ICON_OP_PROF4, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS4 } }, profileOwner { mccMnc #MCC_MNC4 }, profilePolicyRules { ppr1 } }

METADATA_OP_PROF5	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF5, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF5, iconType png, icon #ICON_OP_PROF5, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 } }, profileOwner { mccMnc #MCC_MNC1 } }
METADATA_OP_PROF6	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF6, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF6, iconType png, icon #ICON_OP_PROF6, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS2 } }, profileOwner { mccMnc #MCC_MNC2 } }

METADATA_OP_PROF7	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF7, serviceProviderName #SP_NAME2, profileName #NAME_OP_PROF7, iconType png, icon #ICON_OP_PROF7, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS8 } }, profileOwner { mccMnc #MCC_MNC2 }, profilePolicyRules { ppr2 } }
METADATA_OP_PROF8	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF8, serviceProviderName #SP_NAME8, profileName #NAME_OP_PROF8, iconType png, icon #ICON_OP_PROF8, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS8 } }, profileOwner { mccMnc #MCC_MNC2 }, profilePolicyRules { ppr2 } }

METADATA_OP_PROF9	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF9, serviceProviderName #SP_NAME9, profileName #NAME_OP_PROF9, profileOwner { mccMnc #MCC_MNC9, gid1 #GID1, gid2 #GID2 }, profilePolicyRules { ppr2 } }
METADATA_OP1_GID1GID2_PRESENT	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileOwner { mccMnc #MCC_MNC1, gid1 #GID1, gid2 #GID2 }, profilePolicyRules {ppr2} }
METADATA_OP9_GID1GID2_MISSING	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF9, serviceProviderName #SP_NAME9, profileName #NAME_OP_PROF9, profileOwner { mccMnc #MCC_MNC9 } }
METADATA_PPR_NO_OWNER	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profilePolicyRules {ppr2} }
METADATA_WILDCARD	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileOwner { mccMnc #MCC_MNC_WILDCARD }, profilePolicyRules {ppr2} }

METADATA_WITH_JPG	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType jpg, icon #ICON_JPG }

METADATA_WITH_NOTIFS	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, notificationConfigurationInfo { { profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS3 }, { profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS2 }, { profileManagementOperation { notificationLocalEnable }, notificationAddress #TEST_DP_ADDRESS2 }, { profileManagementOperation { notificationLocalEnable }, notificationAddress #TEST_DP_ADDRESS3 }, { profileManagementOperation { notificationLocalDisable }, notificationAddress #TEST_DP_ADDRESS3 }, { profileManagementOperation { notificationLocalDisable }, notificationAddress #TEST_DP_ADDRESS4 }, { profileManagementOperation { notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 }, { profileManagementOperation { notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS3 } } }

METADATA_WITH_PPR1_PPR2	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr1,ppr2} }
METADATA_WITH_PPR2	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr2} }
METADATA_WITH_PPRS_AND_ICON	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules {ppr1,ppr2} }
METADATA_WITHOUT_ICON	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType jpg }
REPLACE_S_KEYS_REQ	req ReplaceSessionKeysRequest ::={ initialMacChainingValue <PPK_INIT_MAC>, ppkEnc <PPK_ENC>, ppkCmac <PPK_MAC> }
REPLACE_S_KEYS_REQ_INV_SIZE	req ReplaceSessionKeysRequest ::={ initialMacChainingValue #PPK_INIT_MAC_INV_SIZE, ppkEnc #PPK_ENC_INV_SIZE, ppkCmac #PPK_MAC_INV_SIZE }

S_INIT_SC_PROF1	req InitialiseSecureChannelRequest ::={ remoteOpId #REMOTE_OP_ID_INSTALL, transactionId <S_TRANSACTION_ID>, controlRefTemplate { keyType #KEY_TYPE, keyLen #KEY_LENGTH, hostId #HOST_ID }, smdpOtpk <OTPK_S_SM_DP+_ECKA>, smdpSign <S_SM_DP+_SIGN> }
SMDP_METADATA_ABS	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1 }
SMDP_METADATA_ALL	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #IUT_SM_DP_ADDRESS } }, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules { ppr1, ppr2 } }
SMDP_METADATA_NON_ASCII	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME_NON_ASCII, profileName #NAME_OP_PROF1_NON_ASCII }
SMDP_METADATA_NOTIF_MULTI	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, notificationConfigurationInfo { { profileManagementOperation { notificationInstall,

	notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #IUT_SM_DP_ADDRESS }, { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1 } } }
SMDP_METADATA_OP_PROF1_EN	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationLocalEnable }, notificationAddress #IUT_SM_DP_ADDRESS } } }
SMDP_METADATA_OP_PROF1_PPR2	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileClass operational, profileOwner { mccMnc #MCC_MNC1 }, profilePolicyRules { ppr2 } }
SMDP_METADATA_PN_LONG	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF_LONG }
SMDP_METADATA_SPN_LONG	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME_LONG, profileName #NAME_OP_PROF1 }

D.2 ES9+ Requests And Responses

D.2.1 ES9+ Requests

Name	Content
AUTH_SERVER_RESP_ACT_CODE_UC_OK_EID2	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_ACT_CODE }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_EID2, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_ACT_CODE_2_UC_OK	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_ACT_CODE_2 }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }

AUTH_SERVER_RESP_DEF_DP_OK_eUICC_EXT	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2_UICC_EXT ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_OK DEVICE_EXT	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2_DEV_EXT ctxParams1 #CTX_PARAMS1_DEVICE_INFO_EXT }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_1_3_8	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_EID2, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_4_8	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 {

	transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2_INSUF_MEM_ERROR, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_BC_cA	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_BC_cA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_BC_PLC	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_BC_PLC }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_CP	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress

	#IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_CP }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_EX_KU	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_KU }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_1_SIG	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_SIG }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_2_6_3	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge

	<SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_EXPIRED }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_EX_CP	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_EX_CP, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_EX_KU	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_EX_KU, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_SIG	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge

	<SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_SIG, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_SUB_ORG	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_SUB_ORG, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_1_SUB_SN	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_SUB_SN, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_3_6_3	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge

	<SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_EXPIRED, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_6_1_CHA	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE_INVALID>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_1_6_1_SIG	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1_INVALID>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_2_5_4_3	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2_PPR2, ctxParams1

	#CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_10_1_3_9	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <INVALID_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_DEF_DP_UC_8_11_1_3_9	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_UNKNOWN }

AUTH_SERVER_RESP_DEF_DP_UC_OK	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2 ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDP_MATCHING_ID_EMPTY	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDP_MATCHING_ID_OMITTED	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2 ctxParams1 #CTX_PARAMS1_MATCHING_ID_OMITTED }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_UC_OK	resp AuthenticateServerResponse ::= authenticateResponseOk : {

	euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_SMDS }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_UC_OK_EID2	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DP_ADDRESS, serverChallenge <SMDP_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_SMDS }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_EID2, eumCertificate #CERT_EUM_ECDSA }
CS_RESP_ERROR_8_1_6_1	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason postponed }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE_INVALID> }
CS_RESP_ERROR_8_8_3_10	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid <INVALID_SM_DP_OID>, reason postponed }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_ERROR_8_10_1_3_9	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned {

	transactionId <INVALID_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason postponed }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_OK_EU_REJ	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason endUserRejection }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_OK_L_BPP_EXE_ERROR	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason loadBppExecutionError }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_OK_M_DATA_MISMATCH	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason metadataMismatch }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_OK_POSTPONED	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason postponed }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_OK_PPR_NOT_ALLOWED	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID,

	reason pprNotAllowed }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_OK_TIMEOUT	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason timeout }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_RESP_OK_UNDEFINED	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #IUT_SM_DP_OID, reason undefinedReason }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CTX_PARAMS1_ACT_CODE	ctx CtxParams1 ::= ctxParamsForCommonAuthentication : { matchingId #MATCHING_ID_1, deviceInfo #S_DEVICE_INFO }
CTX_PARAMS1_ACT_CODE_2	ctx CtxParams1 ::= ctxParamsForCommonAuthentication : { matchingId #MATCHING_ID_2, deviceInfo #S_DEVICE_INFO }
CTX_PARAMS1_MATCHING_ID_EMPTY	ctx CtxParams1 ::= ctxParamsForCommonAuthentication : { matchingId #MATCHING_ID_EMPTY, deviceInfo #S_DEVICE_INFO }
CTX_PARAMS1_SMDS	ctx CtxParams1 ::= ctxParamsForCommonAuthentication : { matchingId <MATCHING_ID_EVENT>, deviceInfo #S_DEVICE_INFO }
EUICC_FIRMWARE_VER	0x01 00 00

EXT_CARD_RESOURCE_LIMITED_SPACE	The Extended Card Resource Information according to ETSI TS 102 226 and set as: 0x81 <L> #INSTALLED_PROFILES 0x82 <L> #NON_VOLATILE_MEM_LIMITED_SPACE 0x83 <L> #S_VOLATILE_MEM
INITIATE_AUTH_DS_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDS_SIGNED1>, "serverSignature1" : <S_SMDS_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DSauth_SIG, "crlList" : #CRL_LIST }

INITIATE_AUTH_DS_OK_1	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDS_SIGNED_ADDR1>, "serverSignature1" : <S_SMDS_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DSauth_ SIG, "crlList" : #CRL_LIST }
INITIATE_AUTH_DS_OK_VARIANT_A	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDS_SIGNED1>, "serverSignature1" : <S_SMDS_SIGNATURE1>, "serverCertificate" : #CERT_S_SM_DSauth_SIG, "otherCertsInChain" : #CERT_S_SM_DS_SubCA_SIG, "crlList" : #CRL_LIST }
INITIATE_AUTH_INV_CERT_DS	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDS_SIGNED1>, "serverSignature1" : <S_SMDS_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, -- NOTE: select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING> "serverCertificate" : #CERT_S_SM_DSauth_INV_SIGN, "crlList" : #CRL_LIST }
INITIATE_AUTH_INV_CI_DS	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success"

