SGP.33-3 eUICC IoT Manager 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 eUICC. The Implementations Under Test (IUT) are:

- The eUICC IoT Manager (eIM)

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
eIM	eUICC IoT Manager
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
A TransactionId is sent with eUICC Package Request	O_S_TRID
The eIM supports the eIM Package Retrieval mode	O_S_PKG_RETRIEVAL
The eIM uses TLS protocol over ESipa	O_S_ESIPA_HTTPS

Table 4: Options

2.1.5 Applicability Table

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

Test case	Name	Role	V1.0	Test Env.
4.2.11.2.1	TC_eIM_ES9+'_GetBoundProfilePackage_Nominal	EIM	SGP.23	
4.4.11.2.2	TC_eIM_ES9+'_GetBoundProfilePackage_Retry	EIM	SGP.23	
4.4.11.2.3	TC_eIM_ES9+'_GetBoundProfilePackage_Error	EIM	SGP.23	
4.4.11.2.2	TC_eIM_ES9+'_GetBoundProfilePackage_Retry	EIM	SGP.23	

Test case	Name	Role	V1.0	Test Env.
4.2.12.2.1	TC_eIM_AuthenticateClient_Nominal	EIM	SGP.23	
4.2.12.2.2	TC_eIM_AuthenticateClient_ErrorCases	EIM	SGP.23	
4.2.13.2.1	TC_eIM_ES9+_HandleNotification_Nominal	EIM	SGP.23	
4.2.14.2.1	TC_eIM_ES9+'_CancelSession_Nominal	EIM	SGP.23	
4.4.14.2.2	TC_eIM_ES9+'_CancelSession_EndUserPostponed_Nominal	EIM	SGP.23	
4.2.14.2.3	TC_eIM_ES9+'_CancelSession_Error	EIM	SGP.23	
4.2.14.2.4	TC_eIM_ES9+'_CancelSession_PPRs	EIM	SGP.23	
4.2.15.2.1	TC_eIM_HTTPS_Nominal	EIM	SGP.23	
4.2.15.2.2	TC_eIM_HTTPS_ErrorCases	EIM	SGP.23	
4.2.16.2.1	TC_eIM_ES11'_InitiateAuthentication_Nominal	EIM	SGP.23	
4.2.16.2.2	TC_eIM_ES11'_InitiateAuthentication_ErrorCases	EIM	SGP.23	
4.2.17.2.1	TC_eIM_ES11'_AuthenticateClient_Nominal	EIM	SGP.23	
4.2.17.2.2	TC_eIM_ES11'_AuthenticateClient_ErrorCases	EIM	SGP.23	
4.2.18.2.1	TC_eIM_ES11'_HTTPS_Nominal	EIM	SGP.23	
4.2.18.2.2	TC_EIM_ES11'_HTTPS_Error	EIM	SGP.23	
5.2.1.2.1	TC_eIM_ProfileEnable_TLS_eIM_Pkg_Retrieval	EIM	C3000	

Table 5: Applicability of Tests

Conditional item	Condition
C3000	IF (O_S_PKG_RETRIEVAL AND O_**S**_ESIPA_HTTPS) 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.

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 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.

