SGP.33-2 IPA Test Specification
Version 1.2
27 January 2025

Security Classification: Non-confidential

Copyright Notice

Disclaimer

Compliance Notice

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

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

Table of Contents

1 Introduction

1.1 Overview

The main aim of the eSIM IoT specifications [2] & [3] is to provide solution for the Remote SIM Provisioning of IoT Devices.

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

1.2 Scope

This document is intended for:

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

The Test Plan consists of a set of relevant test cases for testing the IPAd. The Implementations Under Test (IUT) are:

- the IPA (IoT Profile Assistant)

NOTE: The IPAe is tested in SGP.33-1 [XX].

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 and SGP.32:

- GSMA RSP Technical Specification [4]
- GSMA IoT eSIM Technical Specification [31]

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

1.3 Definition of Terms

In addition to the terms which are defined below, the terms defined in SGP.22 [2] and SGP.32 [31] also apply

Term	Description
Integrated eUICC Test Interface	An external interface provided by its manufacturer for the purpose of testing eUICC functionality.
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] and SGP.32 [31] 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
IPA	IoT Profile Assistant
IUT	Implementation Under Test
KVN	Key Version Number
LPA	Local Profile Assistant
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" V4.3
[2]	SGP.22	RSP Technical Specification V2.5
[3]	SGP.21	RSP Architecture V2.5

Ref	Document Number	Title
[4]	eUICC Profile Package	Trusted Connectivity Alliance (formerly SIMalliance) eUICC Profile Package: Interoperable Format Technical Specification V2.1 or later
[5]	ETSI TS 102 221	Smart Cards; UICC-Terminal interface
[6]	GPC_SPE_034	GlobalPlatform Card Specification v.2.3
[7]	ISO/IEC 7816-4:2013	Identification cards – Integrated circuit cards - Part 4: Organization, security and commands for interchange
[8]	RFC 5639	Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation
[9]	ANSSI ECC FRP256V1	Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français. JORF n°0241 du 16 octobre 2011 page 17533. texte n° 30. 2011
[10]	ITU E.118	The international telecommunication charge card
[11]	NIST SP 800-56A	NIST Special Publication SP 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revision 2), May 2013
[12]	3GPP TS 23.003	Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification
[13]	ETSI TS 102 225	Secured packet structure for UICC based applications; Release 12
[14]	ETSI TS 102 226	Remote APDU structure for UICC based applications; Release 9
[15]	TS.26	GSMA NFC Handset Requirements V9.0
[16]	ITU-T X.690 (11/2008)	ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER) including Corrigendum 1 and 2
[17]	ETSI TS 102 241	Smart cards; UICC Application Programming Interface (UICC API) for Java Card™
[18]	3GPP TS 31.102	Characteristics of the Universal Subscriber Identity Module (USIM) application
[19]	GPC_SPE_095	GlobalPlatform Card - Digital Letter of Approval - Version 1.0
[20]	RFC 2119	Key words for use in RFCs to Indicate Requirement Levels, S. Bradner http://www.ietf.org/rfc/rfc2119.txt
[21]	Void	
[22]	3GPP TS 23.040	Technical realization of the Short Message Service (SMS)
[23]	Void	
[24]	RFC 4492	Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)
[25]	SGP.26	RSP Test Certificates Definition v3.0.2

Ref	Document Number	Title
[26]	3GPP TS 29.002	Mobile Application Part (MAP) specification
[27]	RFC 5246	The Transport Layer Security (TLS) Protocol Version 1.2
[28]	GSMA PRD AA.35	Procedures for Industry Specifications Product
[29]	CCID Rev 1.1	CCID Specification for Integrated Circuit(s) Cards Interface Devices
[30]	SGP.31	eSIM IoT Architecture and Requirement Specification Version 1.2
[31]	SGP.32	eSIM IoT Technical Specification Version 1.2
[32]	SGP.23	SGP.23 Test Specifcation v1.15

1.6 Conventions

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

2 Testing Rules

2.1 Applicability

2.1.1 Format of the Optional Features Table

The columns in Table 4 have the following meaning:

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

Table 1: Format of the Optional Features Table

2.1.2 Format of the Applicability Table

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

The columns in Table 5 have the following meaning:

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

Test Env.	Test environment used for executing the test case.

Table 2: Format of the Applicability Table

2.1.3 Applicability and Notations

The following notations are used for the Applicability column:

Applicability code	Meaning
M	mandatory - the capability is required to be supported.
N/A	not applicable - in the given context, it is impossible to use the capability.
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 IPAd	O_D_IPAD
The Device supports eIM Package retrieval	O_D_EIM_PACKAGE_RETRIEVAL
The Device supports eIM Package injection	O_D_EIM_PACKAGE_INJECTION
The Device supports direct profile download	O_D_DIRECT_DOWNLOAD
The Device supports fallback mechanism	O_D_FALLBACK
The Device supports indirect profile download	O_D_INDIRECT_DOWNLOAD
The Device supports HTTPS connection on ESipa interface	O_D_ESIPA_HTTPS
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 NOT required.	O_D_EMB_ALLOWS_PPR2_EUC_NOT_REQ
The Device sends eUICC Package Result to the eIM using ESipa.HandleNotification.	O_D_ESIPA_HANDLE_NOTIF
The Device sends eUICC Package Result to the eIM using ESipa.ProvideEimPackageResult.	O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT

Table 4: Options

NOTE: this table should contain the IPA Options dedicated to IoT test cases. Those test cases that are applicable as SGP.23 [8] test cases should use the IPA Options as defined by SGP.23 [8].

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	Role	V1.0	Test Env.
4.2.21.1	TC_IPAd_InitiateAuthentication_Nominal	IPAd	SGP.23	C3005
4.2.21.2.2	TC_IPAd_InitiateAuthentication_ErrorCases	IPAd	SGP.23	C3005
4.2.22.2.1	TC_IPAd_ES9+_GetBoundProfilePackage_Nominal	IPAd	SGP.23	C3005
4.2.22.2.2	TC_IPAd_ES9+_GetBoundProfilePackage_Retry	IPAd	SGP.23	C3005
4.2.22.2.3	TC_IPAd_ES9+_GetBoundProfilePackage_Error	IPAd	SGP.23	C3005
4.2.23.2.1	TC_IPAd_ES9+_AuthenticatClient_Nominal	IPAd	SGP.23	C3005
4.2.23.2.2	TC_IPAd_ES9+_AuthenticateClient_ErrorCases	IPAd	SGP.23	C3005
4.2.24.2.1	TC_IPAd_ES9+_HandleNotification_Nominal All test sequences except the sequence #03	IPAd	SGP.23	C3005
4.2.24.2.1	TC_IPAd_ES9+_HandleNotification_Nominal Only the test sequence #03	IPAd	SGP.23	C3005
4.2.25.2.1	TC_IPAd_ES9+_CancelSession_Nominal	IPAd	SGP.23	C3005
4.2.26.2.1	TC_IPAd_HTTPS_Nominal	IPAd	C3005	
4.2.26.2.2	TC_IPAd_HTTPS_ErrorCases	IPAd	C3005	
4.2.27.2.1	TC_IPAd_ES11_InitiateAuthentication_Nominal	IPAd	C3007	
4.2.27.2.2	TC_IPAd_ES11_InitiateAuthentication_ErrorCases	IPAd	C3007	
4.2.28.2.1	TC_IPAd_ES11_AuthenticateClient_Nominal	IPAd	C3007	
4.2.28.2.2	TC_IPAd_ES11_AuthenticateClient_ErrorCases	IPAd	C3007	

Test case	Name	Role	V1.0	Test Env.
4.2.29.2.1	TC_IPAd_HTTPS_Nominal	IPAd	C3007	
4.2.29.2.2	TC_IPAd_HTTPS_Error	IPAd	C3007	
5.4.1.2.1	TC_IPAd_DirectProfileDownload_IPA_initiated_with_Activation_Code	IPAd	C3001	
5.4.1.2.2	TC_IPAd_DirectProfileDownload_IPA_initiated_ActivationCode_InvalidFormat	IPAd	C3001	
5.4.1.2.3	TC_IPAd_DirectProfileDownload_IPA_initiated_with_ConfirmationCode_smdpSigned2	IPAd	C3001	
5.4.1.2.4	TC_IPAd_DirectProfileDownload_IPA_initiated_default SM-DP+	IPAd	C3001	
5.4.1.2.5	TC_IPAd_DirectProfileDownload_IPA_initiated_with_PPRs Only test sequence #1	IPAd	C3002	
5.4.1.2.5	TC_IPAd_DirectProfileDownload_IPA_initiated_with_PPRs Only test sequence #2	IPAd	C3003	
5.4.1.2.5	TC_IPAd_DirectProfileDownload_IPA_initiated_with_PPRs Only test sequence #3	IPAd	C3004	
5.4.1.2.6	TC_IPAd_DirectProfileDownload_IPA_initiated_with_Empty MatchingID	IPAd	C3001	
5.4.1.2.7	TC_IPAd_DirectProfileDownload_IPA_initiated_default SM-DP+_immediate_profile_enabling	IPAd	C3001	
5.4.4.2.1	TC_IPAd_DeleteProfile_Disabled_without_PPR_IPA_initiated	IPAd	C3001	
5.4.4.2.2	TC_IPAd_DeleteProfile_Enabled_without_PPR_IPA_initiated	IPAd	C3001	
5.4.4.2.3	TC_IPAd_DeleteProfile_Error_with_PPR1_IPA_initiated	IPAd	C3002	
5.4.4.2.4	TC_IPAd_DeleteProfile_Error_Disabled_with_PPR2_IPA_initiated	IPAd	C3003	
5.4.4.2.5	TC_IPAd_DeleteProfile_Error_Enabled_with_PPR2_IPA_initiated	IPAd	C3003	
5.4.5.2.1	TC_IPAd_EnableProfile_IPA_initiated	IPAd	C3001	

Test case	Name	Role	V1.0	Test Env.
5.4.5.2.2	TC_IPAd_EnableProfile_ImplicitDisable_IPA_initiated	IPAd	C3001	
5.4.5.2.3	TC_IPAd_EnableProfile_Error_ProfileAlreadyEnabled_IPA_initiated	IPAd	C3001	
5.4.5.2.4	TC_IPAd_EnableProfile_Error_PPR1Set_IPA_initiated	IPAd	C3002	
5.4.5.2.5	TC_LPAd_EnableProfile_Error_RollbackNoEnabledProfile_IPA_initiated	IPAd	C3001	
5.4.6.2.1	TC_IPAd_DisableProfile_IPA_initiated	IPAd	C3001	
5.4.6.2.2	TC_IPAd_DisableProfile_Error_ProfileAlreadyDisabled_IPA_initiated	IPAd	C3001	
5.4.6.2.3	TC_IPAd_DisableProfile_Error_PPR1Set_IPA_initiated	IPAd	C3002	
5.4.12.2.1	TC_IPAd_SetFallbackAttribute	IPAd	C3008	
5.4.12.2.2	TC_IPAd_SetFallbackAttribute_Error_Target_Profile_Not_Available	IPAd	C3008	
5.4.12.2.3	TC_IPAd_SetFallbackAttribute_Error_Not_Allowed	IPAd	C3008	
5.4.12.2.4	TC_IPAd_SetFallbackAttribute_Error_Fallback_Profile_Already_Enabled	IPAd	C3008	
5.4.13.2.1	TC_IPAd_UnsetFallbackAttribute	IPAd	C3008	
5.4.13.2.2	TC_IPAd_UnsetFallbackAttribute_Error_Fallback_Profile_Enabled	IPAd	C3008	
5.4.13.2.3	TC_IPAd_UnsetFallbackAttribute_Error_No_Fallback_Profile	IPAd	C3008	

Table 5: Applicability of Tests

Conditional item	Condition
C3001	IF O_D_IPAD AND O_D_DIRECT_DOWNLOAD AND O_D_EIM_PACKAGE_RETRIEVAL AND O_D_ESIPA_HTTPS THEN M ELSE N/A
C3002	IF (O_D_IPAD AND O_D_DIRECT_DOWNLOAD AND O_D_EIM_PACKAGE_RETRIEVAL AND O_D_ESIPA_HTTPS AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ)) THEN M ELSE N/A
C3003	IF (O_D_IPAD AND O_D_DIRECT_DOWNLOAD AND O_D_EIM_PACKAGE_RETRIEVAL AND O_D_ESIPA_HTTPS AND

Conditional item	Condition
	(O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR2_EUC_NOT_REQ)) THEN M ELSE N/A
C3004	IF (O_D_IPAD AND O_D_DIRECT_DOWNLOAD AND O_D_EIM_PACKAGE_RETRIEVAL AND O_D_ESIPA_HTTPS AND O_D_ADDPREPPR1 AND (O_D_REMOVABLE_DOWNLOAD_PPR OR O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ)) THEN M ELSE N/A
C3005	IF (O_D_IPAD AND O_D_DIRECT_DOWNLOAD) THEN M ELSE N/A
C3007	IF (O_D_IPAD AND O_D_DIRECT_DOWNLOAD AND O_D_SMDS) THEN M ELSE N/A
C3008	IF O_D_IPAD AND O_D_DIRECT_DOWNLOAD AND O_D_EIM_PACKAGE_RETRIEVAL AND O_D_ESIPA_HTTPS AND O_D_FALLBACK THEN M ELSE N/A

Table 6: Conditional Items Referenced by Table 5

2.2 General Consideration

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

2.2.1 Test Case Definition

Test descriptions are independent.

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

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

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

2.2.2 Test Cases Format

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

4.X.Y.Z Test Cases

4.X.Y.Z.1 TC_IUT_TestName1

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

Test Sequence #01: Short Description

Description of the aim of the test sequence N°1

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

Step	Direction	Sequence / Description	Expected result	REQ
IC1	Entity1 → Entity2	Command or Message to send from Entity1 to Entity2	Expected result N°1.1	
1	Entity1 → Entity2	Command or Message to send from Entity1 to Entity2	1- expected result N°1.2 2- expected result N°1.3	REQ1
2	Entity2 → Entity3	Command or Message to send from Entity2 to Entity3		

Test Sequence #02

Description of the aim of the test sequence N°2

Step	Direction	Sequence / Description	Expected result	REQ
1	Entity1 → Entity2	Command or Message to send from Entity1 to Entity2		
2	Entity2 → Entity3	Command or Message to send from Entity2 to Entity3	1- expected result N°2.1 2- expected result N°2.2	REQ2

4.X.Y.Z.2 TC_IUT_TestName2

…

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

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

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

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

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

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

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

The tables defining the different initial conditions are optional.

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

Unless otherwise defined, 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.

2.2.3 General Rules for Device Testing

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

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

The default way to execute the Profile download SHALL be the Add Profile procedure with Activation Code #ACTIVATION_CODE_1. The way to apply the Activation Code (manual typing or QR code scanning) depends on the Device/LPAd implementation. In order to execute the Common Mutual Authentication procedure and the Sub-procedure Profile Download and Installation (End User Confirmation), the following responses SHALL be sent by the S_SM-DP+:

- #INITIATE_AUTH_OK
    - with the <EUICC_CI_PK_ID_TO_BE_USED> set to the CI for signing indicated as highest priority in euiccCiPKIdListForSigning in the #R_EUICC_INFO1
    - with the #CERT_S_SM_DPauth_ECDSA leading to the same CI as the one chosen for signing
    - with the SM-DP+ address #TEST_DP_ADDRESS1
- #AUTH_CLIENT_OK
    - with the #CERT_S_SM_DPpb_ECDSA leading to the same CI as the one chosen for signing

  - Metadata of the downloaded Profile instead of #METADATA_OP_PROF1

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

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

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

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

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

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

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

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

2.2.3.2 TLS Testing Recommendations

The TLS connection may be rejected either:

- by sending a TLS alert, or
- by closing of the TCP connection, though TLS handshake completed, or
- TLS handshake not completed without sending a TLS alert, or
- No further RSP communication has been initiated by LPAd on ES9+/ES11 within the #IUT_LPAd_SESSION_CLOSE_TIMEOUT

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

2.2.4 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.

2.2.5 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.

2.2.6 Adaptation of ES9+ and ES11 test cases

For ES9+ test cases when the General Initial Condition, or the Initial Condition in SGP.23 says that "Add Profile operation is initiated by using #ACTIVATION_CODE_X" in this document this is equivalent with the Profile Download Trigger Request with ACTIVATION_CODE_X is received by the IPA. The Profile Download Trigger Request is sent without eimTransactionId.

The IPA is expected to send Handle Notification with profileDownloadTriggerResult on ESipa interface. It needs to be handled by the test tool, but it does not need to be validated, because it is not in the scope of ES9+ testing. However, it is validated in chapter 5.