	} }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDS_SIGNED1>, "serverSignature1" : <S_SMDS_SIGNATURE1>, "euiccCiPKIdTobeUsed" : #CI_PK_ID_INV, "serverCertificate" : #CERT_S_SM_DSauth_SIG, "crlList" : #CRL_LIST } NOTE: select and choose the #CERT_S_SM_DSauth_SIG leading to the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING>
INITIATE_AUTH_INV_SIGN_DS	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDS_SIGNED1>, "serverSignature1" : <S_SMDS_SIGNATURE_INV>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DSauth_SIG, "crlList" : #CRL_LIST } -- NOTE: select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING> and choose the #CERT_S_SM_DSauth_SIG leading to the same Root CI certificate
INITIATE_AUTH_INV_SMDS_ADDRESS	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDS_SIGNED_INV_ADDR>, "serverSignature1" : <S_SMDS_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DSauth_SIG, "crlList" : #CRL_LIST } -- NOTE: select the CI Key ID in highest priority from the

	<EUICC_CI_PK_ID_LIST_FOR_SIGNING> and choose the #CERT_S_SM_DSauth_SIG leading to the same Root CI certificate
INITIATE_AUTH_INV_CERT	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>,-- NOTE: select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING> "serverCertificate" : #CERT_S_SM_DPauth_INV_SIGN, "crlList" : #CRL_LIST }
INITIATE_AUTH_INV_CI	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "euiccCiPKIdTobeUsed" : #CI_PKI_ID2, "serverCertificate" : #CERT_S_SM_DPauth, "crlList" : #CRL_LIST } NOTE: select and choose the #CERT_S_SM_DPauth_SIG leading to the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING>

INITIATE_AUTH_INV_OID	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DP2auth_SIG, "crlList" : #CRL_LIST } -- NOTE: select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING> -- NOTE: serverSignature1 SHALL be calculated correctly, using the secret key related to CERT_S_SM_DP2auth_SIG.
INITIATE_AUTH_INV_SIGN	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE_INV>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DPauth_SIG, "crlList" : #CRL_LIST } -- NOTE: select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING> and choose the #CERT_S_SM_DPauth_SIG leading to the same Root CI certificate

INITIATE_AUTH_INV_SMDP+_ADDRESS	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED_INV_ADDR>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DPauth_SIG, "crlList" : #CRL_LIST } -- NOTE: select the CI Key ID in highest priority from the <EUICC_CI_PK_ID_LIST_FOR_SIGNING> and choose the #CERT_S_SM_DPauth_SIG leading to the same Root CI certificate -- NOTE: serverSignature1 SHALL be calculated correctly, using <S_SMDP_SIGNED_INV_ADDR>.
INITIATE_AUTH_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "euiccCiPKIdTobeUsed" : <EUICC_CI_PK_ID_TO_BE_USED>, "serverCertificate" : #CERT_S_SM_DPauth_SIG, "crlList" : #CRL_LIST }
INITIATE_AUTH_OK_DIFF_CERT_VARIANT_A	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>,

	"serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "serverCertificate" : #CERT_S_SM_DPauth_PK_CI2_SIG, "otherCertsInChain" : #CERT_S_SM_DP_SubCA_SIG, "crlList" : #CRL_LIST }
INITIATE_AUTH_OK_VARIANT_A	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "serverCertificate" : #CERT_S_SM_DPauth_SIG, "otherCertsInChain" : #CERT_S_SM_DP_SubCA_SIG, "crlList" : #CRL_LIST }
INITIATE_AUTH_OK_VARIANT_B	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "serverCertificate" : #CERT_S_SM_DPauth_SIG, “otherCertsInChain” : #CERT_CI_SubCA_SIG, “crlList” : #CRL_LIST }
INITIATE_AUTH_OK_VARIANT_C	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>,

	"serverSigned1" : <S_SMDP_SIGNED1>, "serverSignature1" : <S_SMDP_SIGNATURE1>, "serverCertificate" : #CERT_S_SM_DPauth_SIG, "otherCertsInChain" : { #CERT_S_SM_DP_SubCAList_SIG, #CERT_CI_SubCA_SIG}, "crlList" : #CRL_LIST }
INITIATE_AUTH_OK_VARIANT_FLEX	Depending on the certificate chain supported by the Test eUICC, either #INITIATE_AUTH_OK_VARIANT_A or #INITIATE_AUTH_OK_VARIANT_B or #INITIATE_AUTH_OK_VARIANT_C
MATCHING_ID_EMPTY	
NON_VOLATILE_MEM_LIMITED_SPACE	'0x00 01'
PENDING_NOTIF_DEL1	response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_DEL2	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF2 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }

PENDING_NOTIF_DEL4	response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS4, iccid #ICCID_OP_PROF4 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_DEL5	response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF5 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_DEL6	response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDelete }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF6 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }

PENDING_NOTIF_DIS1	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDisable }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_DIS2	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDisable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF2 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_DIS5	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDisable }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF5 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }

PENDING_NOTIF_DIS8	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDisable }, notificationAddress #TEST_DP_ADDRESS8, iccid #ICCID_OP_PROF8 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_EN1	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalEnable }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_EN2	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalEnable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF2 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }

PENDING_NOTIF_EN5	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalEnable }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF5 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_EN6	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalEnable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF6 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_IMPLICIT_DIS2	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDisable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF2 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain <EUICC_NEXT_CERT>, otherCertsInChain <EUICC_OTHER_CERTS> }

PENDING_NOTIF_RPM_EN1	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationRpmEnable }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain <EUICC_NEXT_CERT>, otherCertsInChain <EUICC_OTHER_CERTS> }
PENDING_NOTIF_RPM_EN2	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationRpmEnable }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain <EUICC_NEXT_CERT>, otherCertsInChain <EUICC_OTHER_CERTS> }
PP_VERSION	0x01 00 00

S_PN_PIR_OK1	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult successResult : { aid <ISD_P_AID>, simaResponse #SIMA_RESULT_OK } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_INVALID_TRANS_ID	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <INVALID_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult successResult : { aid <ISD_P_AID>, simaResponse #SIMA_RESULT_OK } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_INCORRECT_INPUT_VALUES	response PendingNotification ::= profileInstallationResult : profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId configureISDP, errorReason incorrectInputValues } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_INVALID_SIGN	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId initialiseSecureChannel, errorReason invalidSignature } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_UNSUPPORTED_CRT	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId initialiseSecureChannel, errorReason unsupportedCrtValues } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_UNSUP_REMOTE_OP_TYPE	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId initialiseSecureChannel, errorReason unsupportedRemoteOperationType } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_UNSUP_PROFILE_CLASS	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId storeMetadata, errorReason unsupportedProfileClass } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_SCP03T_STRUCTURE_ERROR	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId storeMetadata, errorReason scp03tStructureError } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_SCP03T_SECURITY_ERROR	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId replaceSessionKeys, errorReason scp03tSecurityError } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_ICCID_ALREADY_EXISTS	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId storeMetadata, errorReason installFailedDueToIccidAlreadyExistsOnEuic c } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_INSUFFICIENT_MEMORY	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId storeMetadata, errorReason installFailedDueToInsufficientMemoryForPro file } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_INSTALL_INTERRUPTION	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId storeMetadata, errorReason installFailedDueToInterruption } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_PE_PROCESSING_ERROR	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId loadProfileElements, errorReason installFailedDueToPEProcessingError } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_DATA_MISMATCH	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId loadProfileElements, errorReason installFailedDueToDataMismatch } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_TEST_PROFILE_INVALID_NAA_KEY	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId loadProfileElements, errorReason testProfileInstallFailedDueToInvalidNaaKey } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PN_PIR_PPR_NOT_ALLOWED	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId storeMetadata, errorReason pprNotAllowed } }, euiccSignPIR <EUICC_SIGN_PIR> }

S_PN_PIR_UNKNOWN_ERROR	response PendingNotification ::= profileInstallationResult : { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, smdpOid #IUT_SM_DP_OID, finalResult errorResult : { bppCommandId storeMetadata, errorReason installFailedDueToUnknownError } }, euiccSignPIR <EUICC_SIGN_PIR> }
S_PENDING_NOTIF_OTHER_INST1	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
S_PENDING_NOTIF_EN1	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalEnable }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }

S_PENDING_NOTIF_DIS1	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDisable }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
S_PENDING_NOTIF_DE1	response PendingNotification ::= otherSignedNotification :{ tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationLocalDelete }, notificationAddress #IUT_SM_DP_ADDRESS, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
S_SMDP_SIGNED2	req SmdpSigned2 ::= { transactionId <S_TRANSACTION_ID>, ccRequiredFlag FALSE }
S_SMDP_SIGNED2_CC	req SmdpSigned2 ::= { transactionId <S_TRANSACTION_ID>, ccRequiredFlag TRUE }
S_SMDP_SIGNED2_INV_TRANSACTION_ID	req SmdpSigned2 ::= { transactionId <INVALID_TRANSACTION_ID>, ccRequiredFlag FALSE }
S_SMDP_SIGNED2_RPM_PENDING	req SmdpSigned2 ::= { transactionId <S_TRANSACTION_ID>, ccRequiredFlag FALSE, rpmPending NULL }