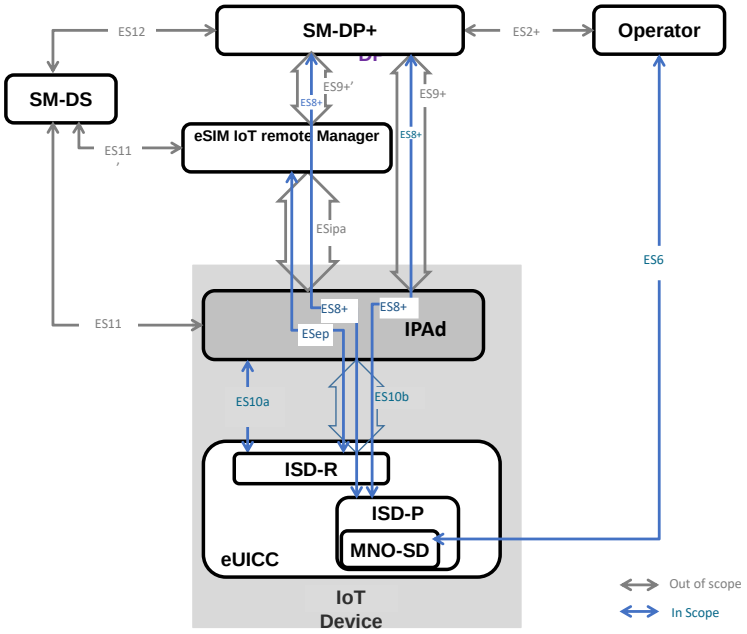
2.2.4 Future Study

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.

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].	Out of 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].	In scope
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

Interface	Between		Description	SGP.33-2
ES11	IPA	SM-DS	Used by the IPA to retrieve Event Records for the respective eUICC as defined in SGP.31 [2].	Out of scope
ES11'	eIM	SM-DS	Used by the eIM to retrieve Event Records for the respective eUICC as defined in SGP.31 [2].	In 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].	In 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.

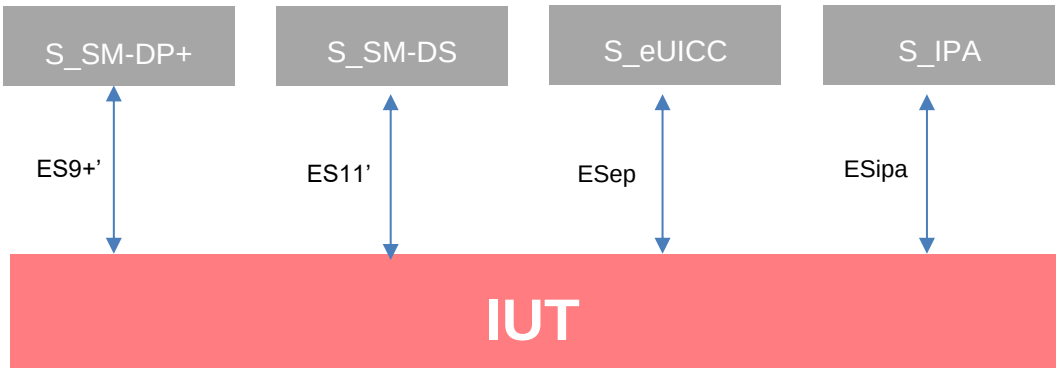
To permit the execution of the different test cases described in this Test Plan, specifics simulators SHALL be used. The simulators that have been defined are listed hereafter:

- 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_EIM: the EIM Simulator
- S_IPAe: the IPAe 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, IPAe, EIM, 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.

3.2.3.1 General (eIM) Test Environment



The Test Environment consists of:

- IUT: the eUICC IoT Manager under the test.
- S_SM-DP+: a simulated SM-DP+ supporting a connection used by the eIM to establish ES9+'
- S_SM-DS: a simulated SM-DS supporting a connection used by the eIM to establish ES11'
- S_eUICC: a simulated eUICC supporting a connection used by the eIM to establish ESep
- S_IPA: a simulated IPA supporting a connection used by the eIM to establish ESipa.

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 eIM Interfaces

4.2.1 ESep (eIM -- eUICC): eUICC Package with single PSMO command: Enable

This function can be tested as Enable Procedures

4.2.1.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

- Section 2.11.1.1
- Section 2.11.1.1.3
- Section 2.11.2.1
- Section 3.3.1
- Section 5.13.1

4.2.1.2 Test Cases

4.2.1.2.1 TC_eIM_ESep.Enable

The test sequences of this section are FFS.

4.2.1.2.2 TC_eIM_ESep.Enable_ErrorCases

The test sequences of this section are FFS.

4.2.2 ESep (eIM -- eUICC): eUICC Package with single PSMO command: Disable

4.2.2.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

- Section 2.11.1.1
- Section 2.11.1.1.3
- Section 2.11.2.1
- Section 5.13.2

- Section 3.3.1

4.2.2.2 Test Cases

4.2.2.2.1 TC_ eIM_ESep.Disable

The test sequences of this section are FFS.