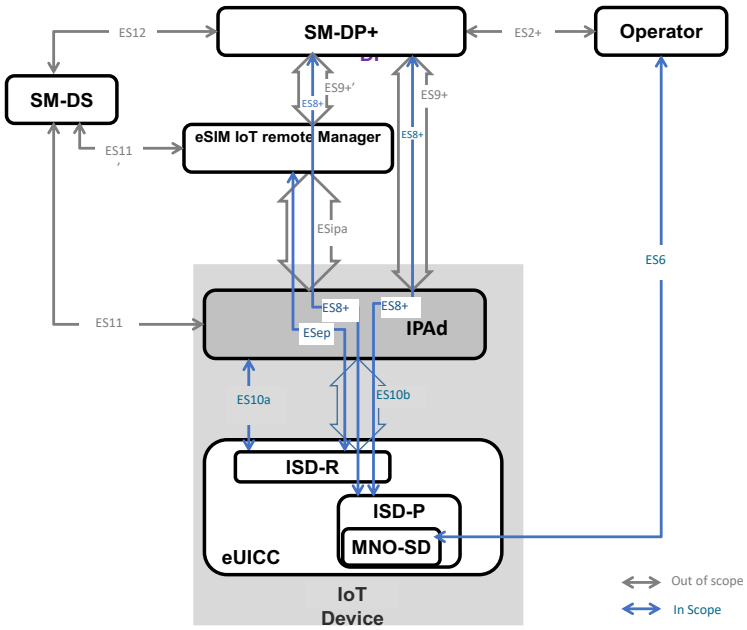
For ES11 test cases when the General Initial Condition says "The Profile Download is initiated using SM-DS" in this document this is equivalent with the Profile Download Trigger Request with contactSmds is received by the IPA. The Profile Download Trigger Request is sent without eimTransactionId and without smdsAddress.

The IPA is expected to send Handle Notification with profileDownloadTriggerResult on ESipa interface. It needs to be handled by the test tool, but it does not need to be validated, because it is not in the scope of ES11 testing. However, it is validated in chapter 5.

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	SGP.33-2
ES2+	Operator	SM-DP+	Used by the Operator to order Profiles for specific eUICCs as well as other administrative functions as defined in SGP.31 [2].	Out of scope
ES6	Operator	eUICC	Used by the Operator for the management of Operator services via OTA services as defined in SGP.31 [2].	Out of scope
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 as defined in SGP.31 [2].	Out of scope
ES9+	SM-DP+	IPA	Used to provide a secure transport between the SM-DP+ and the IPA for the delivery of the Bound Profile Package as defined in SGP.31 [2].	In scope
ES9+'	SM-DP+	eIM	Used to provide a secure transport between the SM-DP+ and the eIM for the delivery of the Bound Profile Package as defined in SGP.31 [2].	Out of scope

Interface	Between		Description	SGP.33-2
ES10a	IPA	eUICC	Used between the IPA (in the IoT Device) and the eUICC to handle a Profile discovery as defined in SGP.31 [2].	Out of scope
ES10b	IPA	eUICC	Used between the IPA (in the IoT Device) and the IPA Services to transfer a Bound Profile Package to the eUICC as defined in SGP.31 [2]. This interface plays no role in the decryption of Profile Packages.	Out of scope
ES11	IPA	SM-DS	Used by the IPA to retrieve Event Records for the respective eUICC as defined in SGP.31 [2].	In scope
ES11'	eIM	SM-DS	Used by the eIM to retrieve Event Records for the respective eUICC as defined in SGP.31 [2].	Out of scope
ES12	SM-DP+	SM-DS	Used by the SM-DP+ to issue or remove Event Registrations on the SM-DS as defined in SGP.31 [2].	Out of scope
ESep	eIM	eUICC	Logical end-to-end interface between the eIM and the eUICC used to transfer eUICC Packages for Profile State management and eIM configuration by eIM, as defined in SGP.31 [2].	Out of scope
ESipa	eIM	IPA	Logical interface between an eIM and an IPA, as defined in SGP.31 [2], used to trigger a Profile download at the IPA and to provide a secure transport for the delivery of eUICC Packages, unless the underlying transport provides necessary security.	In scope

Table 7: Interfaces Descriptions

3.2 Testing Execution

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

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

- S_Device: the Device Simulator used to send some commands to the eUICC under test using ISO/IEC 7816-4 [7] on the contact interface
- S_SM-DP+: the SM-DP+ Simulator
- S_SM-DS: the SM-DS Simulator
- S_MNO: the MNO Simulator
- S_IPAd: the LPAd Simulator
- S_LPAe: the LPAe Simulator
- S_eIM: the eIM Simulator
- S_CLIENT: the HTTPs client Simulator for the purpose of TLS testing. The S_CLIENT MAY be S_SM-DP+, S_SM-DS depending on the component under test.
- S_SERVER: the HTTPs server Simulator for the purpose of TLS testing. The S_SERVER MAY be S_SM-DP+ or S_SM-DS depending on the component under test.

- Implementation of these simulators remains under the responsibility of the test tool providers.
- The aim of all the test cases is to verify the compliance of an Actor/Component (i.e. eUICC, SM-DP+, Alternative SM-DS, Root SM-DS, LPAe, LPAd, Device).

Following notations are used:

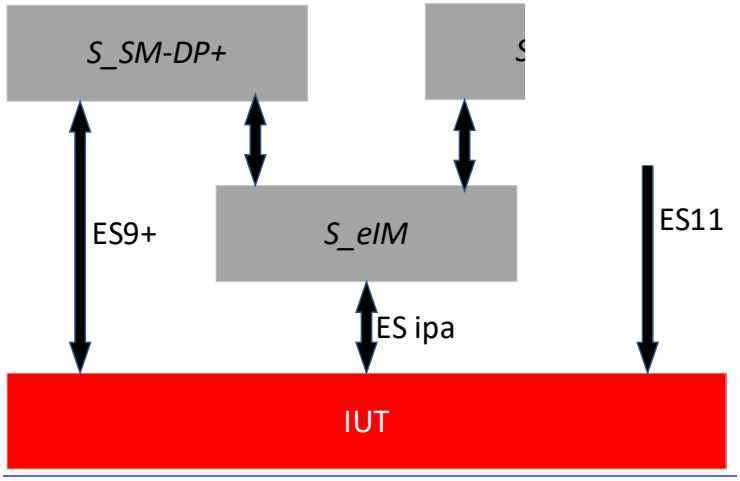
- `S_ComponentName` for a simulated component
- `ComponentName` for the Implementation Under Test (IUT)
- Where `ComponentName` is indicated by CLIENT, SERVER
- Depending on the component under test, the CLIENT MAY be the SM-DP+ or the SM-DS. The Operator component is currently out of scope.
- Depending on the component under test, the SERVER MAY be the SM-DP+ or the SM-DS. The Operator component is currently out of scope.
- The use of "-- optional" in any ASN.1 elements defined within this document indicate that the test tool SHALL allow for the value either being present with that value, or being absent.

3.2.3 Device/IPAd - Test Environment

The following test environment is used for all IPAd 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, Test Keys and eIM Configuration
- 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 IPAd testing and SHALL not be considered as an IUT
- The Test eUICC SHALL not support IPAe
- The Test eUICC SHOULD be compliant with the GSMA RSP IoT Technical Specification [31]
- SM-DP+ Simulator(s) SHALL be implemented by the test tools
- SM-DS Simulator(s) SHALL be implemented by the test tools
- No modification of the Device HW is required
- eIM Simulator(s) SHALL be implemented by the test tools. For the purpose of direct Profile download test cases, eIM Simulator(s) SHALL support at least HTTPS connection on ESipa and eIM Package retrieval by IPA.
- Test Root Certificate SHALL be configured in the Device

3.2.3.1 General (Device/IPAd) Test Environment



The Test Environment consists of:

- IUT: IoT 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_eIM: a simulated eIM supporting at least HTTPS connection to the Device
- the interface between S_eIM and S_SM_DP+ is internal to the test tool
- the interface between S_eIM and S_SM_DS is internal to the test tool

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 SGP.31[30]/SGP.32 [31] 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

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 IPAd Interfaces

4.2.1 ES10a (IPA -- eUICC): GetEuiccConfiguredAddresses

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

4.2.2 ES10b (IPA -- eUICC): LoadEUICCPackage

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

4.2.3 ES10b (IPA -- eUICC): GetEUICCInfo

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

4.2.4 ES10b (IPA -- eUICC): GetEUICCChallenge

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

4.2.5 ES10b (IPA -- eUICC): AddInitialEim

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

4.2.6 ES10b (IPA -- eUICC): eUICCMemoryReset

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

4.2.7 ES10b (IPA -- eUICC): AuthenticateServer

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

4.2.8 ES10b (IPA -- eUICC): PrepareDownload

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

4.2.9 ES10b (IPA -- eUICC): LoadBoundProfilePackage

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

4.2.10 ES10b (IPA -- eUICC): CancelSession

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

4.2.11 ES10b (IPA -- eUICC): GetCerts

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

4.2.12 ES10b (IPA -- eUICC): RetrieveNotificationList

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

4.2.13 ES10b (IPA -- eUICC): RetrieveNotificationFromList

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

4.2.14 ES10b (IPA -- eUICC): GetRAT

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

4.2.15 ES10b (IPA -- eUICC): GetProfileInfo

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

4.2.16 ES10b (IPA -- eUICC): EnableUsingDD

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

4.2.17 ES10b (IPA -- eUICC): ProfileRollBack

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

4.2.18 ES10b (IPA -- eUICC): ConfigureAutomaticProfileEnabling

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

4.2.19 ES10b (IPA -- eUICC): GetEimConfigurationData

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

4.2.20 ES10b (IPA -- eUICC): GetEID

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

4.2.21 ES9+ (IPA -- SM-DP+): InitiateAuthentication

4.2.21.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

The IPA SHALL communicate with the SM-DP+ secured by HTTPS in server authentication mode as described in SGP.22 [4] section 2.6.6.

This interface is identical to the ES9+ interface defined in section 5.6 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.21.2 Test Cases

4.2.21.2.1 TC_IPAd_InitiateAuthentication_Nominal

Test Sequence #01 Nominal: Initiate Authentication

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Initiate Authentication* defined in section 4.4.21.2.1 TC_LPAd_InitiateAuthentication_Nominal where the LPAd play the role of IPAd.

4.2.21.2.2 TC_IPAd_InitiateAuthentication_ErrorCases

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Error: Invalid SM-DP+ Address* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #02 Error: Unsupported Security Configuration

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Error: Unsupported Security Configuration* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #03 Error: Unsupported SVN

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Error: Unsupported SVN* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #04 Error: Unavailable SM-DP+ Certificate* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #05 Error: Invalid SM-DP+ Certificate* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #06 Error: Invalid SM-DP+ Signature* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #07 Error: Invalid SM-DP+ Address sent by the SM-DP+* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #08 Error: Unsupported CI Key ID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #08 Error: Unsupported CI Key ID* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #09 Error: Invalid SM-DP+ OID Address* defined in section 4.4.21.2.2 TC_LPAd_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

4.2.22 ES9+ (IPA -- SM-DP+): GetBoundProfilePackage

4.2.22.1 Conformance Requirements

References

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

The IPA SHALL communicate with the SM-DP+ secured by HTTPS in server authentication mode as described in SGP.22 [4] section 2.6.6.

This interface is identical to the ES9+ interface defined in section 5.6 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.22.2 Test Cases

4.2.22.2.1 TC_IPAd_ES9+_GetBoundProfilePackage_Nominal

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Get BPP using S-ENC and S-MAC without Confirmation Code* defined in section 4.4.22.2.1 TC_LPAd_ES9+_GetBoundProfilePackage_Nominal where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Nominal: Get BPP using S-ENC and S-MAC with Confirmation Code* defined in section 4.4.22.2.1 TC_LPAd_ES9+_GetBoundProfilePackage_Nominal where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Nominal: Get BPP using PPK-ENC and PPK-MAC without Confirmation Code* defined in section 4.4.22.2.1 TC_LPAd_ES9+_GetBoundProfilePackage_Nominal where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #04 Nominal: Get BPP using PPK-ENC and PPK-MAC with Confirmation Code* defined in section 4.4.22.2.1 TC_LPAd_ES9+_GetBoundProfilePackage_Nominal where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

4.4.22.2.2 TC_IPAd_ES9+_GetBoundProfilePackage_Retry

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Get BPP Retry after incorrect Confirmation Code* defined in section 4.4.22.2.2 TC_LPAd_ES9+_GetBoundProfilePackage_Retry where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

4.4.22.2.3 TC_IPAd_ES9+_GetBoundProfilePackage_Error

Test Sequence #01 Error: Wrong eUICC Signature

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Error: Wrong eUICC Signature* defined in section 4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

Test Sequence #02 Error: BPP Not Available

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Error: BPP Not Available* defined in section 4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Error: Unknown TransactionID received by SM-DP+* defined in section 4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

Test Sequence #04 Error: Missing Confirmation Code

This test sequence is the same as SGP.23 [32] - the *Test Sequence #04 Error: Missing Confirmation Code* defined in section 4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

Test Sequence #05 Error: Download Order Expired

This test sequence is the same as SGP.23 [32] - the *Test Sequence #05 Error: Download Order Expired* defined in section 4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

Test Sequence #06 Error: Wrong Confirmation Code

This test sequence is the same as SGP.23 [32] - the *Test Sequence #06 Error: Wrong Confirmation Code* defined in section 4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #07 Error: Maximum number of Confirmation Code retries exceeded* defined in section 4.4.22.2.3 TC_LPAd_ES9+_GetBoundProfilePackage_Error where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

4.2.23 ES9+ (IPA -- SM-DP+): AuthenticateClient

4.2.23.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

The IPA SHALL communicate with the SM-DP+ secured by HTTPS in server authentication mode as described in SGP.22 [4] section 2.6.6.

This interface is identical to the ES9+ interface defined in section 5.6 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.23.2 Test Cases

4.2.23.2.1 TC_IPAd_AuthenticateClient_Nominal

Test Sequence #01 Nominal: Authenticate Client without Confirmation Code

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Authenticate Client without Confirmation Code* defined in section 4.4.23.2.1 TC_LPAd_AuthenticateClient_Nominal where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

Test Sequence #02 Nominal: Authenticate Client with Confirmation Code

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Nominal: Authenticate Client with Confirmation Code* defined in section 4.4.23.2.1 TC_LPAd_AuthenticateClient_Nominal where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Nominal: Authenticate Client with Confirmation Code Retry* defined in section 4.4.23.2.1 TC_LPAd_AuthenticateClient_Nominal where the LPAd play the role of IPAd.

The method to provide confirmation code to IPAd is IPAd dependant.

4.2.23.2.2 TC_IPAd_AuthenticateClient_ErrorCases

Test Sequence #01 Error: Invalid EUM Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Error: Invalid EUM Certificate* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #02 Error: Expired EUM Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Error: Expired EUM Certificate* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #03 Error: Invalid eUICC Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Error: Invalid eUICC Certificate* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #04 Error: Expired eUICC Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #04 Error: Expired eUICC Certificate* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #05 Error: Invalid eUICC Signature or serverChallenge

This test sequence is the same as SGP.23 [32] - the *Test Sequence #05 Error: Invalid eUICC Signature or serverChallenge* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #06 Error: Insufficient Memory

This test sequence is the same as SGP.23 [32] - the *Test Sequence #06 Error: Insufficient Memory* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #07 Error: Unknown CI Root Key

This test sequence is the same as SGP.23 [32] - the *Test Sequence #07 Error: Unknown CI Root Key* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #08 Error: Profile not Allowed (Not in 'released' State)* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #09 Error: Unknown TransactionID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #09 Error: Unknown TransactionID* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #10 Error: Refused MatchingID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #10 Error: Refused MatchingID* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #11 Error: Refused EID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #11 Error: Refused EID* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #12 Error: No Eligible Profile for this eUICC/Device* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #13 Error: Expired Download Order

This test sequence is the same as SGP.23 [32] - the *Test Sequence #13 Error: Expired Download Order* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #14 Error: Maximum Number of Retries Exceeded

This test sequence is the same as SGP.23 [32] - the *Test Sequence #14 Error: Maximum Number of Retries Exceeded* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #15 Error: Invalid SM-DP+(pb) certificate* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #16 Error: Different OID for SM-DP+ Certificates (CERT.DPpb.ECDSA and CERT.DPauth.ECDSA not belonging to the same entity)* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #17 Error: Invalid SM-DP+ signature (smdpSignature2)* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #18 Error: Invalid TransactionID sent by SM-DP+* defined in section 4.4.23.2.2 TC_LPAd_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

4.2.24 ES9+ (IPA – SM-DP+): HandleNotification

4.2.24.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

The IPA SHALL communicate with the SM-DP+ secured by HTTPS in server authentication mode as described in SGP.22 [4] section 2.6.6.

This interface is identical to the ES9+ interface defined in section 5.6 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.24.2 Test Cases

4.2.24.2.1 TC_IPAd_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] or SGP.23 [31] for the number of Notifications that can be stored by the eUICC.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Successful PIR and Install Notifications to the Same SM-DP+ Address* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Nominal: Successful PIR and Enable Notifications to the Same SM-DP+ Address* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Nominal: Disable and Delete Notifications to the Same SM-DP+ Address* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #04 Nominal: Enable and Disable Notifications with Different SM-DP+ Addresses* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #05 Nominal: Different SM-DP+ Addresses in PIR and Install Notifications* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

Test Sequence #06 Nominal: Profile Download with PIR Failed

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Successful PIR and Install Notifications to the Same SM-DP+ Address* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

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.

This test sequence is the same as SGP.23 [32] - the *Test Sequence #08 Nominal: No Acknowledge for Successful PIR results in No Further Notifications* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

Test Sequence #09 Nominal: Disable and Delete Notifications to the Same SM-DP+ Address using Delete Operation

This test sequence is the same as SGP.23 [32] - the *Test Sequence #09 Nominal: Disable and Delete Notifications to the Same SM-DP+ Address using Delete Operation* defined in section 4.4.24.2.1 TC_LPAd_ES9+_HandleNotification_Nominal where the LPAd play the role of IPAd.