S_SMDP_SIGNED3	req SmdpSigned3 ::= { transactionId <S_TRANSACTION_ID>, rpmPackage #RPM_PKG_EN, rpmPending NULL }
S_SMDP_SIGNED3_DELETE	req SmdpSigned3 ::= { transactionId <S_TRANSACTION_ID>, rpmPackage #RPM_PKG_DELETE }
S_SMDP_SIGNED3_DISABLE	req SmdpSigned3 ::= { transactionId <S_TRANSACTION_ID>, rpmPackage #RPM_PKG_DISABLE }
S_SMDP_SIGNED3_RPM_PENDING	req SmdpSigned3 ::= { transactionId <S_TRANSACTION_ID>, rpmPackage #RPM_PKG_EN, rpmPending NULL }
S_SMDP_SIGNED3_UM_PPR	req SmdpSigned3 ::= { transactionId <S_TRANSACTION_ID>, rpmPackage #RPM_PKG_UM_PPR }
S_SMDP_SIGNED3_UM_REF_ENT_RULE	req SmdpSigned3 ::= { transactionId <S_TRANSACTION_ID>, rpmPackage #RPM_PKG_UM_REF_ENT_RULE }
S_VOLATILE_MEM	'0x01 00'
SAS_ACREDITATION_NUMBER	GSMA_SAS_123456789
UICC_CAPABILITY	uiccCapability UICCCapability ::= { contactlessSupport, usimSupport, isimSupport, akaMilenage, akaTuak128, gbaAuthenUsim, eapClient, javacard, multipleUsimSupport }
UICC_CAPABILITY_EXT	uiccCapability UICCCapability ::= { contactlessSupport, usimSupport, isimSupport, akaMilenage, akaTuak128, gbaAuthenUsim, eapClient, javacard, multipleUsimSupport, berTlvFileSupport, dfLinkSupport, catTp, getIdentity, profile-a-x25519, profile-b-

	p256, suciCalculatorApi, unknownServiceSupport } Note: the definition of UICCCapability used above is equivalent to the definition in SGP.22 v2.3 (specific version of [2]) with the additional of a further field called "unknownServiceSupport" after the "suciCalculatorApi" field.

D.2.2 ES9+ Responses

Name	Content
AUTH_CLIENT_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : <S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DPpb_SIG }
AUTH_CLIENT_OK_CC	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2_CC, "smdpSignature2" : <S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DPpb_SIG }
AUTH_CLIENT_OK_OLD_KEYS	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2_OLD_KEYS, "smdpSignature2" :

	<S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DPpb_SIG }
AUTH_CLIENT_INV_PB_CERT	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : <S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DPpb_INV_SIGN }
AUTH_CLIENT_INV_CI	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : <S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DP2pb_SIG }
AUTH_CLIENT_INV_SIGN	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : <S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DPpb_SIG } The <S_SM_DP+_SIGNATURE2> SHALL NOT be computed using the #SK_S_SM_DPpb_SIG *but SHALL have the same length as for a valid signature*

AUTH_CLIENT_INV_TRANSACTION_ID	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2_INV_TRANSACTION_ID, "smdpSignature2" : <S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DPpb_SIG }
AUTH_CLIENT_RPM_DELETE_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "smdpSigned3" : #S_SMDP_SIGNED3_DELETE, "smdpSignature3" : <S_SM_DP+_SIGNATURE3> }
AUTH_CLIENT_RPM_DISABLE_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "smdpSigned3" : #S_SMDP_SIGNED3_DISABLE, "smdpSignature3" : <S_SM_DP+_SIGNATURE3> }
AUTH_CLIENT_RPM_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "smdpSigned3" : #S_SMDP_SIGNED3, "smdpSignature3" : <S_SM_DP+_SIGNATURE3> }

AUTH_CLIENT_RPM_OK_AND_RPM_PENDING	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "smdpSigned3" : #S_SMDP_SIGNED3_RPM_PENDING, "smdpSignature3" : <S_SM_DP+_SIGNATURE3> }
AUTH_CLIENT_RPM_UM_PPR_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "smdpSigned3" : #S_SMDP_SIGNED3_UM_PPR, "smdpSignature3" : <S_SM_DP+_SIGNATURE3> }
AUTH_CLIENT_RPM_UM_REF_ENT_RULE_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "smdpSigned3" : #S_SMDP_SIGNED3_UM_REF_ENT_RULE, "smdpSignature3" : <S_SM_DP+_SIGNATURE3> }
CS_OK_EU_LOAD_BPP_ERROR	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #S_SM_DP+_OID, reason loadBppExecutionError }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_OK_EU_POSTPONED	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #S_SM_DP+_OID,

	reason postponed }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_OK_EU_REJ	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #S_SM_DP+_OID, reason endUserRejection }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_OK_PPR_NOT_ALLOWED	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #S_SM_DP+_OID, reason pprNotAllowed }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_OK_RPM_DISABLED	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #S_SM_DP+_OID, reason rpmDisabled }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_OK_RPM_EU_REJECT	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #S_SM_DP+_OID, reason endUserRejection }, euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
CS_OK_TIMEOUT	resp CancelSessionResponse ::= cancelSessionResponseOk : { euiccCancelSessionSigned { transactionId <S_TRANSACTION_ID>, smdpOid #S_SM_DP+_OID, reason timeout },

	euiccCancelSessionSignature <EUICC_CANCEL_SESSION_SIGNATURE> }
GET_BPP_LOAD_ERROR	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "boundProfilePackage" : BoundProfilePackage { #S_INIT_SC_PROF1, firstSequenceOf87 { #CONF_ISDP_PROF1 }, sequenceOf88 { <METADATA_OP_PROF1_SEG> … <METADATA_OP_PROF1_SEG> } } } NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1. NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.
GET_BPP_LOAD_ERROR_UNKNOWN_TAG	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } },

	"transactionId" : <S_TRANSACTION_ID>, "boundProfilePackage" { #S_INIT_SC_PROF1, #UNKNOWN_BPP_SEGMENT, firstSequenceOf87 { #CONF_ISDP_PROF1 }, sequenceOf88 { <METADATA_OP_PROF1_SEG> … <METADATA_OP_PROF1_SEG> }, sequenceOf86 { <PPP_OP_PROF1_SEG_SK> … <PPP_OP_PROF1_SEG_SK> } } } NOTE 1: boundProfilePackage is encoded as base64 therefore the test tool shall decode boundProfilePackage to access the ASN.1. NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.
GET_BPP_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "boundProfilePackage" : BoundProfilePackage { #S_INIT_SC_PROF1, firstSequenceOf87 { #CONF_ISDP_PROF1 }, sequenceOf88 { <METADATA_OP_PROF1_SEG> … <METADATA_OP_PROF1_SEG> }, sequenceOf86 { <PPP_OP_PROF1_SEG_SK> … <PPP_OP_PROF1_SEG_SK> } } } NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1. NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.

GET_BPP_OK_PPK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "boundProfilePackage" : BoundProfilePackage { #S_INIT_SC_PROF1, firstSequenceOf87 { 0x87 <L> #CONF_ISDP_PROF1 }, sequenceOf88 { <METADATA_OP_PROF1_SEG> … <METADATA_OP_PROF1_SEG> }, secondSequenceOf87 { 0x87 <L> #REPLACE_S_KEYS_REQ }, sequenceOf86 { <PPP_OP_PROF1_SEG_SK> … <PPP_OP_PROF1_SEG_SK> } } }
GET_BPP_INV	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "boundProfilePackage" : BoundProfilePackage { #S_INIT_SC_PROF1, firstSequenceOf87 { 0x87 <L> #CONF_ISDP_PROF1 }, sequenceOf88 { <METADATA_OP_PROF1_SEG> … <METADATA_OP_PROF1_SEG> }, sequenceOf86 { <PPP_OP_PROF1_SEG_SK_INV> … <PPP_OP_PROF1_SEG_SK_INV> } } }

NOTIF_METADATA_PROF1_DP1_RPR (NotificationMetadata)	{ seqNumber <NOTIF_SEQ_NO_PROF1_RPR>, profileManagementOperation { loadRpmPackageResult }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }
PENDING_NOTIF_INST_ADDRESS2	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS2, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
PENDING_NOTIF_INST1	response PendingNotification ::= otherSignedNotification : { tbsOtherNotification { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }, euiccNotificationSignature <TBS_EUICC_NOTIF_SIG>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }

R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO	resp AuthenticateServerResponse ::authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #TEST_ROOT_DS_ADDRESS, serverChallenge <S_SMDS_CHALLENGE>, euiccInfo2 <R_EUICC_INFO2>, ctxParams1 #CTX_PARAMS1_MATCH_ID_DEV_INFO }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
R_AUTH_SERVER_DS_MATCH_ID_DEV_INFO_1	resp AuthenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #TEST_DS_ADDRESS1, serverChallenge <S_SMDS_CHALLENGE>, euiccInfo2 <R_EUICC_INFO2>, ctxParams1 #CTX_PARAMS1_MATCH_ID_DEV_INFO }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }
R_AUTH_SERVER_MATCH_ID_DEV_INFO	resp AuthenticateServerResponse ::= authenticateResponseOk { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #TEST_DP_ADDRESS1, serverChallenge <S_SMDP_CHALLENGE>, euiccInfo2 <R_EUICC_INFO2>, ctxParams1 #CTX_PARAMS1_MATCH_ID_DEV_INFO }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain #CERT_EUM_SIG }

R_AUTH_SERVER_MATCH_ID_DEV_INFO_V3	resp AuthenticateServerResponse ::= authenticateResponseOk { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #TEST_DP_ADDRESS1, serverChallenge <S_SMDP_CHALLENGE>, euiccInfo2 <R_EUICC_INFO2>, ctxParams1 #CTX_PARAMS1_MATCH_ID_DEV_INFO_V3 }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_SIG, nextCertInChain <EUICC_NEXT_CERT>, otherCertsInChain <EUICC_OTHER_CERTS> }
R_GET_BPP_RESP_OP1_PPK (Pre-generated PPP for Profiles)	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId": <TRANSACTION_ID_GBPP>, "boundProfilePackage" : BoundProfilePackage { #INIT_SC_PROF1, firstSequenceOf87 { <CONF_ISDP_PROF1_ENC> }, sequenceOf88 { <SMDP_METADATA_SEG_MAC> … <SMDP_METADATA_SEG_MAC> }, secondSequenceOf87 { <REPLACE_S_KEYS_REQ_ENC> }, sequenceOf86 { <PPP_OP_PROF1_SEG_PPK> … <PPP_OP_PROF1_SEG_PPK> } } } NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1. NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.