4.2.2.2.2 TC_ eIM_ESep.Disable_ErrorCases

The test sequences of this section are FFS.

4.2.3 ESep (eIM -- eUICC): eUICC Package with single PSMO command: Delete

4.2.3.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

- Section 2.11.1.1
- Section 2.11.1.1.3
- Section 2.11.2.1
- Section 3.3.1
- Section 5.13.3

4.2.3.2 Test Cases

4.2.3.2.1 TC_eIM_ESep.Delete

The test sequences of this section are FFS.

4.2.3.2.2 TC_eIM_ESep.Delete_ErrorCases

The test sequences of this section are FFS.

4.2.4 ESep (eIM -- eUICC): eUICC Package with single PSMO command: ListProfileInfo

This function allows the eIM to retrieve the list of Profile information for installed Profiles including their current state (Enabled/Disabled) and their associated Profile Metadata..

4.2.4.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

- Section 2.11.1.1
- Section 2.11.1.1.3
- Section 2.11.2.1
- Section 3.3.1
- Section 5.13.4

4.2.4.2 Test Cases

4.2.4.2.1 TC_eIM_ESep.ListProfileInfo

The test sequences of this section are FFS.

4.2.4.2.2 TC_ eIM_ESep.ListProfileInfo_ErrorCases

The test sequences of this section are FFS.

4.2.5 ESep (eIM -- eUICC): eUICC Package with single PSMO command: GetRat

This function allows the eIM to retrieve the Rules Authorisation Table (RAT) from the eUICC.

4.2.5.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

- Section 2.11.1.1
- Section 2.11.1.1.3
- Section 2.11.2.1
- Section 3.3.1
- Section 5.13.5

4.2.5.2 Test Cases

4.2.5.2.1 TC_ eIM_ESep.GetRat

Test Sequence #01 Nominal Case

4.2.6 ESep (eIM -- eUICC): eUICC Package with single eCO command: AddEim

This function adds an Associated eIM to the eUICC by providing its eIM Configuration Data including the eimID to the eUICC.

4.2.6.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

4.2.6.2 Test Cases

4.2.6.2.1 TC_eIM_ESep.AddEim

The test sequences of this section are FFS.

4.2.6.2.2 TC_eIM_ESep.AddEim_ErrorCases

The test sequences of this section are FFS.

4.2.7 ESep (eIM -- eUICC): eUICC Package with single eCO command: UpdateEim

This function updates eIM Configuration Data, i.e., the public key or Certificate and the related anti-replay counter value of an Associated eIM with a given eimID within the eUICC while keeping the same eimID.

4.2.7.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

4.2.7.2 Test Cases

4.2.7.2.1 TC_eIM_ESep.UpdateEIM

The test sequences of this section are FFS.

4.2.7.2.2 TC_eIM_ESep.UpdateEim_ErrorCases

The test sequences of this section are FFS.

4.2.8 ESep (eIM -- eUICC): eUICC Package with single eCO command: DeleteEim

This function deletes an Associated eIM identified by its eimID from the eUICC. If the successfully deleted Associated eIM was the last available Associated eIM, the eUICC SHALL allow ES10b.AddInitialEim again.

4.2.8.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

4.2.8.2 Test Cases

4.2.8.2.1 TC_eIM_ESep.DeleteEim

The test sequences of this section are FFS.

4.2.8.2.2 TC_eIM_ESep.DeleteEim_ErrorCases

The test sequences of this section are FFS.

4.2.9 ESep (eIM -- eUICC): eUICC Package with single eCO command: ListEim

This function requests the eUICC to provide a list of all currently configured Associated eIMs to the eIM.

4.2.9.1 Conformance Requirements

References

GSMA RSP Technical Specification [4] and eSIM IoT Technical Specification [3]

4.2.9.2 Test Cases

4.2.9.2.1 TC_eIM_ESep.ListEim

The test sequences of this section are FFS.

4.2.9.2.2 TC_eIM_ESep.ListEim_ErrorCases

Test Sequence #01 Error: Command Error
The test sequence of this error is not defined in this version of the specification

4.2.10 ES9+' (eIM -- SM-DP+): InitiateAuthentication

4.2.10.1 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.