4.2.25 ES9+ (IPA – SM-DP+): CancelSession

4.2.25.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

The IPA SHALL communicate with the SM-DP+ secured by HTTPS in server authentication mode as described in SGP.22 [4] section 2.6.6.

This interface is identical to the ES9+ interface defined in section 5.6 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.25.2 Test Cases

4.2.25.2.1 TC_IPAd_ES9+_CancelSession_Nominal

Test Sequence #01 Nominal: Profile Download with PPR1 not allowed due to Operational Profile already present

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Profile Download with PPR1 not allowed due to Operational Profile already present* defined in section 4.4.25.2.1 TC_LPAd_ES9+_CancelSession_Nominal where the LPAd play the role of IPAd.

Test Sequence #02 Nominal: Load BPP Error

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Nominal: Load BPP Error* defined in section 4.4.25.2.1 TC_LPAd_ES9+_CancelSession_Nominal where the LPAd play the role of IPAd.

Test Sequence #03 Nominal: Load BPP Error due to unknown TAG

This test sequence is the same as SGP.23 [32] - the *Test Sequence #05 Nominal: Load BPP Error due to unknown TAG* defined in section 4.4.25.2.1 TC_LPAd_ES9+_CancelSession_Nominal where the LPAd play the role of IPAd.

4

4.2.26 ES9+ (IPA – SM-DP+): HTTPS

4.2.26.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

The IPA SHALL communicate with the SM-DP+ secured by HTTPS in server authentication mode as described in SGP.22 [4] section 2.6.6.

This interface is identical to the ES9+ interface defined in section 5.6 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.26.2 Test Cases

4.2.26.2.1 TC_IPAd_HTTPS_Nominal

Test Sequence #01 Nominal: HTTPS Session Establishment

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: HTTPS Session Establishment* defined in 4.4.26.2.1 TC_LPAd_HTTPS_Nominal where the LPAd play the role of IPAd.

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.

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Nominal: non-reuse of session keys* defined in 4.4.26.2.1 TC_LPAd_HTTPS_Nominal where the LPAd play the role of IPAd.

4.2.26.2.2 TC_IPAd_HTTPS_ErrorCases

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Error: Invalid (SM-DP+) TLS Certificate signature* defined in 4.4.26.2.2 TC_LPAd_HTTPS_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #02 Error: Expired TLS Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Error: Expired TLS Certificate* defined in 4.4.26.2.2 TC_LPAd_HTTPS_ErrorCases where the LPAd play the role of IPAd.

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)

This test sequence is the same as SGP.23 [32] - the *Test Sequence #07 Error: Invalid TLS Certificate based on Invalid CI (Invalid Curve)* defined in 4.4.26.2.2 TC_LPAd_HTTPS_ErrorCases where the LPAd play the role of IPAd.

4.2.27 ES11 (IPA – SM-DS): InitiateAuthentication

4.2.27.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This interface is identical to the ES11 interface defined in section 5.8 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.27.2 Test Cases

4.2.27.2.1 TC_IPAd_ES11_InitiateAuthentication_Nominal

Test Sequence #01 Nominal: Initiate Authentication

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Initiate Authentication* defined in section 4.4.27.2.1 TC_LPAd_ES11_InitiateAuthentication_Nominal where the LPAd play the role of IPAd.

4.2.27.2.2 TC_IPAd_ES11_InitiateAuthentication_ErrorCases

Test Sequence #01 Error: Invalid SM-DS Address

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Error: Invalid SM-DS Address* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #02 Error: Unsupported Security Configuration

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Error: Unsupported Security Configuration* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #03 Error: Unsupported SVN

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Error: Unsupported SVN* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #04 Error: Unavailable SM-DS Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #04 Error: Unavailable SM-DS Certificate* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #05 Error: Invalid SM-DS Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #05 Error: Invalid SM-DS Certificate* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #06 Error: Invalid SM-DS Signature

This test sequence is the same as SGP.23 [32] - the *Test Sequence #06 Error: Invalid SM-DS Signature* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #07 Error: Invalid SM-DS Address sent by the SM-DS* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #08 Error: Unsupported CI Key ID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #08 Error: Unsupported CI Key ID* defined in section 4.4.27.2.2 TC_LPAd_ES11_InitiateAuthentication_ErrorCases where the LPAd play the role of IPAd.

4.2.28 ES11 (IPA – SM-DS): AuthenticateClient

4.2.28.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This interface is identical to the ES11 interface defined in section 5.8 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.28.2 Test Cases

4.2.28.2.1 TC_IPAd_ES11_AuthenticateClient_Nominal

Test Sequence #01 Nominal: Authenticate Client with empty MatchingID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: Authenticate Client with empty MatchingID* defined in section 4.4.28.2.1 TC_LPAd_ES11_AuthenticateClient_Nominal where the LPAd play the role of IPAd.

Test Sequence #02 Nominal: Authenticate Client with MatchingID set to EventID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Nominal: Authenticate Client with MatchingID set to EventID* defined in section 4.4.28.2.1 TC_LPAd_ES11_AuthenticateClient_Nominal where the LPAd play the role of IPAd.

4.2.28.2.2 TC_IPAd_ES11_AuthenticateClient_ErrorCases

Test Sequence #01 Error: Invalid EUM Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Error: Invalid EUM Certificate set to EventID* defined in section 4.4.28.2.2 TC_LPAd_ES11_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #02 Error: Expired EUM Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Error: Expired EUM Certificate* defined in section 4.4.28.2.2 TC_LPAd_ES11_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #03 Error: Invalid eUICC Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #03 Error: Invalid eUICC Certificate* defined in section 4.4.28.2.2 TC_LPAd_ES11_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #04 Error: Expired eUICC Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #04 Error: Expired eUICC Certificate* defined in section 4.4.28.2.2 TC_LPAd_ES11_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #05 Error: Invalid eUICC signature or serverChallenge

This test sequence is the same as SGP.23 [32] - the *Test Sequence #05 Error: Invalid eUICC signature or serverChallenge* defined in section 4.4.28.2.2 TC_LPAd_ES11_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #06 Error: Unknown TransactionID

This test sequence is the same as SGP.23 [32] - the *Test Sequence #06 Error: Unknown TransactionID* defined in section 4.4.28.2.2 TC_LPAd_ES11_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

Test Sequence #07 Error: Unknown Event Record

This test sequence is the same as SGP.23 [32] - the *Test Sequence #07 Error: Unknown Event Record* defined in section 4.4.28.2.2 TC_LPAd_ES11_AuthenticateClient_ErrorCases where the LPAd play the role of IPAd.

4.2.29 ES11 (IPA -- SM-DS): HTTPS

4.2.29.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This interface is identical to the ES11 interface defined in section 5.8 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.29.2 Test Cases

4.2.29.2.1 TC_IPAd_ES11_HTTPS_Nominal

Test Sequence #01 Nominal: HTTPS Session Establishment

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Nominal: HTTPS Session Establishment* defined in section 4.4.29.2.1TC_LPAd_ES11_HTTPS_Nominal where the LPAd play the role of IPAd.

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.

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Nominal: non-reuse of session keys* defined in section 4.4.29.2.1 TC_LPAd_ES11_HTTPS_Nominal where the LPAd play the role of IPAd.

4.2.29.2.2 TC_IPAd_ES11_HTTPS_Error

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

This test sequence is the same as SGP.23 [32] - the *Test Sequence #01 Error: Invalid (SM-DS) TLS Certificate signature* defined in section 4.4.29.2.2 TC_LPAd_ES11_HTTPS_Error where the LPAd play the role of IPAd.

Test Sequence #02 Error: Expired TLS Certificate

This test sequence is the same as SGP.23 [32] - the *Test Sequence #02 Error: Expired TLS Certificate* defined in section 4.4.29.2.2 TC_LPAd_ES11_HTTPS_Error where the LPAd play the role of IPAd.

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)

This test sequence is the same as SGP.23 [32] - the *Test Sequence #07 Error: Invalid TLS Certificate based on Invalid CI (Invalid Curve)* defined in section 4.4.29.2.2 TC_LPAd_ES11_HTTPS_Error where the LPAd play the role of IPAd.

4.2.29 ES11 (IPA -- SM-DS): HTTPS

4.2.29.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This interface is identical to the ES11 interface defined in section 5.8 of SGP.22 [2], where the IPA plays the role of LPA.

4.2.30 ESipa (IPA -- EIM): InitiateAuthentication

4.2.30.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This function requests the SM-DP+/SM-DS authentication via the eIM.

According to SGP.32 [31], the error codes returned by ESipa.InitiateAuthentication SHALL be the same as those of ES9+'.InitiateAuthentication / ES11'.InitiateAuthentication with the following additions:

- smdpAddressMismatch – indicates an error when matching SM-DP+/SM-DS Address sent in ES9+'.InitiateAuthentication with / ES11'.InitiateAuthentication SM-DP+/SM-DS Address received from the SM-DP+/SM-DS,
- smdpOidMismatch – indicates an error when matching SM-DP+ OID from AC with SM-DP+ OID from SM-DP+ Certificate

4.2.30.2 Test Cases

4.2.30.2.1 TC_IPAd_ESipa_ InitiateAuthentication_Nominal#

The test sequences for this section are FFS **4.2.30.2.2 TC_IPAd_ESipa_ InitiateAuthentication_ErrorCases**

The test sequences for this section are FFS.

4.2.31 ESipa (IPA -- EIM): GetBoundProfilePackage

4.2.31.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This function requests the delivery and the binding of a Profile Package for the eUICC.

4.2.31.2 Test Cases

4.2.31.2.1 TC_IPAd_ESipa_GetBoundProfilePackage_Nominal

The test sequences for this section are FFS.

4.2.31.2.2 TC_IPAd_ESipa_GetBoundProfilePackage _ErrorCases

The test sequences for this section are FFS.

4.2.32 ESipa (IPA -- EIM): AuthenticateClient

4.2.32.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This function SHALL be called by the IPA to request the authentication of the eUICC by the SM-DP+/SM-DS

4.2.32.2 Test Cases

4.2.32.2.1 TC_IPAd_ESipa_AuthenticateClient _Nominal

The test sequences for this section are FFS.

4.2.32.2.2 TC_IPAd_ESipa_ AuthenticateClient _ErrorCases

The test sequences for this section are FFS.

4.2.33 ESipa (IPA -- EIM): TransferEimPackage

4.2.33.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This function is used by the eIM to transfer single eIM Package to the IPA

4.2.33.2 Test Cases

4.2.33.2.1 TC_IPAd_ESipa_TransferEimPackage_Nominal

The test sequences for this section are FFS.

4.2.33.2.2 TC_IPAd_ESipa_TransferEimPackage_ErrorCases

The test sequences for this section are FFS.

4.2.34 ESipa (IPA -- EIM): GetEIMPackage

4.2.34.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This function is used by the IPA to retrieve an eIM Package.

4.2.34.2 Test Cases

4.2.34.2.1 TC_IPAd_ESipa_GetEIMPackage_Nominal

The test sequences for this section are FFS.

4.2.34.2.2 TC_IPAd_ESipa_GetEIMPackage_ErrorCases

The test sequences for this section are FFS.

4.2.35 ESipa (IPA -- EIM): ProvideEimPackageResult

This function is used by the IPA to retrieve an eIM Package.

This function is used by the IPA to deliver an eIM Package Result optionally including one or more Notifications to the eIM in the same function call.

4.2.35.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

4.2.35.2 Test Cases

4.2.35.2.1 TC_IPAd_ESipa_ ProvideEimPackageResult_Nominal

The test sequences for this section are FFS.

4.2.35.2.2 TC_IPAd_ESipa_ ProvideEimPackageResult_ErrorCases

The test sequences for this section are FFS.

4.2.36 ESipa (IPA -- EIM): HandleNotification

4.2.36.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This function SHALL be called by the IPA to notify the eIM and/or SM-DP+ that a Profile has been successfully installed on the eUICC or that a profile has been successfully enabled, disabled, or deleted on the eUICC

4.2.36.2 Test Cases

4.2.36.2.1 TC_IPAd_ESipa_HandleNotification_Nominal

The test sequences for this section are FFS.

4.2.36.2.2 TC_IPAd_ESipa_HandleNotification_ErrorCases

The test sequences for this section are FFS.

4.2.37 ESipa (IPA -- EIM): CancelSession

4.2.37.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

This function SHALL be called by the eIM to request the cancellation of an on-going RSP session.

4.2.37.2 Test Cases

4.2.37.2.1 TC_IPAd_ESipa_CancelSession_Nominal

The test sequences for this section are FFS.

4.2.37.2.2 TC_IPAd_ESipa_CancelSession_ErrorCases

The test sequences for this section are FFS.

4.3 TLS Interface

4.3.1 TLS, Server Authentication, TLS Establishment

4.3.1.1 Conformance Requirements

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

4.3.1.2 Test Cases

4.3.1.2.1 TC_Server_Authentication_for_HTTPS_EstablishmentNIST

The test sequences for this section are FFS.

4.3.1.2.2 TC_Server_Authentication_for_HTTPS_EstablishmentBRP

The test sequences for this section are FFS.

5 Procedure - Behaviour Testing

5.1 General Overview

5.2 VOID

5.3 VOID

5.4 Device Procedures

5.4.1 Profile Download

5.4.1.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

5.4.1.2 Test Cases

5.4.1.2.1 TC_IPAd_DirectProfileDownload_IPA_initiated_with_Activation_Code

Test Sequence #01 Nominal: Add a new Operational Profile initiated by IPA by using Activation 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_1 (PROFILE_OPERATIONAL1).
S_eIM	#ACTIVATION_CODE_1 is available on S_eIM
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC with #ACTIVATION_CODE_1 as <ACTIVATION_CODE>		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
3	PROC_ES9+_INIT_AUTH		
4	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>		
5	PROC_ES9+_GET_BPP		
6	PROC_ES9+_HANDLE_NOTIF See NOTE1		

7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
8	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE1
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
NOTE1: The Notification and eIM Package Result (steps 6 and 8) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

Test Sequence #02 Nominal: Add a new Operational Profile initiated by IPA by using Activation Code 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.
S_eIM	#ACTIVATION_CODE_3 is available on S_eIM
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC with #ACTIVATION_CODE_3 as <ACTIVATION_CODE>		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
3	PROC_ES9+_INIT_AUTH		
4	PROC_ES9+_AUTH_CLIENT_CC with #MATCHING_ID_3 as <MATCHING_ID>		
5	CONFIRMATION_CODE1 is provided to the IPA See NOTE1		

6	PROC_ES9+_GET_BPP_CC
7	PROC_ES9+_HANDLE_NOTIF See NOTE2
8	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE3
9	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE2
10	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE3
IF O_D_ESIPA_HANDLE_NOTIF	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
12	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
NOTE1: the method to provide confirmation code to IPAd is IPAd dependant NOTE2: The Notification and eIM Package Result (steps 7 and 9) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE3: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.1.2.2 TC_IPAd_DirectProfileDownload_IPA_initiated_ActivationCode_InvalidFormat

Test Sequence #01 Error: Add a new Operational Profile initiated by eIM by using wrongly formatted Activation Code

Initial Conditions	
Entity	**Description of the initial condition**
S_eIM	#ACTIVATION_CODE_INVALID_FORMAT is available on S_eIM
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPAd is triggered to send ESipa.GetEimPackage method See NOTE		

1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_PROFILE_DOWNLOAD_TRIGGER_AC_OK with #ACTIVATION_CODE_INVALID_FORMAT as <ACTIVATION_CODE>)	
3	IPAd → S_EIM+	Send ESipa.HandleNotification method with eIMPackageResultResponseError	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPRRE_INVALID_PACKAGE_FORMAT))
4	S_EIM → IPAd	#R_HTTP_204_OK	IPAd does not send TLS Client Hello to S_SM-DP+ within IUT_IPAd_TLS_INIT
NOTE: It is IPAd dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

5.4.1.2.3 TC_IPAd_DirectProfileDownload_IPA_initiated_with_ConfirmationCode_smdpSigned2

Test Sequence #01 Nominal: Add a new Operational Profile initiated by IPA by using Activation Code 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.
S_eIM	#ACTIVATION_CODE_1 is available on S_eIM
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC with #ACTIVATION_CODE_1 as <ACTIVATION_CODE>		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
3	PROC_ES9+_INIT_AUTH		
4	PROC_ES9+_AUTH_CLIENT_CC with #MATCHING_ID_1 as <MATCHING_ID>		

5	CONFIRMATION_CODE1 is provided to the IPA See NOTE1
6	PROC_ES9+_GET_BPP_CC
7	PROC_ES9+_HANDLE_NOTIF See NOTE2
8	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE3
9	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE2
10	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE3
IF O_D_ESIPA_HANDLE_NOTIF	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
12	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
NOTE1: the method to provide confirmation code to IPAd is IPAd dependant NOTE2: The Notification and eIM Package Result (steps 7 and 9) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE:3 This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.1.2.4 TC_IPAd_DirectProfileDownload_IPA_initiated_default SM-DP+

Test Sequence #01 Nominal: DirectProfileDownload from 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.
S_eIM	No secure connection is established between S_eIM and IPAd.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_DEFAULT_SM-DP+		
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
6	PROC_ES9+_HANDLE_NOTIF See NOTE1
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
8	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
NOTE1: The Notification and eIM Package Result (steps 6 and 8) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.1.2.5 TC_IPAd_DirectProfileDownload_IPA_initiated_with_PPRs