R_GET_BPP_RESP_OP1_SK (Dynamically-generated PPP for Profiles)	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId": <TRANSACTION_ID_GBPP>, "boundProfilePackage" : BoundProfilePackage { #INIT_SC_PROF1, firstSequenceOf87 { <CONF_ISDP_PROF1_ENC> }, sequenceOf88 { <SMDP_METADATA_SEG_MAC> … <SMDP_METADATA_SEG_MAC> }, sequenceOf86 { <PPP_OP_PROF1_SEG_SK> … <PPP_OP_PROF1_SEG_SK> } } } NOTE 1: boundProfilePackage is enconded as base64 therefore the test tool SHALL decode boundProfilePackage to access the ASN.1. NOTE 2: For sequenceOf88 there will be only one or two '88' TLV segments depending on the size of StoreMetadata.
R_HTTP_204_OK	HTTP/1.1 204 No Content X-Admin-Protocol: gsma/rsp/v<2.1.0> NOTE: If the HTTP response is being received from the server under test, then the "Content-type" header MAY be present.
R_PIR_OK	response ProfileInstallationResult ::= { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, notificationMetadata { seqNumber <SEQ_NUMBER>, profileManagementOperation { notificationInstall }, notificationAddress #TEST_DP_ADDRESS1, iccid #ICCID_OP_PROF1 }, smdpOid #S_SM_DP+_OID, finalResult successResult : { aid <ISD_P_AID>, ppiResponse #SIMA_RESULT_OK }

	}, euiccSign <EUICC_SIGN_PIR> }
R_PIR_SECU_INVALID	resp ProfileInstallationResult ::= { profileInstallationResultData { transactionId <S_TRANSACTION_ID>, … smdpOid #S_SM_DP+_OID, finalResult errorResult : { bppCommandId loadProfileElements, errorReason incorrectInputValues OR bspStructureError OR bspSecurityError … } }, euiccSign <EUICC_SIGN_PIR> }
SMDP_METADATA_OP_PROF1	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, profileClass operational }
SMDP_METADATA_OP_PROF1_2_SEG	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF1, serviceProviderName #SP_NAME1, profileName #NAME_OP_PROF1, iconType png, icon #ICON_OP_PROF1_2_SEG, profileClass operational, notificationConfigurationInfo { { profileManagementOperation { notificationInstall, notificationLocalEnable, notificationLocalDisable, notificationLocalDelete }, notificationAddress #IUT_SM_DP_ADDRESS } }, profileOwner { mccMnc #MCC_MNC1 } }

SMDP_METADATA_OP_PROF3	metadataReq StoreMetadataRequest ::= { iccid #ICCID_OP_PROF3, serviceProviderName #SP_NAME3, profileName #NAME_OP_PROF3, profileClass operational, profileOwner { mccMnc #MCC_MNC2 }, profilePolicyRules { ppr2 } }
SMDP_SIGNED2	smdpSigned2 SmdpSigned2 ::= { transactionId <TRANSACTION_ID_SIGNED_AC>, ccRequiredFlag FALSE }
SMDP_SIGNED2_CC	smdpSigned2 SmdpSigned2 ::= { transactionId <TRANSACTION_ID_SIGNED_AC>, ccRequiredFlag TRUE }
SMDP_SIGNED2_CC_RETRY	smdpSigned2 SmdpSigned2 ::= { transactionId <TRANSACTION_ID_SIGNED_AC>, ccRequiredFlag TRUE, bppEuiccOtpk <BPP_OTPK_EUICC_ECKA> }
SMDP_SIGNED2_RETRY	smdpSigned2 SmdpSigned2 ::= { transactionId <TRANSACTION_ID_SIGNED_AC>, ccRequiredFlag FALSE, bppEuiccOtpk <BPP_OTPK_EUICC_ECKA> }

D.3 VOID

D.4 VOID

D.5 VOID

D.6 ES11 Requests And Responses

D.6.1 ES11 Requests

Name	Content
AUTH_SERVER_RESP_MATCHING_ID_EMPTY	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11,

	serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_MATCHING_ID_EVENT_ID	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EVENT_ID }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_MATCHING_ID_EVENT_ID_R	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EVENT_ID_R }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_MATCHING_ID_OMITTED	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress

	#IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_OMITTED }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_BC_cA	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_BC_cA }
AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_BC_PLC	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_BC_PLC }
AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_CP	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 {

	transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_CP }
AUTH_SERVER_RESP_SMDS_8_1_2_6_1_EX_KU	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_INVALID_EX_KU }
AUTH_SERVER_RESP_SMDS_8_1_2_6_1_SIG	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate

	#CERT_EUM_ECDSA_INVALID_SIG }
AUTH_SERVER_RESP_SMDS_8_1_2_6_3	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA_EXPIRED }
AUTH_SERVER_RESP_SMDS_8_1_3_6_1_EX_CP	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_EX_CP, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_3_6_1_EX_KU	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>,

	euiccCertificate #CERT_EUICC_ECDSA_INVALID_EX_KU, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_3_6_1_SIG	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_SIG, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_3_6_1_SUB_ORG	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_SUB_ORG, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_3_6_1_SUB_SN	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1

	<EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_INVALID_SUB_SN, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_3_6_3	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA_EXPIRED, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_6_1_CHA	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE_INVALID>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_1_6_1_SIG	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <S_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY },

	euiccSignature1 <EUICC_SIGNATURE1_INVALID>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
AUTH_SERVER_RESP_SMDS_8_10_1_3_9	resp authenticateServerResponse ::= authenticateResponseOk : { euiccSigned1 { transactionId <INVALID_TRANSACTION_ID>, serverAddress #IUT_SM_DS_ADDRESS_ES11, serverChallenge <SMDS_CHALLENGE>, euiccInfo2 #S_EUICC_INFO2, ctxParams1 #CTX_PARAMS1_MATCHING_ID_EMPTY }, euiccSignature1 <EUICC_SIGNATURE1>, euiccCertificate #CERT_EUICC_ECDSA, eumCertificate #CERT_EUM_ECDSA }
CTX_PARAMS1_MATCHING_ID_EVENT_ID (CtxParams1)	ctxParamsForCommonAuthentication : { matchingId #EVENT_ID_1, deviceInfo #S_DEVICE_INFO }
CTX_PARAMS1_MATCHING_ID_EVENT_ID_R (CtxParams1)	ctxParamsForCommonAuthentication : { matchingId <EVENT_ID_R>, deviceInfo #S_DEVICE_INFO }
CTX_PARAMS1_MATCHING_ID_OMITTED (CtxParams1)	ctxParamsForCommonAuthentication : { deviceInfo #S_DEVICE_INFO }

D.6.2 ES11 Responses

Name	Content
AUTH_CLIENT_DS_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : #EVENT_ENTRY }

AUTH_CLIENT_DS_OK1	{ "header" :{ "functionExecutionStatus":{ "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : [#EVENT_ENTRY_1] }
AUTH_CLIENT_DS_OK2	{ "header" :{ "functionExecutionStatus":{ "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : [#EVENT_ENTRY_2] }
AUTH_CLIENT_DS_OK_DSADDR1	{ "header" :{ "functionExecutionStatus":{ "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : [#EVENT_ENTRY_DSADDR1] }
EVENT_ENTRY	{ "eventId" : <EVENT_ID>, "rspServerAddress" : <RSP_SERVER_ADDRESS> }
EVENT_ENTRY_1	{ "eventId" : #EVENT_ID_1, "rspServerAddress" : #TEST_DP_ADDRESS1 }
EVENT_ENTRY_1_ALT_DS	{ "eventId" : #EVENT_ID_1, "rspServerAddress" : #TEST_ALT_DS_ADDRESS }
EVENT_ENTRY_2	{ "eventId" : #EVENT_ID_2, "rspServerAddress" : #TEST_DP_ADDRESS1 }
EVENT_ENTRY_DSADDR1	{ "eventId" : #EVENT_ID_1, "rspServerAddress" : #TEST_DS_ADDRESS1 }
EVENT_ENTRY_MULTI	{ "eventId" : #EVENT_ID_1, "rspServerAddress" : #TEST_DP_ADDRESS1 }, { "eventId" : #EVENT_ID_2, "rspServerAddress" : #TEST_DP_ADDRESS2 }

R_AUTH_CLIENT_DS_EVENT_ENTRY_1_ALT_DS_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : [#EVENT_ENTRY_1_ALT_DS] }
R_AUTH_CLIENT_DS_EVENT_ENTRY_1_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : [#EVENT_ENTRY_1] }
R_AUTH_CLIENT_DS_EVENT_ENTRY_EMPTY_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : [] }
R_AUTH_CLIENT_DS_EVENT_ENTRY_MULTI_OK	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } }, "transactionId" : <S_TRANSACTION_ID>, "eventEntries" : [#EVENT_ENTRY_MULTI] }

D.7 VOID

D.8 VOID

D.9 Common Server Responses

For all responses with a JSON component the “subjectIdentifier” and “message” are optional and may or may not be present in the response received from the RSP server.