4.2.10.2 Test Cases

4.2.10.2.1 TC_eIM_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 eIM plays the role of LPAd.

4.2.10.2.2 TC_ eIM _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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.11 ES9+' (eIM -- SM-DP+): GetBoundProfilePackage

4.2.11.1 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.

4.2.11.2 Test Cases

4.2.11.2.1 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.4.11.2.2 TC_eIM_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 eIM plays the role of LPAd.

4.4.11.2.3 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.12 ES9+' (eSIM -- SM-DP+): AuthenticateClient

4.2.12.1 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.

4.2.12.2 Test Cases

4.2.12.2.1 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.12.2.2 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.13 ES9+' (eIM – SM-DP+): HandleNotification

4.2.13.1 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.

4.2.13.2 Test Cases

4.2.13.2.1 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.14 ES9+' (eIM – SM-DP+): CancelSession

4.2.14.1 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.

4.2.14.2 Test Cases

4.2.14.2.1 TC_eIM_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 eIM plays the role of LPAdwhere the eIM plays the role of LPAd.

Test Sequence #02 Nominal: VOID

Test Sequence #03 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 eIM plays the role of LPAdwhere the eIM plays the role of LPAd.

Test Sequence #04 Nominal: VOID

.

Test Sequence #05 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 eIM plays the role of LPAd.

4.4.14.2.2 TC_eIM_ES9+'_CancelSession_EndUserPostponed_Nominal

Test Sequence #01 Nominal: VOID

4.2.14.2.3 TC_eIM_ES9+'_CancelSession_Error

Test Sequence #01 Error: VOID

Test Sequence #02 Error: VOID

Test Sequence #03 Error: VOID

4.2.14.2.4 TC_ eIM _ES9+'_CancelSession_PPRs

Test Sequence #01 Nominal: VOID

Test Sequence #02 Nominal: VOID

4.2.15 ES9+' (eIM – SM-DP+): HTTPS

4.2.15.1 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.

4.2.15.2 Test Cases

4.2.15.2.1 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.15.2.2 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.16 ES11’ (eIM – SM-DS): InitiateAuthentication

4.2.16.1 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.

4.2.16.2 Test Cases

4.2.16.2.1 TC_eIM_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 eIM plays the role of LPAd.

4.2.16.2.2 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.17 ES11 (LPA – SM-DS): AuthenticateClient

4.2.17.1 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.

4.2.17.2 Test Cases

4.2.17.2.1 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.17.2.2 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.18 ES11’ (eIM -- SM-DS): HTTPS

4.2.18.1 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.

4.2.18.2 Test Cases

4.2.18.2.1 TC_eIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.18.2.2 TC_EIM_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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

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 eIM plays the role of LPAd.

4.2.19 ESipa (EIM -- LPA): InitiateAuthentication

4.2.19.1 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

4.2.19.2 Test Cases

4.2.19.2.1 TC_eIM_ESipa_ InitiateAuthentication_Nominal

The test sequences of this section are FFS.

4.2.19.2.2 TC_eIM_ESipa_ InitiateAuthentication_ErrorCases

The test sequences of this section are FFS.

4.2.20 ESipa (EIM -- LPA): GetBoundProfilePackage

4.2.20.1 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.

4.2.20.2 Test Cases

4.2.20.2.1 TC_eIM_ESipa_GetBoundProfilePackage_Nominal

The test sequences of this section are FFS.

4.2.20.2.2 TC_eIM_ESipa_GetBoundProfilePackage _ErrorCases

The test sequences of this section are FFS.

4.2.21 ESipa (EIM -- LPA): AuthenticateClient

4.2.21.1 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

4.2.21.2 Test Cases

4.2.21.2.1 TC_eIM_ESipa_AuthenticateClient _Nominal

The test sequences of this section are FFS.

4.2.21.2.2 TC_eIM_ESipa_ AuthenticateClient _ErrorCases

The test sequences of this section are FFS.