Test Sequence #01 Nominal: DirectProfileDownload with PPR1

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 not required for #MCC_MNC4 with gid1 and gid2 absent.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_4 (associated with PROFILE_OPERATIONAL4).
S_eIM	#ACTIVATION_CODE_4 is available on S_eIM
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC with #ACTIVATION_CODE_4 as <ACTIVATION_CODE>		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		

3	PROC_ES9+_INIT_AUTH
4	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_4 as <MATCHING_ID>
5	PROC_ES9+_GET_BPP
6	PROC_ES9+_HANDLE_NOTIF See NOTE1
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
8	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE1
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_4> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_4> as <PROFILE_INFO>
ENDIF	
NOTE1: The Notification and eIM Package Result (steps 6 and 8) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

Test Sequence #02 Nominal: DirectProfileDownload with PPR2

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 not required for #MCC_MNC2 with gid1 and gid2 absent.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_3 (associated with PROFILE_OPERATIONAL3).
S_eIM	#ACTIVATION_CODE_3_NO_CC is available on S_eIM
S_eIM	No secure connection is established between S_eIM and IPAd

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC with #ACTIVATION_CODE_3_NO_CC as <ACTIVATION_CODE>		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+		
3	PROC_ES9+_INIT_AUTH		

4	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_3 as <MATCHING_ID>
5	PROC_ES9+_GET_BPP
6	PROC_ES9+_HANDLE_NOTIF See NOTE1
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
8	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE1
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_3> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_3> as <PROFILE_INFO>
ENDIF	
NOTE1: The Notification and eIM Package Result (steps 6 and 8) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

Test Sequence #03 Nominal: DirectProfileDownload when 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 and End User Consent is not required for #MCC_MNC4 with gid1 and gid2 absent.
eUICC	The PROFILE_OPERATIONAL4 with PPR1 is installed and enabled on the eUICC.
S_SM-DP+	There is a pending Profile download order for #MATCHING_ID_1 (associated with PROFILE_OPERATIONAL1).
S_eIM	#ACTIVATION_CODE_1 is available on S_eIM
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC with #ACTIVATION_CODE_1 as <ACTIVATION_CODE>		

2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+
3	PROC_ES9+_INIT_AUTH
4	PROC_ES9+_AUTH_CLIENT with #MATCHING_ID_1 as <MATCHING_ID>
5	PROC_ES9+_GET_BPP
6	PROC_ES9+_HANDLE_NOTIF See NOTE1
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
8	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE1
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> and <PROFILE_INFO_IOT_4> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> and <PROFILE_INFO_IOT_4> as <PROFILE_INFO>
ENDIF	
NOTE1: The Notification and eIM Package Result (steps 6 and 8) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.1.2.6 TC_IPAd_DirectProfileDownload_IPA_initiated_with_Empty MatchingID

Test Sequence #01 Nominal: Add a new Operational Profile initiated by IPA by using Activation Code with empty MatchingID

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.
S_eIM	No secure connection is established between S_eIM and IPAd.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC with #ACTIVATION_CODE_5 as <ACTIVATION_CODE>		

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>
5	PROC_ES9+_GET_BPP
6	PROC_ES9+_HANDLE_NOTIF See NOTE1
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
8	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE1
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
NOTE1: The Notification and eIM Package Result (steps 6 and 8) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.1.2.7 TC_IPAd_DirectProfileDownload_IPA_initiated_default SM-DP+_immediate_profile_enabling

Test Sequence #01 Nominal: DirectProfileDownload from the default SM-DP+ address with immediate profile enabling

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.
S_eIM	No secure connection is established between S_eIM and IPAd.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		

IC2	PROC_ESIPA_GET_EIM_PACKAGE_CONFIGURE_IMMEDIATE_ENABLE		
IF O_D_ESIPA_HANDLE_NOTIF			
IC3	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_CIER		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
IC4	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_CIER		
ENDIF			
IC4	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
1	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_DEFAULT_SM-DP+		
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		
6	PROC_ES9+_HANDLE_NOTIF See NOTE2		
7	IPAd → 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_IPAd_NOTIFICATION_TIMEOUT Verify the euiccNotificationSignature <TBS_EUICC_NOTIF_SIG> using the #PK_EUICC_ECDSA S_SM-DP+ SHALL return #R_HTTP_204_OK
8	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
9	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR See NOTE2		
10	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_2> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
12	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_2> as <PROFILE_INFO>		
ENDIF			
NOTE1: This procedure needs to be run only if the TLS connection is not initialized on ESipa. NOTE2: The Notification and eIM Package Result (steps 6 and 9) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel.			

5.4.2 VOID

5.4.3 VOID

5.4.4 Local Profile Management - Delete Profile

5.4.4.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

5.4.4.2 Test Cases

5.4.4.2.1 TC_IPAd_DeleteProfile_Disabled_without_PPR_IPA_initiated

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_OPERATIONAL1 is in Disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_DELETE_PROFILE		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ (S_SERVER configured with #TEST_DP_ADDRESS1)		
3	PROC_ES9+_HANDLE_NOTIF_DEL1 See NOTE1		
4	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
5	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_DELPR See NOTE1		
ENDIF			

IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
6	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_DELPR See NOTE1
ENDIF	
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with empty <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with empty <PROFILE_INFO>
ENDIF	
NOTE1: The Notifications (steps 3, 5 and 6) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.4.2.2 TC_IPAd_DeleteProfile_Enabled_without_PPR_IPA_initiated

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

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.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_DELETE_PROFILE		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ (S_SERVER configured with #TEST_DP_ADDRESS1)		
3	PROC_ES9+_HANDLE_NOTIF_DIS1 See NOTE1		
4	PROC_ES9+_HANDLE_NOTIF_DEL1 See NOTE1		
5	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			

6	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_DPR See NOTE1
7	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_DELPR See NOTE1
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
8	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_DPR See NOTE1
9	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_DELPR See NOTE1
ENDIF	
10	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with empty <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
12	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with empty <PROFILE_INFO>
ENDIF	
NOTE1: The Notifications (steps 3, 4, 6, 7, 8 and 9) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.4.2.3 TC_IPAd_DeleteProfile_Error_with_PPR1_IPA_initiated

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
eUICC	The PROFILE_OPERATIONAL4 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_DELETE_PROFILE_4_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_DELPR_ERR_PPR))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_DELPR_ERR_PPR))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_4_EN> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_4_EN> as <PROFILE_INFO>		
ENDIF			
NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.			

5.4.4.2.4 TC_IPAd_DeleteProfile_Error_Disabled_with_PPR2_IPA_initiated

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
eUICC	The PROFILE_OPERATIONAL7 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL7 is in Disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_DELETE_PROFILE_7_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_DELPR_ERR_PPR))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_DELPR_ERR_PPR))

6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_7_DIS> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_7_DIS> as <PROFILE_INFO>		
ENDIF			
NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.			

5.4.4.2.5 TC_IPAd_DeleteProfile_Error_Enabled_with_PPR2_IPA_initiated

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
eUICC	The PROFILE_OPERATIONAL8 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL8 is in Enabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_DELETE_PROFILE_8_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			

3	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_DPR See NOTE3		
4	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result See NOTE3	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_DELPR_ERR_PPR))
5	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
6	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_DPR See NOTE3		
7	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result See NOTE3	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_DELPR_ERR_PPR))
8	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_8_DIS> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_8_DIS> as <PROFILE_INFO>		
ENDIF			
NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa. NOTE3: The Notifications (steps 3, 4, 6, 7) MAY be sent to eIM in any order.			

5.4.5 Profile State Management Operation - Enable Profile

5.4.5.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

5.4.5.2 Test Cases

5.4.5.2.1 TC_IPAd_EnableProfile_IPA_initiated

Test Sequence #01 Nominal: Enable a formerly 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.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_ENABLE_PROFILE		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ (S_SERVER configured with #TEST_DP_ADDRESS1)		
3	PROC_ES9+_HANDLE_NOTIF_EN1 See NOTE1		
4	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
5	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_EPRSee NOTE1		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
6	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_EPR See NOTE1		
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_2> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_2> as <PROFILE_INFO>		
ENDIF			

NOTE1: The Notifications (steps 3, 5 and 6) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel.
NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.

5.4.5.2.2 TC_IPAd_EnableProfile_ImplicitDisable_IPA_initiated

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_OPERATIONAL1 and PROFILE_OPERATIONAL2 are installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
eUICC	The PROFILE_OPERATIONAL2 is in Enabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_ENABLE_PROFILE		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ (S_SERVER configured with #TEST_DP_ADDRESS1)		
3	PROC_ES9+_HANDLE_NOTIF_EN1 See NOTE1		
4	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ (S_SERVER configured with #TEST_DP_ADDRESS2)		
5	PROC_ES9+_HANDLE_NOTIF_DIS2 See NOTE1		
6	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
7	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_EPR See NOTE1		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
8	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_EPR		

	See NOTE1
ENDIF	
9	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_5> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
11	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_5> as <PROFILE_INFO>
ENDIF	
NOTE: The Notifications (steps 3, 5, 7 and 8) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.5.2.3 TC_IPAd_EnableProfile_Error_ProfileAlreadyEnabled_IPA_initiated

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.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))

2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_NO_RB_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_NOT_DIS))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_NOT_DIS))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_2> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_2> as <PROFILE_INFO>		
ENDIF			
NOTE: This procedure needs to be run only if the TLS connection is not initialized on ESipa.			

5.4.5.2.4 TC_IPAd_EnableProfile_Error_PPR1Set_IPA_initiated

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
eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
eUICC	The PROFILE_OPERATIONAL4 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_NO_RB_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_PPR))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_PPR))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE		

IF O_D_ESIPA_HANDLE_NOTIF	
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_6> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_6> as <PROFILE_INFO>
ENDIF	
NOTE: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.5.2.5 TC_LPAd_EnableProfile_Error_RollbackNoEnabledProfile_IPA_initiated

Test Sequence #01 Error: Enable a Profile with Rollback set while no other profile is Enabled

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.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_RB_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_UNKNOWN))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error

ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_UNKNOWN))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>		
ENDIF			
NOTE: This procedure needs to be run only if the TLS connection is not initialized on ESipa.			

5.4.6 Local Profile Management - Disable Profile

5.4.6.1 Conformance Requirements

References

GSMA RSP Technical Specification [2] and GSMA IoT eSIM Technical Specification [31]

5.4.6.2 Test Cases

5.4.6.2.1 TC_IPAd_DisableProfile_IPA_Initiated

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.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_DISABLE_PROFILE		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+ (S_SERVER configured with #TEST_DP_ADDRESS1)		
3	PROC_ES9+_HANDLE_NOTIF_DIS1 See NOTE1		
4	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
5	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_DPRSee NOTE1		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
6	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_DPR See NOTE1		
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>		
ENDIF			
NOTE1: The Notifications (steps 3, 5 and 6) MAY be sent to S_eIM and S_SM-DP+ in any order or in parallel. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.			

5.4.6.2.2 TC_IPAd_DisableProfile_Error_ProfileAlreadyDisabled_IPA_Initiated

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.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_DISABLE_PROFILE_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_DPR_ERR_NOT_EN))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_DPR_ERR_NOT_EN))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

ENDIF	
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2
IF O_D_ESIPA_HANDLE_NOTIF	
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1> as <PROFILE_INFO>
ENDIF	
NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.	

5.4.6.2.3 TC_IPAd_DisableProfile_Error_PPR1Set_IPA_Initiated

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 and End User Consent is not required for #MCC_MNC4 with gid1 and gid2 absent.
eUICC	The PROFILE_OPERATIONAL4 is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL4 is in Enabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_DISABLE_PROFILE_4_TRIGGER_OK)	IPAd does not send any notification to S_SM-DP+
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PA

			CKAGE_RESULT (#R_EPR_DPR_ERR_PPR))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_DPR_ERR_PPR))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_4_EN> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_4_EN> as <PROFILE_INFO>		
ENDIF			
NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure. NOTE2: This procedure needs to be run only if the TLS connection is not initialized on ESipa.			

5.4.7 VOID

5.4.8 VOID

5.4.9 VOID

5.4.10 VOID

5.4.11 VOID

5.4.12 Local Profile Management – Set fallback attribute

5.4.12.1 Conformance Requirements

References

GSMA IoT eSIM Technical Specification [31]

5.4.12.2 Test Cases

5.4.12.2.1 TC_IPAd_SetFallbackAttribute

Test Sequence #01 Nominal: Set Fallback Attribute

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_ALLOWED is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_ALLOWED is in Disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_SET_FALLBACK		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
3	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_SET_FALLBACK		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
4	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_SET_FALLBACK		
ENDIF			
5	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
6	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1_FALLBACK> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
7	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1_FALLBACK> as <PROFILE_INFO>		
ENDIF			
NOTE1: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.			

Test Sequence #02 Nominal: Set Fallback Attribute when already set

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_SET is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_SET is in Disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_SET_FALLBACK		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
3	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_SET_FALLBACK		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
4	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_SET_FALLBACK		
ENDIF			
5	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
6	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1_FALLBACK> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
7	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1_FALLBACK> as <PROFILE_INFO>		
ENDIF			
NOTE1: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.			

Test Sequence #03 Nominal: Set Fallback Attribute with a Disabled Fallback Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL2 with METADATA_OP_PROF2_FALLBACK_SET is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 with METADATA_OP_PROF2_FALLBACK_SET is in disabled state.

eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_ALLOWED is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_ALLOWED is in disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_SET_FALLBACK		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
3	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_SET_FALLBACK		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
4	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_SET_FALLBACK		
ENDIF			
5	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
6	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_7> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
7	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_7> as <PROFILE_INFO>		
ENDIF			
NOTE1: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.			

5.4.12.2.2 TC_IPAd_SetFallbackAttribute_Error_Target_Profile_Not_Available

Test Sequence #01 Error: Target_Profile_Not_Available

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 is not installed on the eUICC.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_SET_FALLBACK_TRIGGER_OK)	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
4	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_ERR_PROFILE_NOT_AVAILABLE))
5	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
6	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_ERR_PROFILE_NOT_AVAILABLE))
7	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure. NOTE2: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.			

5.4.12.2.3 TC_IPAd_SetFallbackAttribute_Error_Not_Allowed

Test Sequence #01 Error: Fallback_Not_allowed

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.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_SET_FALLBACK_TRIGGER_OK)	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
4	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_ERR_PROFILE_NOT_ALLOWED))
5	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
6	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_ERR_PROFILE_NOT_ALLOWED))
7	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

ENDIF	
8	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1
IF O_D_ESIPA_HANDLE_NOTIF	
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1 > as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1 > as <PROFILE_INFO>
ENDIF	
NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure. NOTE2: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.	

5.4.12.2.4 TC_IPAd_SetFallbackAttribute_Error_Fallback_Profile_Already_Enabled

Test Sequence #01 Error: Fallback_Profile_Already_Enabled

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL2 with METADATA_OP_PROF2_FALLBACK_SET is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL2 with METADATA_OP_PROF2_FALLBACK_SET is in Enabled state.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_ALLOWED is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_ALLOWED is in disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IC2	IPA is triggered to send ESipa.GetEimPackage method See NOTE1		
1	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))

2	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_SET_FALLBACK_TRIGGER_OK)	No error
3	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE2		
IF O_D_ESIPA_HANDLE_NOTIF			
4	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_ERR_FALLBACK_PROFILE_ENABLED))
5	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
6	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_ERR_FALLBACK_PROFILE_ENABLED))
7	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
8	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_8> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
10	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_8> as <PROFILE_INFO>		
ENDIF			

NOTE1: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.

NOTE2: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.

5.4.13 Local Profile Management – Unset fallback attribute

5.4.13.1 Conformance Requirements

References

GSMA IoT eSIM Technical Specification [31]

5.4.13.2 Test Cases

5.4.13.2.1 TC_IPAd_UnsetFallbackAttribute

Test Sequence #01 Nominal: Unset_Fallback_Attribute

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_SET is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_SET is in Disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_UNSET_FALLBACK		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
3	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_UNSET_FALLBACK		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
4	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_UNSET_FALLBACK		
ENDIF			
5	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
6	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1_FALLBACK_ALLOWED> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
7	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1_FALLBACK_ALLOWED> as <PROFILE_INFO>		

ENDIF
NOTE1: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.

5.4.13.2.2 TC_IPAd_UnsetFallbackAttribute_Error_Fallback_Profile_Enabled

Test Sequence #01 Error: Unset_Fallback_Attribute_Fallback_Profile_Enabled

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_SET is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_SET is in Enabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_UNSET_FALLBACK		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_UNSET_FALLBACK_ERR_FALLBACK_PROFILE_ENABLED))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_UNSET_FALLBACK_ERR_FALLBACK_PROFILE_ENABLED))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		

	See NOTE1
IF O_D_ESIPA_HANDLE_NOTIF	
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1_FALLBACK_ENABLED> as <PROFILE_INFO>
ENDIF	
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT	
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1_FALLBACK_ENABLED> as <PROFILE_INFO>
ENDIF	
NOTE1: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.	