Name	Content
R_ERROR_1_2_4_2	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "1.2", "reasonCode" : "4.2" } } } }
R_ERROR_8_1_1_2_2	{ "header" : {

	"functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.1.1", "reasonCode" : "2.2" } } } }
R_ERROR_8_1_1_3_8	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.1.1", "reasonCode" : "3.8" } } } }
R_ERROR_8_1_1_3_10	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.1.1", "reasonCode" : "3.10" } } } }
R_ERROR_8_1_3_6_1	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.1.3", "reasonCode" : "6.1" } } } }
R_ERROR_8_1_3_6_3	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.1.3", "reasonCode" : "6.3" } } } }

R_ERROR_8_1_4_8	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.1", "reasonCode" : "4.8" } } } }
R_ERROR_8_1_6_1	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.1", "reasonCode" : "6.1" } } } }
R_ERROR_8_2_1_2	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.2", "reasonCode" : "1.2" } } } }
R_ERROR_8_2_1_3_3	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.2.1", "reasonCode" : "3.3" } } } }
R_ERROR_8_2_1_3_9	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.2.1", "reasonCode" : "3.9" } }

	} }
R_ERROR_8_2_3_7	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.2", "reasonCode" : "3.7" } } } }
R_ERROR_8_2_5_4_3	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.2.5", "reasonCode" : "4.3" } } } }
R_ERROR_8_2_6_3_3	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.2.6", "reasonCode" : "3.3" } } } }
R_ERROR_8_2_6_3_8	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.2.6", "reasonCode" : "3.8" } } } }
R_ERROR_8_2_6_3_10	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.2.6", "reasonCode" : "3.10"

	} } } }
R_ERROR_8_2_7_2_2	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.2.7", "reasonCode" : "2.2" } } } }
R_ERROR_8_2_7_3_8	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.2.7", "reasonCode" : "3.8" } } } }
R_ERROR_8_2_7_6_4	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.2.7", "reasonCode" : "6.4" } } } }
R_ERROR_8_8_1_3_8	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.8.1", "reasonCode" : "3.8" } } } }
R_ERROR_8_8_2_3_1	{ "header" : { "functionExecutionStatus" : { "status" : "Failed",

	"statusCodeData" : { "subjectCode" : "8.8.2", "reasonCode" : "3.1" } } } }
R_ERROR_8_8_3_3_1	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.8.3", "reasonCode" : "3.1" } } } }
R_ERROR_8_8_3_10	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.8", "reasonCode" : "3.10" } } } }
R_ERROR_8_8_4_3_7	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.8.4", "reasonCode" : "3.7" } } } }
R_ERROR_8_8_5_4_10	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.8.5", "reasonCode" : "4.10" } } } }
R_ERROR_8_8_5_6_4	{ "header" : {

	"functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.8.5", "reasonCode" : "6.4" } } } }
R_ERROR_8_9_1_3_8	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.9.1", "reasonCode" : "3.8" } } } }
R_ERROR_8_9_2_3_1	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.9.2", "reasonCode" : "3.1" } } } }
R_ERROR_8_9_3_3_1	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.9.3", "reasonCode" : "3.1" } } } }
R_ERROR_8_9_4_2	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.9", "reasonCode" : "4.2" } } } }

R_ERROR_8_9_4_3_7	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.9.4", "reasonCode" : "3.7" } } } }
R_ERROR_8_9_5_1	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.9", "reasonCode" : "5.1" } } } }
R_ERROR_8_9_5_3_3	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.9.5", "reasonCode" : "3.3" } } } }
R_ERROR_8_9_5_3_9	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.9.5", "reasonCode" : "3.9" } } } }
R_ERROR_8_10_1_3_9	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData” : { "subjectCode" : "8.10.1", "reasonCode" : "3.9" } }

	} }
R_ERROR_8_11_1_3_9	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : "8.11.1", "reasonCode" : "3.9" } } } }
R_ERROR_ANY	{ "header" : { "functionExecutionStatus" : { "status" : "Failed", "statusCodeData" : { "subjectCode" : <SUBJECT_CODE_ANY>, "reasonCode" : <REASON_CODE_ANY> } } } }
R_SUCCESS	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } } }

D.10 ES2+ Requests And Responses

D.10.1 ES2+ Requests

D.10.2 ES2+ Responses

Name	Content
R_SUCCESS_ICCID1	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } } "iccid" : "#ICCID_OP_PROF1" }
R_SUCCESS_MATCHING_ID	{ "header" : {

	"functionExecutionStatus" : { "status" : "Executed-Success" } } "matchingId" : <MATCHING_ID> }
R_SUCCESS_MATCHING_ID_EID	{ "header" : { "functionExecutionStatus" : { "status" : "Executed-Success" } } "matchingId" : <MATCHING_ID> "eid" : "#EID1" }