4.2.22 ESipa (EIM -- LPA): InitiateAuthentication

4.2.22.1 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

4.2.22.2 Test Cases

4.2.22.2.1 TC_eIM_ESipa_ InitiateAuthentication_Nominal

The test sequences of this section are FFS.

4.2.22.2.2 TC_eSIM_ESipa_ InitiateAuthentication_ErrorCases

The test sequences of this section are FFS.

4.2.23 ESipa (EIM -- LPA): GetBoundProfilePackage

4.2.23.1 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.

4.2.23.2 Test Cases

4.2.23.2.1 TC_eIM_ESipa_GetBoundProfilePackage_Nominal

The test sequences of this section are FFS.

4.2.23.2.2 TC_eIM_ESipa_GetBoundProfilePackage _ErrorCases

The test sequences of this section are FFS.

4.2.24 ESipa (EIM -- LPA): AuthenticateClient

4.2.24.1 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

4.2.24.2 Test Cases

4.2.24.2.1 TC_ eIM _ESipa_AuthenticateClient _Nominal

The test sequences of this section are FFS.

4.2.24.2.2 TC_eIM_ESipa_ AuthenticateClient _ErrorCases

The test sequences of this section are FFS.

4.2.25 ESipa (EIM -- LPA): TransferEimPackage

4.2.25.1 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

4.2.25.2 Test Cases

4.2.25.2.1 TC_eIM_ESipa_TransferEimPackage_Nominal

The test sequences of this section are FFS.

4.2.25.2.2 TC_eIM_ESipa_TransferEimPackage_ErrorCases

The test sequences of this section are FFS.

4.2.26 ESipa (EIM -- LPA): GetEIMPackage

4.2.26.1 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.

4.2.26.2 Test Cases

4.2.26.2.1 TC_eIM_ESipa_GetEIMPackage_Nominal

The test sequences of this section are FFS.

4.2.26.2.2 TC_eIM_ESipa_GetEIMPackage_ErrorCases

The test sequences of this section are FFS.

4.2.27 ESipa (EIM -- LPA): 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.

4.2.27.1 Conformance Requirements

References

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

4.2.27.2 Test Cases

4.2.27.2.1 TC_eIM_ESipa_ ProvideEimPackageResult_Nominal

The test sequences of this section are FFS.

4.2.27.2.2 TC_eIM_ESipa_ ProvideEimPackageResult_ErrorCases

The test sequences of this section are FFS.

4.2.28 ESipa (EIM -- LPA): 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

4.2.28.2 Test Cases

4.2.28.2.1 TC_eIM_ESipa_HandleNotification_Nominal

The test sequences of this section are FFS.

4.2.28.2.2 TC_eIM_ESipa_HandleNotification_ErrorCases

The test sequences of this section are FFS.

4.2.29 ESipa (EIM -- LPA): CancelSession

4.2.29.1 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.

4.2.29.2 Test Cases

4.2.29.2.1 TC_eIM_ESipa_CancelSession_Nominal

The test sequences of this section are FFS.

4.2.29.2.2 TC_eIM_ESipa_CancelSession_ErrorCases

The test sequences of this section are FFS.

5 Procedure - Behaviour Testing

5.1 General Overview

5.2 eIM Procedures

5.2.1 Profile State Management Operation - Enable Profile

5.2.1.1 Conformance Requirements

References

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

5.5.1.2 Test Cases

5.5.1.2.1 TC_eIM_ProfileEnable_TLS_eIM_Pkg_Retrieval

General Initial Conditions	
Entity	**Description of the initial condition**
S_IPAd	The S_IPAD is configured for eIM Package retrieval
S_eUICC	EIM has been associated to the S_eUICC as #EIM_ID1 by configuring eIM Configuration Data
EIM	EID #EID1 is known to the EIM and associated to PROFILE_OPERATIONAL1.