5.4.13.2.3 TC_IPAd_UnsetFallbackAttribute_Error_No_Fallback_Profile

Test Sequence #01 Error: Unset_Fallback_Attribute_No_Fallback_Profile

Initial Conditions	
Entity	**Description of the initial condition**
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_ALLOWED is installed on the eUICC.
eUICC	The PROFILE_OPERATIONAL1 with METADATA_OP_PROF1_FALLBACK_Allowed is in Disabled state.
S_eIM	No secure connection is established between S_eIM and IPAd

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
1	PROC_ESIPA_GET_EIM_PACKAGE_UNSET_FALLBACK		
2	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
3	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_UNSET_FALLBACK_ERR_NO_FALLBACK_PROFILE))
4	S_EIM → IPAd	#R_HTTP_204_OK	No error
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
5	IPAd → S_EIM	Send ESipa.ProvideEimPacka	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1,

		geResult method with eIM Package Result	#PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_UNSET_FALLBACK_ERR_NO_FALLBACK_PROFILE))
6	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
ENDIF			
7	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA See NOTE1		
IF O_D_ESIPA_HANDLE_NOTIF			
8	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF with <PROFILE_INFO_IOT_1_FALLBACK_ALLOWED> as <PROFILE_INFO>		
ENDIF			
IF O_D_ESIPA_PROVIDE_EIM_PACKAGE_RESULT			
9	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT with <PROFILE_INFO_IOT_1_FALLBACK_ALLOWED> as <PROFILE_INFO>		
ENDIF			
NOTE1: This procedure needs to be ran only if the TLS connection is not initialized on ESipa.			

6 End-to-End Testing

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

Annex A Constants

A.1 Generic Constants

Name	Content
ACTIVATION_CODE_1	1$#TEST_DP_ADDRESS1$#MATCHING_ID_1
ACTIVATION_CODE_3	1$#TEST_DP_ADDRESS1$#MATCHING_ID_3$$1
ACTIVATION_CODE_3_NO_CC	1$#TEST_DP_ADDRESS1$#MATCHING_ID_3
ACTIVATION_CODE_4	1$#TEST_DP_ADDRESS1$#MATCHING_ID_4
ACTIVATION_CODE_5	1$#TEST_DP_ADDRESS1$#MATCHING_ID_EMPTY
ACTIVATION_CODE_INVALID_FORMAT	1#TEST_DP_ADDRESS1$#MATCHING_ID_1
CONFIRMATION_CODE1	0102030405

CTX_PARAMS1_MATCH_ID_DEV_INFO
(CtxParams1)
	ctxParamsForCommonAuthentication : {
	  matchingId <MATCHING_ID>, -- OPTIONAL - see NOTE
	  #DEVICE_INFO
	}
	NOTE: the matchingId field may be present (with value <MATCHING_ID>) or may be absent. The presence or absence of matchingId may be checked in individual test cases.

DEVICE_INFO	deviceInfo {
	  tac ...,
	  deviceCapabilities {
	  ...
	  },
	  imei ... -- Optional
	}--
	Check only that the field is present and has a valid TLV asn.1 structure
	NOTE: The content of deviceInfo is verified in individual test cases.

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
EID1	0x89 04 90 32 12 34 51 23 45 12 34 56 78 90 12 35
EIM_ID	Identical with #TEST_EIM_ADDRESS1

Name	Content
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_PROF7	0x98 92 09 01 87 09 21 43 65 F5
ICCID_OP_PROF8	0x98 92 09 01 98 10 32 54 76 F9
ICON_OP_PROF1	profile_O1.png as defined in Annex H
ICON_OP_PROF2	profile_O2.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_PROF3	0x08 29 99 28 11 32 54 76 96
IMSI_OP_PROF4	0x08 29 99 48 43 65 87 09 21
IMSI_OP_PROF7	0x08 29 99 28 43 65 87 09 21
IMSI_OP_PROF8	0x08 29 99 28 43 65 87 09 21
KEY_LENGTH	0x10
KEY_TYPE	0x88
MATCHING_ID_1	04386-AGYFT-A74Y8-3F815
MATCHING_ID_3	04386-AGYFT-A74Y8-3F817
MATCHING_ID_4	04386-AGYFT-A74Y8-3F818
MCC_MNC1	0x92 F9 18
MCC_MNC2	0x92 F9 28
MCC_MNC4	0x92 F9 48
NAME_OP_PROF1	Operational Profile Name 1
NAME_OP_PROF2	Operational Profile Name 2
NAME_OP_PROF3	Operational Profile Name 3
NAME_OP_PROF4	Operational Profile Name 4

Name	Content
NAME_OP_PROF7	Operational Profile Name 7
NAME_OP_PROF8	Operational Profile Name 8
PATH_AUTH_CLIENT	/gsma/rsp2/es9plus/authenticateClient
PATH_GET_BPP	/gsma/rsp2/es9plus/getBoundProfilePackage
PATH_GET_EIM_PACKAGE	/gsma/rsp2/esipa/getEimPackage
PATH_HANDLE_NOTIF	/gsma/rsp2/es9plus/handleNotification
PATH_HANDLE_NOTIF_IPA	/gsma/rsp2/esipa/handleNotification
PATH_INITIATE_AUTH	/gsma/rsp2/es9plus/initiateAuthentication
PATH_PROVIDE_EIM_PACKAGE_RESULT	/gsma/rsp2/esipa/provideEimPackageResult

PPRS_ALLOWED_EUC_NOT_REQ (ProfilePolicyAuthorisationRule)	{
	pprIds { ppr1, ppr2 },
	allowedOperators {
	{ mccMnc 'EEEEEE'H, gid1 ''H, gid2 ''H}
	},
	pprFlags ''B
	}

REMOTE_OP_ID_INSTALL	1

SIMA_RESULT_OK	simaresp EUICCResponse ::= {
	peStatus {
	{status ok}
	}
	}

SP_NAME1	SP Name 1
SP_NAME2	SP Name 2
SP_NAME3	SP Name 3
SP_NAME4	SP Name 4
SP_NAME7	SP Name 7
SP_NAME8	SP Name 8
S_SM_DP+_OID	2.999.10
S_TLS_CIPHER_SUITE	TLS cipher suite selected as follows: o TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 if present in <TLS_CIPHER_SUITES>, otherwise o TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TEST_DP_ADDRESS1	testsmdpplus1.example.com

Name	Content
TEST_DP_ADDRESS2	testsmdpplus2.example.com
TEST_DP_ADDRESS4	testsmdpplus4.example.com
TEST_EIM_ADDRESS1	testeim1.example.com
TLS_VERSION_1_2	1.2 The minimum TLS Version supported by the Server.
UPP_OP_PROF1	The Unprotected Profile Package related to the PROFILE_OPERATIONAL1 (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).

A.2 Test Certificates and Test Keys

All ECC certificates and keys described below are based on either:

- NIST P-256 curve, defined in Digital Signature Standard [11]

NOTE: SGP.26 [25] contains test keys, valid test certificates and instructions for how to generate invalid certificates. Unless specified differently, the test keys and test certificates used in the present document are bundled with SGP.26 [25].

Name	Description
CERT_CI_ECDSA	Certificate of the CI for its Public ECDSA Key
CERT_EUICC_ECDSA	Certificate of the eUICC for its Public ECDSA key CERT.EUICC.ECDSA in the X.509 format signed by the EUM with SK.EUM.ECDSA
CERT_EUM_ECDSA	Certificate of the EUM for its Public ECDSA key CERT.EUM.ECDSA in the X.509 format signed by the requested CI with SK.CI.ECDSA.
CERT_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_ECDSA	Certificate of the S_SM-DP+ for its Public ECDSA key used for SM-DP+ authentication. This certificate contains the OID #S_SM_DP+_OID.

CERT_S_SM_DPpb_ECDSA	Certificate of the S_SM-DP+ for its Public ECDSA key used for Profile Package Binding. This certificate contains the OID #S_SM_DP+_OID.
PK_CI_ECDSA	Public Key of the CI, contained within #CERT_CI_ECDSA
PK_EUICC_ECDSA	Public Key of the eUICC, contained within #CERT_EUICC_ECDSA
SK_EUICC_ECDSA	Private key of the eUICC for creating signatures
SK_S_EIMsign_ECDSA	Private key of the eIM for creating signatures
SK_S_SM_DPpb_ECDSA	Private key of the S_SM-DP+ used to provide signatures for Profile binding

Annex B Dynamic Content

Variable	Description
ACTIVATION_CODE	An Activation Code value.
ANY_ADD_PP_VERSIONS	Any value of the content of the EUICCInfo2.additionalEuiccProfilePackageVersions field
ANY_PROFILE_VERSION	Any value of type VersionType
ANY_SVN	Any value of type `VersionType`
CLIENT_TLS_EPHEM_KEY	Client's ephemeral key and associated information.
COUNTER_EIM	Integer value coded maximum on two bytes. Incremented each time a test tool generates eUICC Package Request.
EIM_SIGNATURE	The eIM signature 1 (eimSignature) computed using #SK_S_EIMsign_ECDSA across the euiccPackageSigned.
EUICC_SIGN_EPR_EPR	The eUICC signature of the eUICC Package Result containing Enable Profile Result. The input data used to generate the <EUICC_SIGN_EPR_EPR> is the eUICCPackageResultDataSigned TLV.
EIM_TRANSACTION_ID	The TransactionID (Unique Transaction Identifier) generated by the (S_)EIM 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).
EUICC_CHALLENGE	Random eUICC challenge, coded as asn.1 OCTET STRING, 16 bytes.
EUICC_CI_PK_ID_LIST_FOR_SIGNING	List of CI Public Key Identifiers supported on the eUICC for signature creation, coded as ASN.1 sequence of SubjectKeyIdentifier. The CI Public Key Identifier is from the list of possible CI Public Key Identifier. This possible CI Public Key Identifiers as supported by the eUICC will be defined later on.
EUICC_CI_PK_ID_LIST_FOR_VERIFICATION	List of CI Public Key Identifiers supported on the eUICC for signature verification, coded as ASN.1 sequence of SubjectKeyIdentifier. The CI Public Key Identifier is from the list of possible CI Public Key Identifier. This possible CI Public Key Identifiers as supported by the eUICC will be defined later on.
EUICC_CI_PK_ID_TO_BE_USED	CI Public Key Identifier to be used by the eUICC for signature, coded as ASN.1 sequence of SubjectKeyIdentifier.
EUICC_RSP_CAPABILITY	RspCapability of the eUICC, coded as ASN.1 BIT STRING
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_SIGNATURE2	The eUICC signature 2 (euiccSignature2) computed using the #SK_EUICC_ECDSA across the following data objects:

Variable	Description
	• euiccSigned2 • smdpSignature2 present in the PrepareDownloadRequest structure
EUICC_SIGN_EPR_CIER	The eUICC signature of the eUICC Package Result containing Configure Immediate Enable Result. The input data used to generate the <EUICC_SIGN_EPR_CIER > is the eUICCPackageResultDataSigned TLV.
EUICC_SIGN_EPR_DPR	The eUICC signature of the eUICC Package Result containing Disable Profile Result. The input data used to generate the <EUICC_SIGN_EPR_DPR> is the eUICCPackageResultDataSigned TLV.
EUICC_SIGN_EPR_DELPR	The eUICC signature of the eUICC Package Result containing Delete Profile Result. The input data used to generate the <EUICC_SIGN_EPR_DELPR> is the eUICCPackageResultDataSigned TLV.
EUICC_SIGN_EPR_LPIR	The eUICC signature of the eUICC Package Result containing List Profile Info Result. The input data used to generate the <EUICC_SIGN_EPR_LPIR > is the eUICCPackageResultDataSigned TLV.
EUICC_SIGN_EPR_SET_FALLBACK	The eUICC signature of the eUICC Package Result containing Set Fallback Attribute Result. The input data used to generate the <EUICC_SIGN_EPR_SET_FALLBACK> is the eUICCPackageResultDataSigned TLV.
EUICC_SIGN_EPR_UNSET_FALLBACK	The eUICC signature of the eUICC Package Result containing Unset Fallback Attribute Result. The input data used to generate the <EUICC_SIGN_EPR_UNSET_FALLBACK> is the eUICCPackageResultDataSigned TLV.
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.
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).
IOT_VERSION	The value of the iotVersion field in EUICCInfo2.
IPA_MODE	The value of the ipaMode field in EUICCInfo2.
ISD_P_AID	The ISD-P AID newly created in the eUICC. This AID value is in the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 FF FF 00. Last byte is set to '00' as defined in SGP.02[1].
ISD_P_AID1	The ISD-P AID created in the eUICC for the PROFILE_OPERATIONAL1. This AID value belongs to the range from 0xA0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 to 0xA0 00 00

Variable	Description
	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_AID7	The ISD-P AID created in the eUICC for the PROFILE_OPERATIONAL7. 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_AID8	The ISD-P AID created in the eUICC for the PROFILE_OPERATIONAL7. 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].
L	Exact length of the corresponding tag or of the remaining data.
MATCHING_ID	Unique identifier as defined in [2]. The content can be either empty, or the value of the EventID, or the value of the Activation Code token.
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),
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_S_SM_DP+_ECKA	One-time Public Key generated by the S_SM-DP+ for ECKA. Depending on the eUICC configuration, this key is based on NIST P-256, brainpoolP256r1 or FRP256V1.
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.

Variable	Description
PPR_IDS	Forbidden Profile Policy Rules. This PPR list MAY be empty or MAY contain either PPR1 or PPR2 or both.
PROFILE_INFO	ProfileInfo structure(s)

PROFILE_INFO_IOT_1
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF1,
	isdpAid <ISD_P_AID1>,
	profileState disabled,
	serviceProviderName #SP_NAME1,
	profileName #NAME_OP_PROF1,
	profileClass operational
	}

PROFILE_INFO_IOT_1_FALLBACK
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF1,
	isdpAid <ISD_P_AID1>,
	profileState disabled,
	serviceProviderName #SP_NAME1,
	profileName #NAME_OP_PROF1,
	profileClass operational,
	fallbackAttribute TRUE,
	fallbackAllowed TRUE
	}

PROFILE_INFO_IOT_1_FALLBACK_ALLOWED
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF1,
	isdpAid <ISD_P_AID1>,
	profileState disabled,
	serviceProviderName #SP_NAME1,
	profileName #NAME_OP_PROF1,
	profileClass operational,
	fallbackAttribute FALSE,
	fallbackAllowed TRUE
	}

PROFILE_INFO_IOT_1_FALLBACK_ENABLED
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF1,
	isdpAid <ISD_P_AID1>,
	profileState enabled,
	serviceProviderName #SP_NAME1,
	profileName #NAME_OP_PROF1,
	profileClass operational,
	fallbackAttribute TRUE,
	fallbackAllowed TRUE
	}

PROFILE_INFO_IOT_2
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF1,
	isdpAid <ISD_P_AID1>,
	profileState enabled,
	serviceProviderName #SP_NAME1,
	profileName #NAME_OP_PROF1,
	profileClass operational
	}

Variable	Description

PROFILE_INFO_IOT_3
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF3,
	isdpAid <ISD_P_AID3>,
	profileState disabled,
	serviceProviderName #SP_NAME3,
	profileName #NAME_OP_PROF3,
	profileClass operational
	}

PROFILE_INFO_IOT_4
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF4,
	isdpAid <ISD_P_AID4>,
	profileState disabled,
	serviceProviderName #SP_NAME4,
	profileName #NAME_OP_PROF4,
	profileClass operational
	}

PROFILE_INFO_IOT_4_EN
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF4,
	isdpAid <ISD_P_AID4>,
	profileState enabled,
	serviceProviderName #SP_NAME4,
	profileName #NAME_OP_PROF4,
	profileClass operational
	}

PROFILE_INFO_IOT_7
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF2,
	isdpAid <ISD_P_AID2>,
	profileState enabled,
	serviceProviderName #SP_NAME2,
	profileName #NAME_OP_PROF2,
	profileClass operational,
	fallbackAttribute FALSE,
	fallbackAllowed TRUE
	},
	{
	iccid #ICCID_OP_PROF1,
	isdpAid <ISD_P_AID1>,
	profileState disabled,
	serviceProviderName #SP_NAME1,
	profileName #NAME_OP_PROF1,
	profileClass operational,
	fallbackAttribute TRUE,
	fallbackAllowed TRUE
	}

Variable	Description

PROFILE_INFO_IOT_7_DIS
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF7,
	isdpAid <ISD_P_AID7>,
	profileState enabled,
	serviceProviderName #SP_NAME7,
	profileName #NAME_OP_PROF7,
	profileClass operational
	}

PROFILE_INFO_IOT_8
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF2,
	isdpAid <ISD_P_AID2>,
	profileState enabled,
	serviceProviderName #SP_NAME2,
	profileName #NAME_OP_PROF2,
	profileClass operational,
	fallbackAttribute TRUE,
	fallbackAllowed TRUE
	},
	{
	iccid #ICCID_OP_PROF1,
	isdpAid <ISD_P_AID1>,
	profileState disabled,
	serviceProviderName #SP_NAME1,
	profileName #NAME_OP_PROF1,
	profileClass operational,
	fallbackAttribute FALSE,
	fallbackAllowed TRUE
	}

PROFILE_INFO_IOT_8_DIS
(ProfileInfo)
	{
	iccid #ICCID_OP_PROF8,
	isdpAid <ISD_P_AID8>,
	profileState enabled,
	serviceProviderName #SP_NAME8,
	profileName #NAME_OP_PROF8,
	profileClass operational
	}

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].
SEQ_NUMBER	Sequence Number related to a Notification Metadata generated by the eUICC.
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 IPA. When generated by the S_IPAd, the S_IPAd SHALL use #CONFIRMATION_CODE1 in the calculation unless otherwise specified.