Annex E Profiles

Profile	GENERIC_PROFILE_STRUCTURE
Description	Generic Operational Profile ASN.1 structure to be used as a basis for all Profiles used in this specification.
Details	headerValue ProfileElement ::= header : { major-version 2, minor-version 3, profileType "GSMA Profile Package", iccid '89019990001234567893'H, eUICC-Mandatory-services { usim NULL, milenage NULL }, eUICC-Mandatory-GFSTEList { -- see Note 1 id-MF, id-USIM } } mfValue ProfileElement ::= mf : { mf-header { mandated NULL, identification 1 }, templateID id-MF, mf { fileDescriptor : { pinStatusTemplateDO '01020A'H } }, ef-pl { fileDescriptor : { -- EF PL modified to use Access Rule 15 within EF ARR securityAttributesReferenced '0F'H } }, ef-iccid { -- swapped ICCID: 98109909002143658739 fillFileContent '98109909002143658739'H }, ef-dir { fileDescriptor {

```
      -- Shareable Linear Fixed File
      -- 4 records, record length: 38 bytes
      fileDescriptor '42210026'H,
      efFileSize '98'H
    },
    -- USIM AID: A0000000871002FF33FF018900000100
    fillFileContent
      '61184F10A0000000871002FF33FF0189000001005004553494D'H
  },
  ef-arr {
    fileDescriptor : {
      fileDescriptor '42210025'H,
      lcsi '05'H,
      efFileSize '022B'H
    },
    fillFileContent : '800101900080010 2A40683010195010880015 8A40683010A950108'H,
    fillFileOffset : 10,
    fillFileContent : '800101A40683010195010880015AA40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent : '80015BA40683010A950108'H,
    fillFileOffset : 26,
    fillFileContent : '800101900080015A9700'H,
    fillFileOffset : 27,
    fillFileContent : '800103A40683010195010880015 8A40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent : '800111A40683010195010880014AA40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent :
'800103A40683010195010880015 8A40683010A950108840132A406830101950108'H,
    fillFileOffset : 4,
    fillFileContent :
'800101A40683010195010880010 2A40683018195010880015 8A40683010A950108'H,
    fillFileOffset : 4,
    fillFileContent : '800101900080011AA40683010195010880014 0A40683010A950108'H,
    fillFileOffset : 10,
    fillFileContent : '800101900080015AA40683010A950108'H,
    fillFileOffset : 21,
    fillFileContent : '800101900080011 8A40683010A9501088001429700'H,
    fillFileOffset : 16,
    fillFileContent : '800101A40683010195010880015A9700'H,
    fillFileOffset : 21,
    fillFileContent : '800113A40683010195010880014 8A40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent : '80015EA40683010A950108'H,
    fillFileOffset : 26,
    fillFileContent  '800101900080010 2A010A40683010195
```

```
                0108A406830102950108800158A40683
                010A950108'H
  }
}

pukVal ProfileElement ::= pukCodes : {
  puk-Header {
    mandated NULL,
    identification 2
  },
  pukCodes {
    {
      keyReference pukAppl1,
      pukValue '30303030303030'H,
      -- maxNumOfAttemps:9, retryNumLeft:9
      maxNumOfAttemps-retryNumLeft 153
    },
    {
      keyReference pukAppl2,
      pukValue '3132333435363738'H
    },
    {
      keyReference secondPUKAppl1,
      pukValue '3932393435363738'H,
      -- maxNumOfAttemps:8, retryNumLeft:8
      maxNumOfAttemps-retryNumLeft 136
    }
  }
}
pinVal ProfileElement ::= pinCodes : {
  pin-Header {
    mandated NULL,
    identification 3
  },
  pinCodes pinconfig : {
    {
      keyReference pinAppl1,
      pinValue '31323334FFFFFFFF'H,
      unblockingPINReference pukAppl1
    },
    {
      keyReference pinAppl2,
      pinValue '30303030FFFFFFFF'H,
      unblockingPINReference pukAppl2
    },
```

```
    {
      keyReference adm1,
      pinValue '35363738FFFFFFFF'H,
      pinAttributes 1
    }
  }
}
usimValue ProfileElement ::= usim : {
  usim-header {
    mandated NULL,
    identification 4
  },
  templateID id-USIM,
  adf-usim {
    fileDescriptor : {
      fileID '7FF1'H,
      dfName 'A0000000871002FF33FF018900000100'H,
      pinStatusTemplateDO '01810A'H
    }
  },
  ef-imsi {
    -- numerical format: 234101943787656
    fillFileContent '08294301913487 6765'H
  },
  ef-arr {
    fileDescriptor {
      linkPath '2F06'H
    }
  },
  ef-ust {
    -- Service Dialling Numbers, Short Message Storage…
    fillFileContent '0A2E178CE73204000000000000'H
  },
  ef-spn {
    -- ASCII format: "GSMA eUICC"
    fillFileContent '0247534D41206555494343FFFFFFFFFFFF'H
  },
  ef-est {
    -- Services deactivated
    fillFileContent '00'H
  },
  ef-acc {
    -- Access class 4
    fillFileContent '0040'H
  },
```

```
  ef-ecc {
    -- Emergency Call Code 911
    fillFileContent '19F1FF01'H
  }
}

usimPin ProfileElement ::= pinCodes : {
  pin-Header {
    mandated NULL,
    identification 5
  },
  pinCodes pinconfig : {
    {
      keyReference secondPINAppl1,
      pinValue '39323338FFFFFFFF'H
      unblockingPINReference secondPUKAppl1,
      -- PIN is Enabled
      pinAttributes 1,
      -- maxNumOfAttemps:2, retryNumLeft:2
      maxNumOfAttemps-retryNumLeft 34
    }
  }
}

akaParamValue ProfileElement ::= akaParameter : {
  aka-header {
    mandated NULL,
    identification 6
  },
  algoConfiguration algoParameter : {
    algorithmID milenage,
    -- RES and MAC 64 bits, CK and IK 128 bits
    algorithmOptions '01'H,
    key '000102030405060708090A0B0C0D0E0F'H,
    opc '0102030405060708090A0B0C0D0E0F00'H,
    -- rotationConstants uses default: '4000204060'H
    -- xoringConstants uses default value
    authCounterMax '010203'H
  }
  -- sqnOptions uses default: '02'H
  -- sqnDelta uses default: '000010000000'H
  -- sqnAgeLimit uses default: '000010000000'H
  -- sqnInit uses default: all bytes zero
}
```

```
mnoSdValue ProfileElement ::= securityDomain : {
 sd-Header {
  mandated NULL,
  identification 7
 },
 instance {
  applicationLoadPackageAID 'A0000001515350'H,
  classAID 'A000000151535041'H,
  instanceAID 'A000000151000000'H,
  applicationPrivileges '82FC80'H,
  -- Secured
  lifeCycleState '0F'H,
  -- SCP80 supported
  applicationSpecificParametersC9 '810280008201F08701F0'H,
  -- other parameters MAY be necessary
  applicationParameters {
   -- TAR: B20100, MSL: 12
   uiccToolkitApplicationSpecificParametersField
     '01000001000000002011203B2010000'H
  }
 },
 keyList {
  {
   -- C-ENC + R-ENC
   keyUsageQualifier '38'H,
   -- ENC key
   keyIdentifier '01'H,
   keyVersionNumber '01'H,
   keyCompontents {
    {
     -- DES mode implicitly known (as an example)
     keyType '80'H,
     -- This value MAY be freely changed
     keyData '112233445566778899AABBCCDDEEFF10'H
    }
   }
  },
  {
   -- C-MAC + R-MAC
   keyUsageQualifier '34'H,
   -- MAC key
   keyIdentifier '02'H,
   keyVersionNumber '01'H,
   keyCompontents {
    {
```

```
        -- DES mode implicitly known (as an example)
        keyType '80'H,
        -- This value MAY be freely changed
        keyData '112233445566778899AABBCCDDEEFF10'H
      }
    }
  },
  {
    -- C-DEK + R-DEK
    keyUsageQualifier 'C8'H,
    -- data ENC key
    keyIdentifier '03'H,
    keyVersionNumber '01'H,
    keyCompontents {
      {
        -- DES mode implicitly known (as an example)
        keyType '80'H,
        -- This value MAY be freely changed
        keyData '112233445566778899AABBCCDDEEFF10'H
      }
    }
  },
    -- AES Token Key (as an example)
    -- This value MAY be freely changed
          keyUsageQualifier  '81'H,
          -- MAY be used by SD
          keyAccess  '01'H,
          -- Key Id 01
          keyIdentifier  '01'H,
          keyVersionNumber '70'H,
          keyCompontents  {
           {
      -- AES (16 bytes key length)
      -- This value MAY be freely changed
            keyType  '88'H,
            -- This value MAY be freely changed
            keyData  'CDFE56B7B72FAE6A047341F003D7A48D'H
           }
    }
  },
  {
    -- Receipt (the AES scheme SHALL be supported)
          keyUsageQualifier '44'H,
          -- MAY be used by SD
          keyAccess '01'H,
```

```
            -- Key Id 01
            keyIdentifier '01'H,
            keyVersionNumber '71'H,
            keyCompontents  {
             {
              -- AES (16 bytes key length)
              keyType  '88'H,
              -- This value MAY be freely changed
              keyData  '11121314212223243132333441424344'H
             }
            }
   }
  }
}

ssdValue ProfileElement ::= securityDomain : {
  sd-Header {
    mandated NULL,
    identification 8
  },
  instance {
    applicationLoadPackageAID 'A0000001515350'H,
    classAID 'A000000151535041'H,
    instanceAID 'A00000055910100102736456616C7565'H,
    -- by default extradited under MNO-SD
    -- Privileges: Security Domain + Trusted Path
    applicationPrivileges '808000'H,
    -- Personalized
    lifeCycleState '0F'H,
    -- SCP80 supported, extradiction supported
    applicationSpecificParametersC9 '810280008201F0'H,
    applicationParameters {
      -- TAR: 6C7565, MSL: 12
      uiccToolkitApplicationSpecificParametersField
        '0100000100000020112036C756500'H
    }
  },
  keyList {
    {
      -- C-ENC + R-ENC
      keyUsageQualifier '38'H,
      keyIdentifier '01'H,
      keyVersionNumber '01'H,
      keyCompontents {
        {
```

```
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value MAY be freely changed
          keyData '11223344556677881122334455667788'H
        }
      }
    },
    {
      -- C-MAC + R-MAC
      keyUsageQualifier '34'H,
      -- MAC key
      keyIdentifier '02'H,
      keyVersionNumber '01'H,
      keyCompontents {
        {
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value MAY be freely changed
          keyData '11223344556677881122334455667788'H
        }
      }
    },
    {
      -- C-DEK + R-DEK
      keyUsageQualifier 'C8'H,
      -- data ENC key
      keyIdentifier '03'H,
      keyVersionNumber '01'H,
      keyCompontents {
        {
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value MAY be freely changed
          keyData '11223344556677881122334455667788'H
        }
      }
    }
  }
}

rfmUicc ProfileElement ::= rfm : {
  rfm-header {
    identification 11
  },
  -- Instance AID
```

```
  instanceAID ' A00000055910100001'H,
  tarList {
    'B00000'H
  },
  -- cryptographic checksum + counter higher
  minimumSecurityLevel '12'H,
  -- full access
  uiccAccessDomain '00'H,
  -- full access
  uiccAdminAccessDomain '00'H
}

rfmUsim ProfileElement ::= rfm : {
  rfm-header {
    identification 12
  },
  -- Instance AID
  instanceAID 'A00000055910100002'H,
  tarList {
    'B00020'H
  },
  -- cryptographic checksum + counter higher
  minimumSecurityLevel '12'H,
  -- full access
  uiccAccessDomain '00'H,
  -- full access
  uiccAdminAccessDomain '00'H,
  adfRFMAccess {
    adfAID 'A0000000871002FF33FF018900000100'H,
    -- UICC access condition: ADM1
    adfAccessDomain '02000100'H,
    -- UICC access condition: ADM1
    adfAdminAccessDomain '02000100'H
  }
}

endValue ProfileElement ::= end : {
  end-header {
    mandated NULL,
    identification 99
  }
}
```

Note 1: The following OIDs are used:

```
   id-MF OBJECT IDENTIFIER ::=
   {joint-iso-itu-t(2)  international-organizations(23)  simalliance(143)  euicc-
profile(1) template(2) mf(1)}
```

```
   id-USIM OBJECT IDENTIFIER ::=
   {joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) usim(4)}
```

Profile	PROFILE_OPERATIONAL1
Description	Operational Profile This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF1, except if defined differently in the test sequence. The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF1 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF1 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF1 • the pinAttributes of pinAppl1 present in the PE_PIN SHALL be set to 6 • the SCP80 encryption key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO_SCP80_ENC_KEY • the SCP80 message authentication key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO_SCP80_AUTH_KEY • the SCP80 data encryption key configured in the PE-SecurityDomain that corresponds to the MNO-SD SHALL be set to #MNO_SCP80_DATA_ENC_KEY • the instance AID configured in the PE-SecurityDomain that corresponds to the Supplementary Security Domain PE_SSD SHALL be set to #SSD_AID • the ef-dir present in the PE-MF SHALL be configured with the AID #USIM_AID • the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD • the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H The PROFILE_OPERATIONAL1 UPP is named #UPP_OP_PROF1 in the scope of this document.

Profile	PROFILE_OPERATIONAL2
Description	Operational Profile This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF2, except if defined differently in the test sequence. The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF2 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF2 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF2

	• The pinAttributes of pinAppl1 present in the PE_PIN SHALL be set to 6 • the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD • the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H The PROFILE_OPERATIONAL2 UPP is named #UPP_OP_PROF2 in the scope of this document.

Profile	PROFILE_OPERATIONAL3
Description	Operational Profile with PPR2 but without notification This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF3, except if defined differently in the test sequence. The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF3 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF3 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF3 • the pinAttributes of pinAppl1 present in the PE_PIN SHALL be set to 6 • the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD • the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H The PROFILE_OPERATIONAL3 UPP is named #UPP_OP_PROF3 in the scope of this document.