Test Sequence #01 Nominal: Enable an Operational Profile initiated by IPA, with ProvideEimPackageResult – no enabled profile

Initial Conditions	
Entity	**Description of the initial condition**
S_eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
S_eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
S_IPAd	No secure connection is established between eIM and S_IPAd
S_SM-DP+	No secure connection is established between eIM and S_SM-DP+
EIM	An Enable Profile PSMO request #ENABLE_PROFILE1 is pending for #EID1

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IF O_S_TRID			
1	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1)
2	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(# SERVER_ADDRESS, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK_NOTIF_EN1))	MTD_HTTP_RESP_ESIPA (#EIM_ACKNOWLEDGEMENT_EN1) Verify that returned <SEQ_NUMBER> values are matching values in the `ProvideEimPackageResult` request and are provided in the same order
ENDIF			
IF NOT O_S_TRID			
3	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE,	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1_NO_TRID)

		MTD_GET_EIM_PACKAGE (#EID1))	
4	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(# SERVER_ADDRESS, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK_NOTIF_EN1_NO_TRID))	MTD_HTTP_RESP_ESIPA (#EIM_ACKNOWLEDGEMENT_EN1) Verify that returned <SEQ_NUMBER> values are matching values in the `ProvideEimPackageResult` request and are provided in the same order
ENDIF			
5	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+' (S_SERVER configured with #TEST_DP_ADDRESS1)		
6	PROC_ES9+'_HANDLE_NOTIF_EN1		

Test Sequence #02 Nominal: Enable an Operational Profile initiated by IPA, with HandleNotification – no enabled profile

Initial Conditions	
Entity	**Description of the initial condition**
S_eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
S_eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
S_IPAd	No secure connection is established between eIM and S_IPAd
EIM	An Enable Profile PSMO request #ENABLE_PROFILE1 is pending for #EID1

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IF O_S_TRID			
1	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1)
2	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK))	#R_HTTP_204_OK
ENDIF			
IF NOT O_S_TRID			

3	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1_NO_TRID)
4	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK_NO_TRID))	#R_HTTP_204_OK
ENDIF			
5	S_IPAd → EIM	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF(#PENDING_NOTIF_EN1))	#R_HTTP_204_OK
6	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+' (S_SERVER configured with #TEST_DP_ADDRESS1)		
7	PROC_ES9+'_HANDLE_NOTIF_EN1		

Test Sequence #03 Nominal: Enable an Operational Profile with implicit disabling of the formerly enabled Profile, initiated by IPA, with ProvideEimPackageResult

Initial Conditions	
Entity	**Description of the initial condition**
S_eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
S_eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
S_eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
S_eUICC	The PROFILE_OPERATIONAL2 is in Enabled state.
S_IPAd	No secure connection is established between eIM and S_IPAd
EIM	An Enable Profile PSMO request #ENABLE_PROFILE1 is pending for #EID1

Step	**Direction**	**Sequence / Description**	**Expected result**
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IF O_S_TRID			
1	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1)

2	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(# SERVER_ADDRESS, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK_NOTIF_EN1_DIS2))	MTD_HTTP_RESP_ESIPA (#EIM_ACKNOWLEDGEMENT_EN1_DIS2) Verify that returned <SEQ_NUMBER> values are matching values in the `ProvideEimPackageResult` request and are provided in the same order
ENDIF			
IF NOT O_S_TRID			
3	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1_NO_TRID)
4	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(# SERVER_ADDRESS, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK_NOTIF_EN1_DIS2_NO_TRID))	MTD_HTTP_RESP_ESIPA (#EIM_ACKNOWLEDGEMENT_EN1_DIS2) Verify that returned <SEQ_NUMBER> values are matching values in the `ProvideEimPackageResult` request and are provided in the same order
ENDIF			
5	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+' (S_SERVER configured with #TEST_DP_ADDRESS1)		
6	PROC_ES9+'_HANDLE_NOTIF_EN1 See NOTE1		
7	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+' (S_SERVER configured with #TEST_DP_ADDRESS2)		
8	PROC_ES9+'_HANDLE_NOTIF_DIS2 See NOTE1		
NOTE1: The Notifications (steps 6 and 8) MAY be sent to SM-DP+s in any order or in parallel.			