Variable	Description
S_MAC	SCP03T MACing Session key (128 bits length) resulting from the key agreement with eUICC.
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_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_SIGNATURE1	The ASN.1 OCTET STRING encoded SM-DP+ signature (field serverSignature1) computed using the private key related to the server certificate (field serverCertificate) present in the AuthenticateServerRequest structure.

S_SMDP_SIGNED1
(ServerSigned1)
	{
	transactionId <S_TRANSACTION_ID>,
	euiccChallenge <EUICC_CHALLENGE>,
	serverAddress #TEST_DP_ADDRESS1,
	serverChallenge <S_SMDP_CHALLENGE>
	}

S_SM_DP+_SIGN	The S_SM-DP+ signature (smdpSign), computed using the #SK_S_SM_DPpb_ECDSA across the following data objects: • remoteOpId • transactionId • controlRefTemplate • smdpOtpk • euiccOtpk, as provided earlier in the prepareDownloadResponse data object
S_SM_DP+_SIGNATURE2	The ASN.1 OCTET STRING encoded SM-DP+ signature 2 (field smdpSignature2) computed using the private key related to the server certificate (field smdpCertificate) present in the PrepareDownloadRequest structure. This signature SHALL be generated across the following data objects: • smdpSignature2 • euiccSignature1 present in the AuthenticateServerResponse structure
S_TRANSACTION_ID	The TransactionID (Unique Transaction Identifier) generated by the (S_)SM-DP+, or (S_)SM-DS , or (S_)EIM 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).
SEQ_NUMBER	Sequence Number related to a Notification Metadata generated by the eUICC.
SESSION_ID_CLIENT	Random or empty value of the TLS session_id in ClientHello.
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 IPAd or the Client (SM-DP+ or SM-DS) under test.

Variable	Description
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.

Annex C Methods And Procedures

This section describes methods and procedures used in the interfaces compliance test cases. They are part of test cases and SHALL not be executed in standalone mode.

C.1 Methods

If the method is used in the “expected result” column, all parameters SHALL be verified by the simulated entity (test tool). If the method is used in the “Sequence / Description” column, the command SHALL be generated by the simulated entity.

Method	MTD_AUTHENTICATE_CLIENT
Description	Generates or verifies the JSON formatted AuthenticateClient request
Parameter(s)	• paramTransactionId: random 16 byte identifier encoded as String Hexadecimal. • paramAuthenticateServerResponse: server authentication response structured as ASN.1 encoded as base 64.

Details	JSON body
	{
	"transactionId" : paramTransactionId,
	"authenticateServerResponse" : paramAuthenticateServerResponse
	}

Method	MTD_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_GET_EIM_PACKAGE
Description	Generates or verifies the JSON formatted GetEimPackage request
Parameter(s)	• paramEidValue: EID as described in SGP.22

Details	JSON body
	{
	"eidValue" : paramEidValue,
	}

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_HANDLE_NOTIF_EIM_PACKAGE_RESULT
Description	Generates or verifies the JSON formatted HandleNotification request with provideEimPackageResult
Parameter(s)	paramProvideEimPackageResult: ProvideEimPackageResult data object

Details	JSON body
	{
	"provideEimPackageResult" : paramProvideEimPackageResult
	}

Method	MTD_HTTP_REQ
Description	Sends or verifies a secured HTTP request message delivering a JSON object payload using a network to an off-card entity.
Parameter(s)	• paramServerAddress: Target Server address • paramFunctionPath: Function path • paramRequestMessage: JSON Request message
Details	HTTP POST paramFunctionPath HTTP/1.1 Host: paramServerAddress User-Agent: See NOTE 1 X-Admin-Protocol:gsma/rsp/v#RSP_SVN Content-Type:application/json OR application/json;charset=UTF-8 (see NOTE 2) Content-Length: <L> paramRequestMessage NOTE 1: If the request is sent by the IPAd, the User-Agent SHALL be gsma-rsp-ipad. The "User-Agent" field may contain additional information after a semicolon. Otherwise the value of User-Agent is not specified by the current document. The additional information shall not be checked. NOTE 2: the Content-Type checking is relaxed in this specification, in order to allow for common internet usage of “charset=UTF-8” and for compatibility with SGP.22 v3.0. If the request is sent by the entity under test, both values are acceptable (where linear white space as specified in RFC 2616 is allowed after the semi-colon). Further, all parts of these allowed Content-Type value SHALL be checked in a case-insensitive manner, as per RFC 2616.

	If the request is sent by a simulator, application/json shall be used.The HTTP POST request may contain additional header fields. These shall not be checked.

Method	MTD_HTTP_REQ_ESIPA
Description	Sends or verifies a secured HTTP request message delivering a JSON object payload using a network to eIM.
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/v2.1.0 Content-Type: application/json;charset=UTF-8 Content-Length: <L> paramRequestMessage NOTE 1: The value of User-Agent is not specified by [31]. It shall not be checked.

Method	MTD_HTTP_RESP
Description	Sends or verifies a secured HTTP response message delivering a JSON object payload using a network to an off-card entity.
Parameter(s)	• paramResponseMessage: JSON Response message
Details	HTTP/1.1 200 (OK) X-Admin-Protocol: gsma/rsp/v#RSP_SVN Content-Type: application/json OR application/json;charset=UTF-8 (see NOTE) Content-Length: <L> paramResponseMessage NOTE: the Content-Type checking is relaxed in this specification, in order to allow for common internet usage of “charset=UTF-8” and for compatibility with SGP.22 v3.0 If the response is sent by the entity under test, both values are acceptable (where linear white space as specified in RFC 2616 is allowed after the semi-colon). Further, all parts of these allowed Content-Type value SHALL be checked in a case-insensitive manner, as per RFC 2616. If the response is sent by a simulator, application/json shall be used. The HTTP response may contain additional header fields. These shall not be checked.

Method	MTD_HTTP_RESP_ESIPA
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/v2.1.0 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.

Details	JSON body
	{
	"euiccChallenge" : paramEUICCChallenge,
	"euiccInfo1" : paramEUICCInfo1,
	"smdpAddress" : paramServerAddress
	}

Method	MTD_PROVIDE_EIM_PACKAGE_RESULT
Description	Generates or verifies the JSON formatted ProvideEimPackageResult request with eimPackageResult
Parameter(s)	• paramEIDValue: EID value of the targeted eUICC • paramEimPackageResult: eimPackageResult data object

Details	JSON body
	{
	"eidValue" : paramEIDValue,
	"eEimPackageResult" : paramEimPackageResult
	}

Method	MTD_TLS_CLIENT_KEY_EXCH_ETC
Description	Finalizes the Transport Layer Security (TLS) handshake in Server authentication mode on ES9+, or ES11 (Client side).

Parameter(s)	• paramClientKeyExchange: ClientKeyExchange message
Details	Sends the session key information in TLS ClientKeyExchange message, ChangeCipherSpec and Finished message.

Method	MTD_TLS_CLIENT_HELLO
Description	Sends or checks the Client Hello message used to initiate the Transport Layer Security (TLS) handshake in Server authentication or Mutual authentication mode on ES9+, ES11, ES12 or ES15.
Parameter(s)	• paramTLSversion: TLS protocol version • paramAlgs: cipher suite types supported • paramSessionID: Session ID • paramExts: Extensions data for “supported_signature_algorithms”, “trusted_ca_keys” or other (optional)
Details	Sends or receives a TLS ClientHello message according to the parameters defined above. In addition the following parameters will be set: • The list of compression algorithms supported by the client is not explicitly defined, but by default it will be set to NULL. • The random of 4 bytes representing time since epoch on client host and 28 random bytes is not explicitly defined but it SHALL be generated by the test tool TLS implementation NOTE: The Supported Elliptic Curves Extension and the Supported Point Formats Extension extensions MAY be sent by the Client.

Method	MTD_TLS_SERVER_HELLO_ETC
Description	Send or Receives to the Client Hello in the Transport Layer Security (TLS) handshake in Server authentication mode on ES9+, or ES11.
Parameter(s)	• paramTLSversion: TLS protocol version • paramAlgs: cipher suite selected • paramSessionID: Session ID • paramCertificate: TLS server certificate for authentication • paramServerTLSEphemeralKey: TLS Server ephemeral key.
Details	Sends or Receives a TLS ServerHello, Server Certificate, ServerKeyExchange and ServerHelloDone message in this order according to the parameters defined above. NOTE 1: The random of 4 bytes representing time since epoch on client host and 28 random bytes is not explicitly defined in the Server Hello message but it SHALL be generated by the Server under test. NOTE 2: If no parameter mentioned paramServerTLSEphemeralKey, the value SHALL be set as defined in [24] for ServerKeyExchange. No verification required.

C.2 Procedures

Procedure	PROC_ES9+_AUTH_CLIENT

	Description	Authenticate Server procedure without Confirmation Code. #R_AUTH_SERVER_MATCH_ID_DEV_INFO is used with the correct MatchingID contained in the profile download trigger (Activation Code content or Empty MatchingID).

Step	Direction	Sequence / Description	Expected result
1	IPAd → 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+ → IPAd	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 profile download trigger (Activation Code content or Empty MatchingID).

Step	Direction	Sequence / Description	Expected result
1	IPAd → 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+ → IPAd	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	IPAd → 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+ → IPAd	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	IPAd → 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+ → IPAd	MTD_HTTP_RESP(#GET_BPP_OK)	No error

	Procedure	PROC_ES9+_HANDLE_NOTIF	
	Description	Handle Notification procedure.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → 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+ → IPAd	#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_DEL1	
	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(#PENDING_NOTIF_DEL1))
2	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error
NOTE 1: Other Notifications MAY be sent within the same HTTPS session.			

	Procedure	PROC_ES9+_HANDLE_NOTIF_DIS1	
	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(#PENDING_NOTIF_DIS1))
2	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error
NOTE 1: Other Notifications MAY be sent within the same HTTPS session.			

	Procedure	PROC_ES9+_INIT_AUTH	
	Description	Initiate Authentication procedure.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → 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+ → IPAd	MTD_HTTP_RESP(#INITIATE_AUTH_OK)	No error

	Procedure	PROC_TLS_INITIALIZATION_SERVER_AUTH		
	Description	Establishes the Transport Layer Security (TLS) v1.2 connection between the Client IPAd and (S_)SERVER using Server authentication mode on ES9+ or ES11.		
Step	**Direction**	**Sequence / Description**	**Expected result**	**REQ**
1	IPAd → 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 → IPAd	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_S_SERVER_TLS)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)	

3	S_SERVER → IPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established	

	Procedure	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA
	Description	Establishes the Transport Layer Security (TLS) v1.2 connection between the Client IPAd and Server (S_EIM) using Server authentication mode on ES ipa with Variant O certificate.

Step	Direction	Sequence / Description	Expected result
1	IPAd is triggered to establish a secure TLS connection with eIM if the TLS connection is not established yet See NOTE		
2	IPAd → S_EIM	Send TLS Client Hello	MTD_TLS_CLIENT_HELLO(#IUT_TLS_VERSION, <TLS_CIPHER_SUITES>, <SESSION_ID_CLIENT>, <EXT_SHA256_ECDSA>)
3	S_EIM → IPAd	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>)
4	S_EIM → IPAd	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established
NOTE: the method to trigger the IPAd is IPAd dependent.			

	Procedure	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_CIER
	Description	Handle Notification procedure between IPAd and S_EIM for eIM Package Result containing Configure Immediate Enable Result.

Step	Direction	Sequence / Description	Expected result
1	IPAd → S_EIM+	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_CIER_OK))
2	S_EIM → IPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_DPR	
	Description	Handle Notification procedure between IPA and S_EIM for eIM Package Result containing Disable Profile Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_DPR_OK))
2	S_EIM → IPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_DELPR	
	Description	Handle Notification procedure between IPA and S_EIM for eIM Package Result containing Delete Profile Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_DELPR_OK))
2	S_EIM → IPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_PDTR	
	Description	Handle Notification procedure between IPAd and S_EIM for eIM Package Result containing Profile Download Trigger Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM+	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_PDTR_OK)) See NOTE
2	S_EIM → IPAd	#R_HTTP_204_OK	No error
NOTE: The values of notificationAddress, iccid and smdpOid used in #R_EPR_PDTR_OK MAY vary depending on the context (ICCID of the downloaded profile, used SM-DP+ address and certificate).			

	Procedure	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_SET_FALLBACK	
	Description	Handle Notification procedure between IPA and S_EIM for eIM Package Result containing Set Fallback Attribute Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_OK))
2	S_EIM → IPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_UNSET_FALLBACK	
	Description	Handle Notification procedure between IPA and S_EIM for eIM Package Result containing Unset Fallback Attribute Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_UNSET_FALLBACK_OK))
2	S_EIM → IPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_CONFIGURE_IMMEDIATE_ENABLE	
	Description	Retrieval of the configureImmediateEnable PSMO on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_EPR_C	No error

		ONF_IMMEDIATE_ENABLE_OK)	
NOTE: It is IPAd dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_DELETE_PROFILE	
	Description	Delete Profile Trigger on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_DELETE_PROFILE_TRIGGER_OK)	No error
NOTE: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_DISABLE_PROFILE	
	Description	Disable Profile Trigger on ESipa interface using eIM Package Retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_DISABLE_PROFILE_TRIGGER_OK)	No error
NOTE: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package Retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_AC	
	Description	Profile Download Trigger with Activation Code on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_PROFILE_DOWNLOAD_TRIGGER_AC_OK)	No error
NOTE: It is IPAd dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_PROFILE_DOWNLOAD_DEFAULT_SM-DP+	
	Description	Profile Download Trigger with default SM-DP+ on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_PROFILE_DOWNLOAD_TRIGGER_DEFAULT_SM-DP+_OK)	No error
NOTE: It is IPAd dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_HANDLE_NOTIF	
	Description	List Profile Info on ESipa interface using eIM Package retrieval. It includes handle notification procedure between IPAd and S_EIM for eIM Package Result containing List Profile Info result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_EPR_LIST_PROFILE_INFO_OK)	No error
4	IPAd → S_EIM+	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_LPIR_OK))
5	S_EIM → IPAd	#R_HTTP_204_OK	No error
NOTE: It is IPAd dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_LIST_PROFILE_EIM_PACKAGE_RESULT	
	Description	List Profile Info on ESipa interface using eIM Package retrieval. It includes provide eIM package result procedure between IPAd and S_EIM for eIM Package Result containing List Profile Info result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_EPR_LIST_PROFILE_INFO_OK)	No error
4	IPAd → S_EIM+	Send ESipa.ProvideEimPackageResult method with List Profile Info Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT,

			MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_LPIR_OK))
5	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error
NOTE: It is IPAd dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_SET_FALLBACK	
	Description	Set Fallback attribute trigger on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_SET_FALLBACK_TRIGGER_OK)	No error
NOTE: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_UNSET_FALLBACK	
	Description	Unset Fallback attribute trigger on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_UNSET_FALLBACK_TRIGGER_OK)	No error
NOTE: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_CIER	
	Description	Provide Eim Package Result procedure between IPA and S_EIM with eUICC Package Result containing Configure Immediate Enable Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM+	Send ESipa.ProvideEimPackageResult method with Configure Immediate Enable Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_CIER_OK))
2	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

	Procedure	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_DPR	
	Description	Provide eIM Package Result procedure between IPA and S_EIM with eUICC Package Result containing Disable Profile Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_DPR_OK))
2	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

	Procedure	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_DELPR	
	Description	Provide Eim Package Result procedure between IPA and S_EIM with eUICC Package Result containing Delete Profile Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_DELPR_OK))

2	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

	Procedure	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_SET_FALLBACK
	Description	Provide Eim Package Result procedure between IPA and S_EIM with eUICC Package Result containing Set Fallback Attribute Result.

Step	Direction	Sequence / Description	Expected result
1	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_SET_FALLBACK_OK))
2	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

	Procedure	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_UNSET_FALLBACK
	Description	Provide Eim Package Result procedure between IPA and S_EIM with eUICC Package Result containing Unset Fallback Attribute Result.

Step	Direction	Sequence / Description	Expected result
1	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_UNSET_FALLBACK_OK))
2	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

	Procedure	PROC_ES9+_HANDLE_NOTIF_EN1	
	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(#PENDING_NOTIF_EN1))
2	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error
NOTE 1: Other Notifications MAY be sent within the same HTTPS session.			

	Procedure	PROC_ES9+_HANDLE_NOTIF_DIS2	
	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(#PENDING_NOTIF_DIS2))
2	S_SM-DP+ → LPAd	#R_HTTP_204_OK	No error
NOTE 1: Other Notifications MAY be sent within the same HTTPS session.			