Profile	PROFILE_OPERATIONAL4
Description	Operational Profile with PPR1 and notification This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF4, except if defined differently in the test sequence. The Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE] except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF4 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF4 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF4 • the pinAttributes of pinAppl1 present in the PE_PIN SHALL be set to 6

	• the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD • the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H The PROFILE_OPERATIONAL4 UPP is named #UPP_OP_PROF4 in the scope of this document.

Profile	PROFILE_OPERATIONAL5
Description	Operational Profile with pinAppl1 enabled. This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF5, except if defined differently in the test sequence. The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF5 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF5 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF5 • the pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO1_PIN1 • the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD • the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H

Profile	PROFILE_OPERATIONAL6
Description	Operational Profile with pinAppl1 enabled. This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF6, except if defined differently in the test sequence. The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF6 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF6 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF6 • The pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO2_PIN1 • the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD

	• the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H

Profile	PROFILE_OPERATIONAL7
Description	Operational Profile with PPR2 and notification This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF7, except if defined differently in the test sequence. The Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF7 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF7 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF7 • the pinAttributes of pinAppl1 present in the PE_PIN SHALL be set to 6 • the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD • the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H

Profile	PROFILE_OPERATIONAL8
Description	Operational Profile with PPR2, pinAppl1 enabled and notification This Profile acts as an Operational Profile in the scope of this specification. NOTE: Milenage algorithm is used in this Profile
Details	The Profile Metadata SHALL be set to #METADATA_OP_PROF8, except if defined differently in the test sequence. The Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that: • the *iccid* field SHALL be set to #ICCID_OP_PROF8 in the *ProfileHeader* element, in non-swapped format • the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF8 • the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF8 • The pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO2_PIN1 • the ef-ust SHALL be set in accordance to #EF_UST1 (service 17 and 18 are not available) • the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H • the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD • the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H

Profile	PROFILE_OPERATIONAL9
Description	Operational Profile with GID1 and GID2 set

<table>
<tr><td></td><td>This Profile acts as an Operational Profile in the scope of this specification.
NOTE: Milenage algorithm is used in this Profile</td></tr>
<tr><td>Details</td><td>
The Profile Metadata SHALL be set to #METADATA_OP_PROF9, except if defined differently in the test sequence.
The Unprotected Profile Package content SHALL follow the ASN.1 structure specified above for GENERIC_PROFILE_STRUCTURE except that:

the iccid field SHALL be set to #ICCID_OP_PROF9 in the ProfileHeader element, in non-swapped format
the ef-iccid present in the PE-MF SHALL be set to #ICCID_OP_PROF9
the ef-imsi present in the PE-USIM SHALL be set to #IMSI_OP_PROF9
the pinAppl1 present in the PE_PIN SHALL be enabled and has the value #PO1_PIN1
the ef-ust SHALL be set to #EF_UST2 (service 17 and 18 are available)
the applicationPrivileges in PE-MNO-SD SHALL be set to '82DC00'H
the Token Verification and the Receipt Generation keys SHALL not be set in the PE-MNO-SD
the applicationSpecificParametersC9 in PE-MNO-SD SHALL be set to '810280008201F08701F0'H
the following new Profile Element PE_OPT_USIM SHALL be inserted right after PE_USIM:

PE_OPT_USIM
<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>
NOTE : The following OIDs are used:
<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>
The PROFILE_OPERATIONAL9 UPP is named #UPP_OP_PROF9 in the scope of this document.
</td></tr>
</table>

Annex F IUT Settings

F.1 VOID

F.2 VOID

F.3 Device Settings

Device Setting name	Description
IUT_CAT_CLASSES	The bit string to indicate the set of supported Card Application Toolkit letter classes as defined in TS 102 223.
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_EUICC_FORM_FACTOR	Indicates whether the eUICC is removable (0) or non-removable (1)
IUT_EUTRAN_5GC_REL	If EUTRAN 5GC is supported, this SHALL be the highest 3GPP release N supported by the Device for E-UTRAN 5GC core access and encoded as the octet string {N, 0, 0}.
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_RSP_VERSION_HIGHEST	The highest version of SGP.22 supported by the LPA encoded as the value part of an ASN.1 VersionType (e.g. 0x03 01 00)
IUT_LPAd_Confirmation	Description of the way to perform Strong Confirmation.
IUT_LPAd_CI	CI subjectPublicKeyInfo of CERT.CI.SIG (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.

Device Setting name	Description
IUT_LTE_EUTRAN_REL	If LTE/E-UTRAN is supported, this SHALL be the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}.
IUT_NFC_REL	If NFC is supported, this SHALL be the highest (version, revision) number of TS.26 [15], encoded as the octet string {version, revision, 0}.
IUT_NR_5GC_REL	If NR 5GC is supported, this SHALL be the highest 3GPP release N supported by the Device for NR 5GC core access and encoded as the octet string {N, 0, 0}.
IUT_NR_EPC_REL	If NR EPC is supported, this SHALL be the highest 3GPP release N supported by the Device for NR EPC core access and encoded as the octet string {N, 0, 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 VOID

Annex G Initial States

Unless it is defined differently in a particular test case, the IUTs SHALL be set in the following initial state before the test case execution.

G.1 Device

G.1.1 Device (default)

The Device is “powered on”.

The Device is in the normal execution mode after Device boot-up and Device initial configuration. The Device is NOT in the Test Mode.

The LPAd has access to the root CI key #CERT_CI_ECDSA (or the CI public key) for verification of the TLS certificates of SM-DP+ or SM-DS. No CRL is loaded.

The Device contains a Test eUICC pre-configured as defined below in G.1.3.

G.1.2 Companion Device connected to a Primary Device

The Companion Device is connected to the Primary Device as defined by the Device vendor.

Companion Device and the connected Primary Device are “powered on”.

The Companion Device and Primary Device are in the normal execution mode (NOT in the boot-up mode).

The LPAd of the Companion Device has access to the root CI #CERT_CI_ECDSA (or the CI public key) for verification of the TLS certificates of SM-DP+ or SM-DS. No CRL is loaded.

The Companion Device contains a Test eUICC preconfigured as defined below in G.1.3.

G.1.3 Test eUICC Settings

Depending on the test cases and on the supported options, the Test eUICC SHALL be configured according to the following Initial States.

- The Test eUICC is configured with the ISD-R AID #ISD_R_AID and the EID #EID1.
- The Test eUICC does not contain any Profile.
- The Test eUICC is configured with the default SM-DS address #TEST_ROOT_DS_ADDRESS.
- The Test eUICC contains #TEST_DP_ADDRESS1 as default SM-DP+ address.

The ECASD is configured with at least the following Keys and Certificates based on NIST P-256 [11] or on brainpoolP256r1 [8] for this version of the SGP.23:

- The Test eUICC's Private Key #SK_EUICC_ECDSA (for creating ECDSA signatures)
- The Test eUICC's Certificate #CERT_EUICC_ECDSA (for eUICC authentication) containing the eUICC's Public Key #PK_EUICC_ECDSA
- The GSMA Certificate Issuer's Public Key #PK_CI_ECDSA (for verifying off-card entities certificates)
- The Certificate of the EUM #CERT_EUM_ECDSA

Other Certificates and Keys MAY be present. No CRL is loaded on the Test eUICC.

The Test eUICC SHALL support all certificate chain variants as defined in SGP.22[2] for verification, variant O and at least one of the new certificate chain variant (variant A, B, or C) for signing. Therefore both, euiccCiPKIdListForSigning and euiccCiPKIdListForSigningV3 SHALL be present.

The CI, identified as highest priority in euiccCiPKIdListForSigning or euiccCiPKIdListForSigningV3, is also selectable in the euiccCiPKIdListForVerification (i.e. all EUM and eUICC Certificates lead to a Root CI certificate linked to a #PK_CI_ECDSA contained in the eUICC).

This CI corresponds to the SubjectKeyIdentifier of one of the #CERT_CI_ECDSA defined in sections G.2.2 and G.2.3.

For devices supporting O_D_REMOVABLE_DOWNLOAD_PPR, the Test eUICC SHALL contain the RAT configuration specified in #PPRS_ALLOWED.

For devices supporting a removable eUICC but not supporting O_D_REMOVABLE_DOWNLOAD_PPR, the Test eUICC can be configured with any RAT.

For devices supporting a non-removable eUICC:

- For some combinations of device options, RAT configurations with certain constraints are required for some sequences, as specified below. These constraints can be

satisfied using any valid RAT table; for example, Allowed Operators can be specified explicitly or using wildcards.

Device option(s) supported	RAT configuration of Test eUICC
O_D_EMB_ALLOWS_PPR1_EUC_REQ	PPR1 is allowed and End User Consent is required for #MCC_MNC4 with gid1 and gid2 absent.
O_D_EMB_ALLOWS_PPR2_EUC_REQ	PPR2 is allowed and End User Consent is required for #MCC_MNC2 with gid1 and gid2 absent.
NOT O_D_EMB_ALLOWS_PPR1_EUC_REQ AND O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ	PPR1 is allowed and End User Consent is not required for #MCC_MNC4 with gid1 and gid2 absent.
NOT O_D_EMB_ALLOWS_PPR2_EUC_REQ AND O_D_EMB_ALLOWS_PPR2_EUC_NOT_REQ	PPR2 is allowed and End User Consent is not required for #MCC_MNC2 with gid1 and gid2 absent.

- If none of the constraints above apply, the Test eUICC can be configured with any RAT.
- Note: in the current version of this document, it is possible to satisfy the relevant constraints above with a single RAT configuration. It is recommended to supply a single device for testing with the RAT configuration satisfying all of the relevant constraints above, rather than to supply multiple devices.