Test Sequence #04 Nominal: Enable an Operational Profile with implicit disabling of the formerly enabled Profile, initiated by IPA, with HandleNotification

Initial Conditions	
Entity	**Description of the initial condition**
S_eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
S_eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
S_eUICC	The PROFILE_OPERATIONAL2 is installed on the eUICC.
S_eUICC	The PROFILE_OPERATIONAL2 is in Enabled state.
S_IPAd	No secure connection is established between eIM and S_IPAd

EIM	An Enable Profile PSMO request #ENABLE_PROFILE1 is pending for #EID1

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IF O_S_TRID			
1	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1)
2	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK))	#R_HTTP_204_OK
ENDIF			
IF NOT O_S_TRID			
3	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1_NO_TRID)
4	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(#TEST_EIM_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF_EIM_PACKAGE_RESULT (#R_EPR_EPR_OK_NO_TRID))	#R_HTTP_204_OK
ENDIF			
5	S_IPAd → EIM	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF(#PENDING_NOTIF_EN1))	#R_HTTP_204_OK
6	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+' (S_SERVER configured with #TEST_DP_ADDRESS1)		
7	PROC_ES9+'_HANDLE_NOTIF_EN1		
8	S_IPAd → EIM	MTD_HTTP_REQ(#TEST_DP_ADDRESS1, #PATH_HANDLE_NOTIF_IPA, MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS2))	#R_HTTP_204_OK
9	PROC_TLS_INITIALIZATION_SERVER_AUTH on ES9+' (S_SERVER configured with #TEST_DP_ADDRESS2)		
10	PROC_ES9+'_HANDLE_NOTIF_DIS2		
NOTE1: The Notifications (steps 7 and 10) MAY be sent to SM-DP+s in any order or in parallel.			

5.5.1.2.2 TC_eIM_ProfileEnable_TLS_IPA_initiated_ErrorCases

General Initial Conditions	
Entity	**Description of the initial condition**
S_IPAd	The S_IPAD is configured for eIM Package retrieval
S_eUICC	EIM has been associated to the S_eUICC as #EIM_ID1 by configuring eIM Configuration Data
EIM	- EID #EID1 is known to the EIM and associated to PROFILE_OPERATIONAL1, - EID #EID2 is not known to the EIM

Test Sequence #01 Error: Enable an Operational Profile initiated by IPA, with ProvideEimPackageResult – wrong EID

Initial Conditions	
Entity	**Description of the initial condition**
S_eUICC	The PROFILE_OPERATIONAL1 is installed on the eUICC.
S_eUICC	The PROFILE_OPERATIONAL1 is in Disabled state.
S_IPAd	No secure connection is established between eIM and S_IPAd
EIM	An Enable Profile PSMO request #ENABLE_PROFILE1 is pending for #EID1

Step	Direction	Sequence / Description	Expected result
IC1	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA		
IF O_S_TRID			
1	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1)
2	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(# SERVER_ADDRESS, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_EID))	MTD_HTTP_RESP_ESIPA (#EIM_PK_RES_ERR_WRONG_EID)
ENDIF			
IF NOT O_S_TRID			

3	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA (#SERVER_ADDRESS, #PATH_GET_EIM_PACKAGE, MTD_GET_EIM_PACKAGE (#EID1))	MTD_HTTP_RESP_ESIPA(#GET_EIM_PACKAGE_ENABLE_PROFILE_1_NO_TRID)
4	S_IPAd → EIM	MTD_HTTP_REQ_ESIPA(# SERVER_ADDRESS, #PATH_PROVIDE_EIM_PACKAGE_RESULT, MTD_PROVIDE_EIM_PACKAGE_RESULT (#R_EPR_EPR_ERR_EID_NO_TRID))	MTD_HTTP_RESP_ESIPA (#EIM_PK_RES_ERR_WRONG_EID)