	Procedure	PROC_ESIPA_HANDLE_NOTIF_EIM_PACKAGE_RESULT_EPR	
	Description	Handle Notification procedure between IPA and S_EIM for eIM Package Result containing Enable Profile Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.HandleNotification method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK))
2	S_EIM → IPAd	#R_HTTP_204_OK	No error

	Procedure	PROC_ESIPA_PROVIDE_EIM_PACKAGE_RESULT_EPR	
	Description	Provide Eim Package Result procedure between IPA and S_EIM with eUICC Package Result containing Enable Profile Result.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPAd → S_EIM	Send ESipa.ProvideEimPackageResult method with eIM Package Result	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_PROVIDE_EIM_PACKAGE_RE

			SULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK))
2	S_EIM → IPAd	MTD_HTTP_RESP_ESIPA (#S_EIM_ACKNOWLEDGEMENT)	No error

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_ENABLE_PROFILE	
	Description	Enable Profile Trigger on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_NO_RB_TRIGGER_OK)	No error
NOTE: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

	Procedure	PROC_ESIPA_GET_EIM_PACKAGE_ENABLE_PROFILE_RB	
	Description	Enable Profile Trigger with Rollback on ESipa interface using eIM Package retrieval.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	IPA is triggered to send ESipa.GetEimPackage method See NOTE		
2	IPAd → S_eIM	Send ESipa.GetEimPackage method	MTD_HTTP_REQ_ESIPA (#TEST_EIM_ADDRESS1, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))
3	S_eIM → IPAd	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_RB_TRIGGER_OK)	No error
NOTE: It is IPA dependent, if there is a need for a separate trigger, or the trigger in PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA is triggering the eIM Package retrieval procedure.			

Annex D Commands And Responses

D.1 ES8+ Requests And Responses

D.1.1 ES8+ Requests

Name	Content

CONF_ISDP_PROF1	req ConfigureISDPRequest ::={
	  dpProprietaryData {
	  dpOid #S_SM_DP+_OID
	  }
	}

METADATA_OP_PROF1	metadataReq StoreMetadataRequest ::= {
	  iccid #ICCID_OP_PROF1,
	  serviceProviderName #SP_NAME1,
	  profileName #NAME_OP_PROF1,
	  profileClass operational,
	  notificationConfigurationInfo {
	  { profileManagementOperation {
	  notificationInstall,
	  notificationEnable,
	  notificationDisable,
	  notificationDelete
	  },
	  notificationAddress
	#TEST_DP_ADDRESS1
	  }
	  },
	  profileOwner {
	  mccMnc #MCC_MNC1
	  }
	}

METADATA_OP_PROF1_FALLBACK_ALLOWED	metadataReq StoreMetadataRequest ::= {
	  iccid #ICCID_OP_PROF1,
	  serviceProviderName #SP_NAME1,
	  profileName #NAME_OP_PROF1,
	  iconType png,
	  icon #ICON_OP_PROF1,
	  profileClass operational,
	  notificationConfigurationInfo {
	  { profileManagementOperation {
	  notificationInstall,
	  notificationEnable,
	  notificationDisable,
	  notificationDelete
	  },
	  notificationAddress
	#TEST_DP_ADDRESS1
	  }
	  },
	  profileOwner {
	  mccMnc #MCC_MNC1
	  },
	  fallbackAllowed TRUE
	}

METADATA_OP_PROF1_FALLBACK_SET	metadataReq StoreMetadataRequest ::= {
	  iccid #ICCID_OP_PROF1,
	  serviceProviderName #SP_NAME1,
	  profileName #NAME_OP_PROF1,
	  iconType png,
	  icon #ICON_OP_PROF1,
	  profileClass operational,
	  notificationConfigurationInfo {
	  { profileManagementOperation {
	  notificationInstall,
	  notificationEnable,
	  notificationDisable,
	  notificationDelete
	  },
	  notificationAddress
	#TEST_DP_ADDRESS1
	  }
	  },
	  profileOwner {
	  mccMnc #MCC_MNC1
	  },
	  fallbackAttribute TRUE,
	  fallbackAllowed TRUE
	}

METADATA_OP_PROF2_FALLBACK_SET	metadataReq StoreMetadataRequest ::= {
	iccid #ICCID_OP_PROF2,
	serviceProviderName #SP_NAME2,
	profileName #NAME_OP_PROF2
	iconType png,
	icon #ICON_OP_PROF2,
	profileClass operational,
	notificationConfigurationInfo {
	{ profileManagementOperation {
	notificationInstall,
	notificationEnable,
	notificationDisable,
	notificationDelete
	},
	notificationAddress #TEST_DP_ADDRESS2
	}
	},
	profileOwner {
	mccMnc #MCC_MNC2
	},
	fallbackAttribute TRUE,
	fallbackAllowed TRUE
	}

METADATA_OP_PROF3	metadataReq StoreMetadataRequest ::= {
	iccid #ICCID_OP_PROF3,
	serviceProviderName #SP_NAME3,
	profileName #NAME_OP_PROF3,
	profileClass operational,
	profileOwner {
	mccMnc #MCC_MNC2
	},
	profilePolicyRules { ppr2 }
	}

METADATA_OP_PROF4	metadataReq StoreMetadataRequest ::= {
	iccid #ICCID_OP_PROF4,
	serviceProviderName #SP_NAME4,
	profileName #NAME_OP_PROF4,
	profileClass operational,
	notificationConfigurationInfo {
	{ profileManagementOperation {
	notificationInstall,
	notificationEnable,
	notificationDisable,
	notificationDelete
	},
	notificationAddress #TEST_DP_ADDRESS4
	}
	},
	profileOwner {
	mccMnc #MCC_MNC4
	},
	profilePolicyRules {
	ppr1
	}
	}

METADATA_OP_PROF7	metadataReq StoreMetadataRequest ::= {
	iccid #ICCID_OP_PROF7,
	serviceProviderName #SP_NAME7,
	profileName #NAME_OP_PROF7,
	iconType png,
	icon #ICON_OP_PROF7,
	profileClass operational,
	notificationConfigurationInfo {
	{ profileManagementOperation {
	notificationInstall,
	notificationEnable,
	notificationDisable,
	notificationDelete
	},
	notificationAddress #TEST_DP_ADDRESS8
	}
	},
	profileOwner {
	mccMnc #MCC_MNC2
	},
	profilePolicyRules {
	ppr2
	}
	}

METADATA_OP_PROF8	metadataReq StoreMetadataRequest ::= {
	iccid #ICCID_OP_PROF8,
	serviceProviderName #SP_NAME8,
	profileName #NAME_OP_PROF8,
	iconType png,
	icon #ICON_OP_PROF8,
	profileClass operational,
	notificationConfigurationInfo {
	{ profileManagementOperation {
	notificationInstall,
	notificationEnable,
	notificationDisable,
	notificationDelete
	},
	notificationAddress
	#TEST_DP_ADDRESS8
	}
	},
	profileOwner {
	mccMnc #MCC_MNC2
	},
	profilePolicyRules {
	ppr2
	}
	}

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>
	}

D.1.2 ES8+ Responses

Name	Content

D.2 ES9+ Requests And Responses

D.2.1 ES9+ Requests

Name	Content
INITIATE_AUTH_OK	{ "header" : {

	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"transactionId" : <S_TRANSACTION_ID>,
	"serverSigned1" : <S_SMDP_SIGNED1>,
	"serverSignature1" :
	<S_SMDP_SIGNATURE1>,
	"euiccCiPKIdTobeUsed" :
	<EUICC_CI_PK_ID_TO_BE_USED>,
	"serverCertificate" :
	#CERT_S_SM_DPauth_ECDSA
	}
	-- NOTE: select the CI as defined in the note in the chapter 2.1.4 of SGP.23

MATCHING_ID_EMPTY	

PENDING_NOTIF_EN1	response PendingNotification ::=
	otherSignedNotification : {
	tbsOtherNotification {
	seqNumber <SEQ_NUMBER>,
	profileManagementOperation {
	notificationEnable
	},
	notificationAddress
	#TEST_DP_ADDRESS1,
	iccid #ICCID_OP_PROF1
	},
	euiccNotificationSignature
	<TBS_EUICC_NOTIF_SIG>,
	euiccCertificate #CERT_EUICC_ECDSA,
	eumCertificate #CERT_EUM_ECDSA
	}

S_SMDP_SIGNED2	req SmdpSigned2 ::= {
	transactionId <S_TRANSACTION_ID>,
	ccRequiredFlag FALSE
	}

S_SMDP_SIGNED2_CC	req SmdpSigned2 ::= {
	transactionId <S_TRANSACTION_ID>,
	ccRequiredFlag TRUE
	}

D.2.2 ES9+ Responses

Name	Content

AUTH_CLIENT_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},

	"transactionId" : <S_TRANSACTION_ID>, "profileMetadata" : #METADATA_OP_PROF1, "smdpSigned2" : #S_SMDP_SIGNED2, "smdpSignature2" : <S_SM_DP+_SIGNATURE2>, "smdpCertificate" : #CERT_S_SM_DPpb_ECDSA }

AUTH_CLIENT_OK_CC	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"transactionId" : <S_TRANSACTION_ID>,
	"profileMetadata" :
	#METADATA_OP_PROF1,
	"smdpSigned2" : #S_SMDP_SIGNED2_CC,
	"smdpSignature2" :
	<S_SM_DP+_SIGNATURE2>,
	"smdpCertificate" :
	#CERT_S_SM_DPpb_ECDSA
	}

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.

PENDING_NOTIF_EN1	response PendingNotification ::=
	otherSignedNotification : {
	tbsOtherNotification {
	seqNumber <SEQ_NUMBER>,
	profileManagementOperation {
	notificationEnable
	},
	notificationAddress
	#TEST_DP_ADDRESS1,
	iccid #ICCID_OP_PROF1
	},
	euiccNotificationSignature
	<TBS_EUICC_NOTIF_SIG>,
	euiccCertificate #CERT_EUICC_ECDSA,
	eumCertificate #CERT_EUM_ECDSA
	}

PENDING_NOTIF_DEL1	response PendingNotification ::=
	otherSignedNotification : {
	
	tbsOtherNotification {
	seqNumber <SEQ_NUMBER>,
	profileManagementOperation {
	notificationDelete
	},
	notificationAddress
	#TEST_DP_ADDRESS1,
	iccid #ICCID_OP_PROF1
	},
	euiccNotificationSignature
	<TBS_EUICC_NOTIF_SIG>,
	euiccCertificate #CERT_EUICC_ECDSA,
	eumCertificate #CERT_EUM_ECDSA
	}

PENDING_NOTIF_DIS1	response PendingNotification ::=
	otherSignedNotification : {
	
	tbsOtherNotification {
	seqNumber <SEQ_NUMBER>,
	profileManagementOperation {
	notificationDisable
	},
	notificationAddress
	#TEST_DP_ADDRESS1,
	iccid #ICCID_OP_PROF1
	},
	euiccNotificationSignature
	<TBS_EUICC_NOTIF_SIG>,
	euiccCertificate #CERT_EUICC_ECDSA,
	eumCertificate #CERT_EUM_ECDSA
	}

PENDING_NOTIF_DIS2	response PendingNotification ::=
	otherSignedNotification : {
	
	tbsOtherNotification {
	seqNumber <SEQ_NUMBER>,
	profileManagementOperation {
	notificationDisable
	},
	notificationAddress
	#TEST_DP_ADDRESS2,
	iccid #ICCID_OP_PROF2
	},
	euiccNotificationSignature
	<TBS_EUICC_NOTIF_SIG>,
	euiccCertificate #CERT_EUICC_ECDSA,
	eumCertificate #CERT_EUM_ECDSA
	
	}

R_AUTH_SERVER_MATCH_ID_DEV_INFO	resp AuthenticateServerResponse ::=
	authenticateResponseOk {
	euiccSigned1 {
	transactionId <S_TRANSACTION_ID>,
	serverAddress #TEST_DP_ADDRESS1,
	serverChallenge <S_SMDP_CHALLENGE>,
	euiccInfo2 <R_EUICC_INFO2>, -- check
	only that the field is present but not
	the values
	ctxParams1
	#CTX_PARAMS1_MATCH_ID_DEV_INFO
	},
	euiccSignature1 <EUICC_SIGNATURE1>,
	euiccCertificate #CERT_EUICC_ECDSA,
	eumCertificate #CERT_EUM_ECDSA
	}

R_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> }

D.3 ES10x Requests And Responses

D.3.1 ES10x Requests

There are no ES10x requests for this version of the specification.

D.3.2 ES10x Responses

Name	Content

R_PREP_DOWNLOAD_NO_CC	resp PrepareDownloadResponse ::=
	downloadResponseOk : {
	euiccSigned2 {
	transactionId <S_TRANSACTION_ID>,
	euiccOtpk <OTPK_EUICC_ECKA>
	},
	euiccSignature2 <EUICC_SIGNATURE2>
	}

R_PREP_DOWNLOAD_WITH_CC	resp PrepareDownloadResponse ::=
	downloadResponseOk : {
	euiccSigned2 {
	transactionId <S_TRANSACTION_ID>,
	euiccOtpk <OTPK_EUICC_ECKA>,
	hashCc <S_HASHED_CC>
	},
	euiccSignature2 <EUICC_SIGNATURE2>
	}

D.4 ESipa Requests And Responses

D.4.1 ESipa Requests

Name	Content

DELETE_PROFILE_TRIGGER	value1 EuiccPackageRequest ::= {
	
	euiccPackageSigned {
	
	eimId #EIM_ID1,
	
	eidValue #EID1,
	
	counterValue <COUNTER_EIM>,
	
	eimTransactionId <EIM_TRANSACTION_ID>,
	
	euiccPackage psmoList : {

	delete{
	iccid #ICCID_OP_PROF1
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

DELETE_PROFILE_4_TRIGGER	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	delete {
	iccid #ICCID_OP_PROF4
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

DELETE_PROFILE_7_TRIGGER	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	delete {
	iccid #ICCID_OP_PROF7

	} } }, eimSignature <EIM_SIGNATURE> }

DELETE_PROFILE_8_TRIGGER	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	delete {
	iccid #ICCID_OP_PROF8
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

DISABLE_PROFILE_TRIGGER	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	disable {
	iccid #ICCID_OP_PROF1
	}
	}

	}, eimSignature <EIM_SIGNATURE> }

DISABLE_PROFILE_4_TRIGGER	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	  eimId #EIM_ID1,
	  eidValue #EID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  euiccPackage psmoList : {
	  disable {
	  iccid #ICCID_OP_PROF4
	  }
	  }
	},
	eimSignature <EIM_SIGNATURE>
	}

ENABLE_PROFILE_TRIGGER_NO_RB	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	  eimId #EIM_ID1,
	  eidValue #EID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  euiccPackage psmoList : {
	  enable {
	  iccid #ICCID_OP_PROF1
	  }
	  }
	},
	eimSignature <EIM_SIGNATURE>
	}

ENABLE_PROFILE_TRIGGER_RB	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	enable {
	iccid #ICCID_OP_PROF1,
	rollbackFlag
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

PROFILE_DOWNLOAD_TRIGGER_AC	response ProfileDownloadTriggerRequest ::= {
	profileDownloadData activationCode : #<ACTIVATION_CODE>,
	eimTransactionId <EIM_TRANSACTION_ID>
	}

PROFILE_DOWNLOAD_TRIGGER_DEFAULT_SM-DP+	response ProfileDownloadTriggerRequest ::= {
	profileDownloadData contactDefaultSmdp : NULL,
	eimTransactionId <EIM_TRANSACTION_ID>
	}

SET_FALLBACK_TRIGGER	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,

	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	setFallbackAttribute {
	iccid #ICCID_OP_PROF1
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

UNSET_FALLBACK_TRIGGER	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	unsetFallbackAttribute {
	iccid #ICCID_OP_PROF1
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

D.4.2 ESipa Responses

Name	Content
EUICC_PACKAGE_REQUEST_CONF_IMMEDIATE_ENABLE	value1 EuiccPackageRequest ::= { euiccPackageSigned {

	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	configureImmediateEnable : {
	immediateEnableFlag NULL
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

EUICC_PACKAGE_REQUEST_LIST_PROFILE_INFO	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID1,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	listProfileInfo : {
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

GET_EIM_PACKAGE_DELETE_PROFILE_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"

	} }, "euiccPackageRequest": #DELETE_PROFILE_TRIGGER }

GET_EIM_PACKAGE_DELETE_PROFILE_4_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest":
	#DELETE_PROFILE_4_TRIGGER
	}

GET_EIM_PACKAGE_DELETE_PROFILE_7_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest":
	#DELETE_PROFILE_7_TRIGGER
	}

GET_EIM_PACKAGE_DELETE_PROFILE_8_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest":
	#DELETE_PROFILE_8_TRIGGER
	}

GET_EIM_PACKAGE_DISABLE_PROFILE_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest":
	#DISABLE_PROFILE_TRIGGER
	}

GET_EIM_PACKAGE_DISABLE_PROFILE_4_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {

	"status" : "Executed-Success" } }, "euiccPackageRequest": #DISABLE_PROFILE_4_TRIGGER }

GET_EIM_PACKAGE_ENABLE_PROFILE_NO_RB_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest": #ENABLE_PROFILE_TRIGGER_NO_RB
	}

GET_EIM_PACKAGE_ENABLE_PROFILE_RB_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest": #ENABLE_PROFILE_TRIGGER_RB
	}

GET_EIM_PACKAGE_EPR_CONF_IMMEDIATE_ENABLE_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest": #EUICC_PACKAGE_REQUEST_CONF_IMMEDIATE_ENABLE,
	}

GET_EIM_PACKAGE_EPR_LIST_PROFILE_INFO_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest": #EUICC_PACKAGE_REQUEST_LIST_PROFILE_INFO,

	}

GET_EIM_PACKAGE_PROFILE_DOWNLOAD_TRIGGER_AC_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	
	"profileDownloadTriggerRequest": #PROFILE_DOWNLOAD_TRIGGER_AC
	
	}

GET_EIM_PACKAGE_PROFILE_DOWNLOAD_TRIGGER_DEFAULT_SM-DP+_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	
	"profileDownloadTriggerRequest": #PROFILE_DOWNLOAD_TRIGGER_DEFAULT_SM-DP+
	