A separate Test eUICC needs to be provided for each additional RAT configuration (not used in this version of the test specification). In case the Test eUICC is non-removable the additional Device SHALL contain the same software and hardware except the Test eUICC configuration.

If the DUT indicates support of the O_D_MEP then the Test eUICC SHALL support at least one of the MEP modes (mode A-1/ mode A-2/ mode B).

G.2 VOID

G.3 VOID

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 VOID

Annex K Document Management

K.1 Document History

Version	Date	CR No	Brief Description of Change	Entity	Approval Authority	Editor / Company
v1.0	9th June 2017		Initial version of SGP.23 v1.0 Test Specification		PSMC	Yolanda Sanz, GSMA
v1.1	28th Sept 2017		Minor version of SGP.23 Test specifications		RSPLEN	Yolanda Sanz, GSMA
v1.2	3rd Jan 2018		Minor version of SGP.23 Test specifications		RSPLEN	Yolanda Sanz, GSMA
SGP.23-2 v3.0	01 December 2023		Divide SGP.23 in three different documents (SGP.23-1, 2 and 3)		ISAG	Guido Abate/ST Microelectronics
		CR2004 R00	LPA_GetBPP_Response_Correction	LPA		
		CR2011 R02	LPA_DEVICE_INFO			
		CR2012 R00	LPA_AuthenticateClient_Corrections			
		CR2013 R02	LPA_GetBoundProfilePackage_Restart			
		CR2014 R00	LPA_R_EUICC_INFO1_svn			
		CR2016 R00	LPA_Avoid_EU_Rejection_Option			
		CR2024 R00	LPA_IUT_settings corrected			
		CR2025 R00	LPA_R_EUICC_INFO2_correction			
		CR2027 R02	LPA_Editorial_Constants_Definitions			
		CR2028 R01	LPA_SetNickname_Condtional			
		CR2029 R00	LPA_AC_err_16-18			
		CR2030 R00	LPA_eUICC_Memory_Reset_Corrections_Clarifications			
		CR2032 R01	LPA_Initial_Condition_Notifications			

		CR2034 R00	LPA_IA_err_7,9			
		CR2035 R01	LPA_Notifications_Before_Power_Off			
		CR2038 R00	LPA_Authenticated_Confirmation_Tested_Elsewhere			
		CR2039 R01	LPA_Allowed_Notification_on_PIR_Error			
		CR2033 R00	LPA_HTTPS_Static_RSA_DH_Checks			
		CR2031 R02	UL_LPA_eUICC_Memory_Reset_Multiple_Notification_Correction			
		CR2040 R00	LPA_CancelSession_Error_Clarifications			
		CR2041 R01	LPA_AddProfile_PPRs_Corrections			
		CR2045 R01	COMPRION_LPA_SM-DS_initial_conditions			
		CR2046 R00	LPA_Editorial_Direction_Corrected			
		CR2051 R00	LPA_5.4.1.2.11_Only_IC6_Conditional			
		CR2057 R00	Update_SIMalliance_Reference			
		CR2058 R00	Editorial_REQ_update			
		CR2063 R01	LPA_Multi_SM-XX			
		CR2064 R00	LPA_Invalid_Activation_Code			
		CR2066 R00	LPA_EditNickname_Initial_Condition			
		CR2068 R01	SM_DP+_LPA_ES11_Editorials			
		CR2069 R01	UL_LPA_ES11_S_SM-DP+_Availability			
		CR2065 R00	LPA_Delete_Acknowledge_Consequences			
		CR2075 R02	LPA_AddInfo_MTD_HTTP_REQ_MTD_HTTP_RESP			
		CR2077 R0	LPAd_Precondition_ProfileWithPPR1_enabled			

		CR2083 R01	LPAd_MatchingID_Empty_or_Missing			
		CR2084 R03	COMPRION_LPAd_SetEditNickname			
		CR2093 R01	LPAd_SM-X_Correction_of_Server_Responses			
		CR2094 R02	SM-XX_EUICC_LPA_Update_SGP.22_Version_information			
		CR2095 R00	LPA_Additonal_TLS_Verification_Optional			
		CR2100 R02	LPA_Missing_MatchingId_Section_4_4_28	LPA		
		CR2107 R00	LPA_SM-XX_Expired_Status			
		CR2110 R00	LPA_CR2034_Correction			
		CR2111 R00	LPA_Remove_O_D_ENPROF1ST			
		CR2115 R01	LPA_SM-XX_ASN1_DEFAULT_Processing			
		CR2116 R01	LPA_SM-XX_Content-Type_UTF8			
		CR2037 R02	LPA_Add_TC_MATCHING_ID_EMPTY			
		CR2043 R00	Update figure of scope for Devices			
		CR2096 R01	LPAd_Usage_Old_OTPKeys_NewTC			
		CR2118 R01	LPA_FieldsRenameFrom_notificationEnable_to_NotificationLocalEnable	LPA		
		CR1500 R01	LPA_Initiate_Power_Download_Via_SM-DS			
		CR1501 R00	LPA_Remove_Stored_On_Euicc			
		CR2131 R01	Device_URLs_replacement	LPA		
		CR2121 R02	Device_REQs_ReplacedBy_RefToSectionInSGP.22	LPA		
		CR1600 R03	LPAd_TAC_Verification	LPA		
		CR1601 R01	LPAd_IMEI_Verification	LPA		

			Alignment with SGP.23 v1.10 Draft 8	LPA		
		CR2192 R03	AuthenticateClientV3_RPM	LPA		
		CR2194 R01	AuthenticateClientV3_RPM_CancelSession	LPA		
		CR0193 R02	AuthenticateClientV3_RPM_userConsent	LPA		
		CR2195 R00	AuthenticateClientV3_correction	LPA		
		CR2196 R01	Initiate_Authentication_RPM	LPA		
		CR2197 R02	Initiate_Authentication_RPM_SM-DS_Event	LPA		
		CR2198 R01	Initiate_Authentication_rpmPending	LPA		
		CR11037R01	SIMalliance References cleanup	LPA		
		CR11038R01	LPA_fixes			
		CR11042R02	eUICCMemoryReset_fixes			
		CR11103R01	Update_AddProfile_with_an_enabled_PPR1_Profile	LPA		
		CR2199 R04	AuthenticateClientV3_RPM_CancelSession_2_3	LPA		
		CR2200 R01	AuthenticateClientV3_RPM_userConsent_correction			
		CR2202 R03	RPM Command Execution – EnableProfile_Seq1			
		CR2203 R02	RPM Command Execution – EnableProfile_Seq2			
		CR2205 R02	RPM Command Execution – Delete_2_ListProfileInfo_1_UpdateMD_1			
		CR2206 R01	RPM Notification – RPR_and_RPMEnable_Notification			
		CR2207 R00	RPM Command Execution – EnableProfile_Seq2_Correction			
		CR2201 R02	Authenticate Client Variant A for RPM – PPR Update	LPA		
		CR2204 R02	RPM Command Execution – Disable_Delete_Profile_Seq1			

		CR2212 R01	RPM Command Execution – EnableProfile_with_rpmCommandResultDataError	LPA		
		CR2213 R01	RPM Command Execution – DisableProfile_with_rpmCommandResultDataError			
		CR2214 R01	RPM Command Execution – DeleteProfile_with_rpmCommandResultDataError			
		CR11401R00	LPA_Ensure_Profile_content_and_Profile_Metadata_consistent	LPA		
		Editor's review	Removed the Requirements columns in Test Cases and Procedures. Ordered alphabetically responses in section D.3.2.	LPA		
		Editor's review	Action eSIMWG3.88_AP05 "to remove from the procedure when the following sentence is present "For Server testing (SM-DP+ or SM-DS), executive the following steps." "			
		CR2363 R01	TLS_Handling			
		CR2364 R01	HTTPS_TCs_v3_update			
		CR2210 R02	ConditionNbs_Update			
		CR2215 R00	RPM Command Execution – correction for MTD ListProfileInfo and Seq 1			
		CR2216 R00	RPM Command Execution – ListProfileInfo_2_3			
		CR2218 R01	RPM Command Execution – ListProfileInfo_4_5			
		CR2219 R01	RPM Command Execution – UpdateMD_2_3			
		CR2346 R00	Removing_references_SGP.22_v2.x			
		CR2361 R01	eUICCInfo_v3_Update			
		Editor's review	Accepted removal of test cases for eUICC and Servers	LPA		
		CR2365 R02	InitiateAuthentication_v3_update	LPA		

		CR2366 R01	AuthenticateClient_v3_Update			
		CR2367 R00	Clean-up_Server_and_v3_related_content			
		CR2386 R00	GetBPP_Update_V3			
		CR2387 R00	HandleNotification_Update_V3			
		CR2388 R00	CancelSession_Update_v3			
		CR2389 R00	Section_5_Minor_V3_Updates			
		CR2396 R00	Missing_Applicability			
		CR2395 R01	Test_eUICC_Req			
		CR2393 R01	New_TCs_MEP			
		CR2385 R00	v3.1_replacing_v3.0_in_references	LPA		
		CR2390 R01	QRCodeScanning_Update_v3			
		CR2397 R00	Test_eUICC_Requirements_MEP			
		CR2399 R00	Fixing_version_3.0_to_3.1			
		CR2427 R00	NoErrorUpdate	LPA		
		CR11505R00	NoErrorUpdate	LPA		

K.2 Other Information

Type	Description
Document Owner	Yolanda Sanz, GSMA
Editor / Company	Guido Abate, STMicroelectronics

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.

Your comments or suggestions & questions are always welcome.