	}

GET_EIM_PACKAGE_SET_FALLBACK_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest": #SET_FALLBACK_TRIGGER
	}

GET_EIM_PACKAGE_UNSET_FALLBACK_TRIGGER_OK	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest": #UNSET_FALLBACK_TRIGGER
	}

R_EPR_DELPR_OK	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  DeleteProfileResult : ok
	}
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_DELPR>
	  }
	}

R_EPR_DPR_OK	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  DisableProfileResult : ok
	}
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_DPR>
	  }
	}

R_EPR_DELPR_ERR_PPR	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	DisableProfileResult : disallowedByPolicy}
	},
	euiccSignEPR <EUICC_SIGN_EPR_DELPR>
	}
	}

R_EPR_DPR_ERR_NOT_EN	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	DisableProfileResult : profileNotInEnabledState }
	},
	euiccSignEPR <EUICC_SIGN_EPR_DPR>
	}
	}

R_EPR_DPR_ERR_PPR	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  DisableProfileResult : disallowedByPolicy}
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_DPR>
	  }
	}

R_EPR_EPR_OK	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  EnableProfileResult : ok }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_EPR>
	  }
	}

R_EPR_EPR_ERR_NOT_DIS	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : profileNotInDisabledState }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	}
	}

R_EPR_EPR_ERR_PPR	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : disallowedByPolicy }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	}

R_EPR_EPR_ERR_UNKNOWN	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  EnableProfileResult : undefinedError }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_EPR>
	  }
	}

R_EPR_PDTR_OK	response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult profileDownloadTriggerResult : {
	  eimTransactionId <EIM_TRANSACTION_ID>,
	profileDownloadTriggerResultData profileInstallationResult : {
	  profileInstallationResultData {
	  transactionId <S_TRANSACTION_ID>,
	  notificationMetadata {
	  seqNumber <SEQ_NUMBER>,
	profileManagementOperation notificationInstall,
	  notificationAddress #TEST_EIM_ADDRESS1,
	  iccid #ICCID_OP_PROF1
	  },

	smdpOid #S_SM_DP+_OID,
	
	finalResult successResult :
	{
	
	aid <ISD_P_AID>,
	
	simaResponse
	#SIMA_RESULT_OK
	
	}
	
	},
	
	euiccSignPIR <EUICC_SIGN_PIR>
	
	}
	
	}
	
	}

R_EPR_EPR_CIER_OK	value1 ProvideEimPackageResult ::=
	{
	
	eidValue #EID1,
	
	eimPackageResult
	euiccPackageResult :
	euiccPackageResultSigned : {
	
	euiccPackageResultDataSigned {
	
	eimId #EIM_ID1,
	
	counterValue <COUNTER_EIM>,
	
	eimTransactionId
	<EIM_TRANSACTION_ID>,
	
	seqNumber <SEQ_NUMBER>,
	
	euiccResult {
	
	configureImmediateEnableResult : ok
	}
	
	},
	
	euiccSignEPR
	<EUICC_SIGN_EPR_CIER>
	
	}
	
	}

R_EPR_EPRRE_INVALID_PACKAGE_FORMAT	value1 ProvideEimPackageResult ::= { eidValue #EID1,

	eimPackageResult eimPackageResultResponseError : {
	
	eimPackageResultErrorCode invalidPackageFormat
	
	}
	
	}

R_EPR_EPR_LPIR_OK	value1 ProvideEimPackageResult ::= {
	
	eidValue #EID1,
	
	eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	
	euiccPackageResultDataSigned {
	
	eimId #EIM_ID1,
	
	counterValue <COUNTER_EIM>,
	
	eimTransactionId <EIM_TRANSACTION_ID>,
	
	seqNumber <SEQ_NUMBER>,
	
	euiccResult {
	
	listProfileInfoResult : profileInfoListOk : {
	
	{
	
	<PROFILE_INFO>
	
	}
	
	}
	
	}
	
	},
	
	euiccSignEPR <EUICC_SIGN_EPR_LPIR>
	
	}
	
	}

R_EPR_SET_FALLBACK_OK	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  SetFallbackAttributeResult: ok }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_SET_FALLBACK>
	  }
	}

R_EPR_SET_FALLBACK_ERR_PROFILE_NOT_AVAILABLE	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  setFallbackAttributeResult: iccidOrAidNotFound }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_SET_FALLBACK>
	  }
	}

R_EPR_SET_FALLBACK_ERR_FALLBACK_PROFILE_ENABLED	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  setFallbackAttributeResult: fallbackProfileEnabled }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_SET_FALLBACK>
	  }
	}

R_EPR_SET_FALLBACK_ERR_PROFILE_NOT_ALLOWED	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	  setFallbackAttributeResult: fallbackNotAllowed }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_SET_FALLBACK>
	  }
	}

R_EPR_UNSET_FALLBACK_OK	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	
	UnsetFallbackAttributeResult: ok
	}
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_UNSET_FALLBACK>
	  }
	}

R_EPR_UNSET_FALLBACK_ERR_FALLBACK_PROFILE_ENABLED	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	
	unsetFallbackAttributeResult: fallbackProfileEnabled }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_UNSET_FALLBACK>
	  }
	}

R_EPR_UNSET_FALLBACK_ERR_NO_FALLBACK_PROFILE	Response ProvideEimPackageResult ::= {
	  eidValue #EID1,
	  eimPackageResult euiccPackageResult : euiccPackageResultSigned : {
	  euiccPackageResultDataSigned {
	  eimId #EIM_ID1,
	  counterValue <COUNTER_EIM>,
	  eimTransactionId <EIM_TRANSACTION_ID>,
	  seqNumber <SEQ_NUMBER>,
	  euiccResult {
	
	unsetFallbackAttributeResult: noFallbackAttribute }
	  },
	  euiccSignEPR <EUICC_SIGN_EPR_UNSET_FALLBACK>
	  }
	}

S_EIM_ACKNOWLEDGEMENT	{
	  "header" : {
	  "functionExecutionStatus" : {
	  "status" : "Executed-Success"
	  }
	  },
	
	  "eimAcknowledgements": #EIM_ACKNOWLEDGEMENTS,
	
	}

EIM_ACKNOWLEDGEMENTS	value1 EimAcknowledgements ::= {
	
	  <SEQ_NUMBER>
	
	}

D.5 ES11 Requests And Responses

D.5.1 ES11 Requests

There are no ES11 requests for this version of the specification.

D.5.2 ES11 Responses

There are no ES11 responses for this version of the specification.

D.6 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.

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
      '61184F10A0000000871002FF33FF0189000001005004555349 4D'H
  },
  ef-arr {
    fileDescriptor : {
      fileDescriptor '42210025'H,
      lcsi '05'H,
      efFileSize '022B'H
    },
    fillFileContent : '8001019000800102A40683010195010880015 8A40683010A950108'H,
    fillFileOffset : 10,
    fillFileContent : '800101A4068301019501088001 5AA40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent : '80015BA40683010A950108'H,
    fillFileOffset : 26,
    fillFileContent : '800101900080015A9700'H,
    fillFileOffset : 27,
    fillFileContent : '800103A4068301019501088001 58A40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent : '800111A4068301019501088001 4AA40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent :
'800103A4068301019501088001 58A40683010A950108840132A406830101950108'H,
    fillFileOffset : 4,
    fillFileContent :
'800101A4068301019501088001 02A4068301819501088001 58A40683010A950108'H,
    fillFileOffset : 4,
    fillFileContent : '800101900080011AA4068301019501088001 40A40683010A950108'H,
    fillFileOffset : 10,
    fillFileContent : '800101900080015AA40683010A950108'H,
    fillFileOffset : 21,
    fillFileContent : '8001019000800118A40683010A9501088001429700'H,
    fillFileOffset : 16,
    fillFileContent : '800101A4068301019501088001 5A9700'H,
    fillFileOffset : 21,
    fillFileContent : '800113A4068301019501088001 48A40683010A950108'H,
    fillFileOffset : 15,
    fillFileContent : '80015EA40683010A950108'H,
    fillFileOffset : 26,
    fillFileContent  '8001019000800102A010A40683010195
                    0108A4068301029501088001 58A40683
```

```
                    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 '082943019134876765'H
  },
  ef-arr {
    fileDescriptor {
      linkPath '2F06'H
    }
  },
  ef-ust {
    -- Service Dialling Numbers, Short Message Storage…
    fillFileContent '0A2E178CE73204000000000000'H
  },
  ef-spn {
    -- ASCII format: "GSMA eUICC"
    fillFileContent '0247534D41206555494343FFFFFFFFFFFF'H
  },
  ef-est {
    -- Services deactivated
    fillFileContent '00'H
  },
  ef-acc {
    -- Access class 4
    fillFileContent '0040'H
  },
  ef-ecc {
```

```
    -- Emergency Call Code 911
    fillFileContent '19F1FF01'H
  }
}

usimPin ProfileElement ::= pinCodes : {
  pin-Header {
    mandated NULL,
    identification 5
  },
  pinCodes pinconfig : {
    {
      keyReference secondPINAppl1,
      pinValue '39323338FFFFFFFF'H
      unblockingPINReference secondPUKAppl1,
      -- PIN is Enabled
      pinAttributes 1,
      -- maxNumOfAttemps:2, retryNumLeft:2
      maxNumOfAttemps-retryNumLeft 34
    }
  }
}

akaParamValue ProfileElement ::= akaParameter : {
  aka-header {
    mandated NULL,
    identification 6
  },
  algoConfiguration algoParameter : {
    algorithmID milenage,
    -- RES and MAC 64 bits, CK and IK 128 bits
    algorithmOptions '01'H,
    key '000102030405060708090A0B0C0D0E0F'H,
    opc '0102030405060708090A0B0C0D0E0F00'H,
    -- rotationConstants uses default: '4000204060'H
    -- xoringConstants uses default value
    authCounterMax '010203'H
  }
  -- sqnOptions uses default: '02'H
  -- sqnDelta uses default: '000010000000'H
  -- sqnAgeLimit uses default: '000010000000'H
  -- sqnInit uses default: all bytes zero
}

mnoSdValue ProfileElement ::= securityDomain : {
```

```
sd-Header {
  mandated NULL,
  identification 7
},
instance {
  applicationLoadPackageAID 'A0000001515350'H,
  classAID 'A000000151535041'H,
  instanceAID 'A000000151000000'H,
  applicationPrivileges '82FC80'H,
  -- Secured
  lifeCycleState '0F'H,
  -- SCP80 supported
  applicationSpecificParametersC9 '810280008201F08701F0'H,
  -- other parameters MAY be necessary
  applicationParameters {
    -- TAR: B20100, MSL: 12
    uiccToolkitApplicationSpecificParametersField
      '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_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_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

Annex F IUT Settings

F.3 Device Settings

Device Setting name	Description
IUT_TLS_VERSION	Highest TLS protocol version supported by IPAd, at least v1.2. By versions higher then TLS v1.2 backwards compatibility is assumed.
IUT_IPAd_NOTIFICATION_TIMEOUT	Timeout in seconds for IPAd to send a Notification to the SM-DP+ on ES9+ interface assuming IP connection is available.
IUT_IPAd_TLS_INIT	Timeout in seconds for IPAd to send TLS Client Hello to the SM-DP+ (or SM-DS) on ES9+ (or ES11) interface assuming IP connection is available. The timeout SHALL start after the S-EIM sends the eIM Package with Profile Download Trigger.
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 IPAd for the End User Intent confirmation starting when the LPAd displays the dialog for confirmation.
IUT_GSM_GERAN_REL	If GSM/GERAN is supported, this is the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}.
IUT_IMEI	International Mobile Equipment Identity value of the Device in human readable format, including the check digit. The value is used as a reference for verification of the TAC (mandatory) and IMEI (optional) retrieved from DeviceInfo.
IUT_LPAd_Confirmation	Description of the way to perform Authenticated Confirmation (devices supporting SGP.22 v2.2.2 or earlier) or Strong Confirmation (devices supporting SGP.22 v2.3 or later).
IUT_LPAd_CI	CI subjectPublicKeyInfo of CERT.CI.ECDSA (used to verify CERT.DP.TLS) stored in LPAd. Based on NIST [11] in this version of specification.
IUT_LPAd_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_IPAd_Triggering	Description of the way how to perform manually by the tester the triggering of the establishment of the secure connection on ESipa and the triggering of eIM Package Retrieval.
IUT_LPAd_SESSION_CLOSE_TIMEOUT	Timeout in seconds for LPAd to send a next command for Profile Download to the SM-DP+ (or SM-DS) on ES9+ (or ES11) interface assuming IP connection is available. The timeout SHALL start after sending of the previous command by the LPAd.
IUT_LTE_EUTRAN_REL	If LTE/E-UTRAN is supported, this SHALL be the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}.

Device Setting name	Description
IUT_NFC_REL	If NFC is supported, this SHALL be the highest (version, revision) number of TS.26 [15], encoded as the octet string {version, revision, 0}.
IUT_UMTS_UTRAN_REL	If UMTS/UTRAN is supported, this SHALL be the highest 3GPP release N fully supported by the Device, encoded as the octet string {N, 0, 0}.

F.4 Common Settings

In order to execute the test cases defined in this document, the IUT provider SHALL deliver following settings:

IUT Setting name	Description
IUT_RSP_VERSION	Version of SGP.22 supported by the IUT encoded as a string of three integers separated with dots (for example: 2.1.0). In the scope of this specification, this value SHALL indicate one of the versions of SGP.22 for which this specification contains test cases, as specified in section 1.2.

Annex G Initial States

Unless it is defined differently in a particular test case, the IUTs SHALL be set in the following initial state before the test case execution.

G.1 Device

G.1.1 Device (default)

The Device is “powered on”.

The Device is in the normal execution mode after Device boot-up and Device initial configuration. The Device is NOT in the Test Mode.

The IPAd 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 IPAd can be triggered to establish the secured communication with the eIM.

If the IPAd supports O_IPA_EIM_PACKAGE_RETRIEVAL the IPAd can be triggered to send ESipa.GetEimPackage method to the eIM.

If the IPAd supports O_IPA_DIRECT the IPAd shall provide a way for the tester to input the confirmation code during profile download.

The IPAd has the address of the associated S_eIM.

The Device contains a Test eUICC pre-configured as defined below in G.1.3.

G.1.2 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 Test eUICC supports the fallback mechanism

The ECASD is configured with at least the following Keys and Certificates based on NIST P-256 [11] for this version of the SGP.33:

- The Test eUICC’s Private Key #SK_EUICC_ECDSA (for creating ECDSA signatures)
- The Test eUICC’s Certificate #CERT_EUICC_ECDSA (for eUICC authentication) containing the eUICC’s Public Key #PK_EUICC_ECDSA
- The GSMA Certificate Issuer’s Public Key #PK_CI_ECDSA (for verifying off-card entities certificates)
- The Certificate of the EUM #CERT_EUM_ECDSA

Other Certificates and Keys MAY be present. No CRL is loaded on the Test eUICC.

The CI, identified as highest priority in euiccCiPKIdListForSigning, is also selectable in the euiccCiPKIdListForVerification (i.e. all EUM and eUICC Certificates lead to a Root CI certificate linked to a #PK_CI_ECDSA contained in the eUICC).

This CI corresponds to the SubjectKeyIdentifier of one of the #CERT_CI_ECDSA defined in sections G.2.2 and G.2.3.

For devices supporting O_D_REMOVABLE_DOWNLOAD_PPR, the Test eUICC SHALL contain the RAT configuration specified in #PPRS_ALLOWED_EUC_NOT_REQ

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
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.

The Test eUICC is associated to the S_eIM and has the S_eIM public key and S_eIM ID (#EIM_ID1).

The COUNTER_EIM is set to “0”.

Immediate profile enabling is not activated.

Annex K Document Management

K.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
SGP.33-2 V1.2	22 January	Initial version of SGP.33-2 coming from SGP.23 v1.14	ISAG	Yolanda Sanz, GSMA
	05 April 2024	CR00001R01 Adaption of ES9+ and ES11 test cases CR00002R00 LUI settings deletion CR00003R01 IPAd Test Environment CR00004R02 DirectProfileDownload CR00005R06 Procedure Enable Profile IPA Initiated CR00006R02 ES9+ Test Cases CR00007R00 IUT Setting for IPAd triggering		Yolanda Sanz, GSMA
	30 May 2024	CR00008R00 Fix_TC5.4.1.2.2		
	12 December 2024	Revised during the eSIMWG3.117		Yolanda Sanz, GSMA
	13 December 2024	Revised during the eSIMWG3.117		Yolanda Sanz, GSMA
	17 December 2024	Following eSIMWG116_2 discussion version updated to v1.2 (note v1.0 and v1.1 were never published, updated to 1.2 to align numbering with Core Specifications.) CR00009R01 ES11 Test Cases CR00010R02 Procedure Disable Profile_IPA_Initiated CR00011R03 Procedure Delete Profile_IPA_Initiated CR00012R02 Procedure Set Unset FallbackAttribute CR00013R00 Procedure Enable Profileupdate		Stephen Packer, GSMA

K.2 Other Information

Type	Description
Document Owner	eSIMG
Editor / Company	Stephen Packer, GSMA

It is our intention to provide a quality product for your use. If you find any errors or omissions, please contact us with your comments. You may notify us at prd@gsma.com.

Your comments or suggestions & questions are always welcome.