Annex A Constants

A.1 Generic Constants

Name	Content
TLS_VERSION_1_2	1.2 The minimum TLS Version supported by the Server.
S_SESSION_ID_EMPTY	Empty TLS session ID to identify a new session, with the Length set as `'zero'`.
S_EXT_SHA256_ECDSA	TLS extension data for "supported_signature_algorithms" set as: o `HashAlgorithm sha256 (04)` and o SignatureAlgorithm ecdsa (03).
CHANGE_CIPHER_SPEC	1
SERVER_ADDRESS	FQDN of the SERVER Under Test: • #IUT_EIM_ADDRESS
PATH_GET_EIM_PACKAGE	`/gsma/rsp2/esipa/getEimPackage`
PATH_PROVIDE_EIM_PACKAGE_RESULT	`/gsma/rsp2/esipa/provideEimPackageResult`

Name	Content
PATH_HANDLE_NOTIF	/gsma/rsp2/es9plus/handleNotification
PATH_HANDLE_NOTIF_IPA	/gsma/rsp2/esipa/handleNotification
EIM_ID	Identifier of the eIM SERVER Under Test: • #IUT_EIM_ID
EID1	0x89 04 90 32 12 34 51 23 45 12 34 56 78 90 12 35
EID2	0x89 04 90 32 11 23 41 23 40 12 34 56 78 90 13 75
ICCID_OP_PROF1	0x98 92 09 01 21 43 65 87 09 F5
TEST_DP_ADDRESS1	testsmdpplus1.example.com
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

A.2 Test Certificates and Test Keys

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

- NIST P-256 curve, defined in Digital Signature Standard [11]
- brainpoolP256r1 curve, defined in RFC 5639 [8]
- FRP256V1 curve, defined in ANSSI ECC [9]

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

Name	Description
CERT_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_SERVER_TLS	CERT.SERVER.TLS certificate of the Server under test, based on NIST or Brainpool for this version of the specification, where the Certificate MAY be one of the following depending on the type of Server and whether it is a Server under test or a Server simulator: • #CERT_EIM_TLS • #CERT_S_SM_DP_TLS • #CERT_S_SM_DS_TLS

Annex B Dynamic Content

Variable	Description
TLS_CIPHER_SUITES	TLS cipher suite list supported by S_IPAd or the Client (EIM) under test.
SEL_TLS_CIPHER_SUITE	TLS cipher suite selected by the Server
SESSION_ID_RANDOM	Random value of the TLS session.
CLIENT_TLS_EPHEM_KEY	Client's ephemeral key and associated information.
SERVER_FINISHED	The first protected message with the negotiated algorithms, keys, and secrets. It is the Hash of the concatenation of all the data from all messages in this handshake up to, but not including, this message i.e. all handshake messages starting at ClientHello up to, but not including, this Finished message itself. NOTE: ChangeCipherSpec messages, alerts, and any other record type are not handshake messages and are not included in the hash computations. Also, HelloRequest messages are omitted from handshake hashes.
COUNTER_EIM	Integer value coded maximum on two bytes. Incremented each time the IUT (EIM) generates an eUICC Package Request.
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).
SEQ_NUMBER	Sequence Number related to a Notification Metadata generated by the eUICC. Note: if this variable appears multiple times in a request or a response, each variable has different values.
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.
TBS_EUICC_NOTIF_SIG	The eUICC signature generated over tbsOtherNotification. NotificationMetadata, coded as ASN.1 OCTET STRING.
SESSION_ID_CLIENT	Random or empty value of the TLS session_id in ClientHello.
EXT_SHA256_ECDSA	TLS extension data for "supported_signature_algorithms" set as a minimum of HashAlgorithm sha256 (04) and SignatureAlgorithm ecdsa (03).
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.

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_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 ESIPA, ES9+’ or ES11.
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 ESIPA, 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.

Method	MTD_TLS_CLIENT_KEY_EXCH_ETC
Description	Finalizes the Transport Layer Security (TLS) handshake in Server authentication mode on ESIPA, 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_SERVER_END
Description	Send or checks the finalization of the Transport Layer Security (TLS) handshake in Server or Mutual authentication mode on ESIPA, ES9+ or ES11 (Server side).
Parameter(s)	• paramChangeCipherSpec: ChangeCipherSpec message • paramFinish: Finished message
Details	Sends a ChangeCipherSpec and Finished message in this order according to the parameters defined above.

Method	MTD_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_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_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_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_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
	}

C.2 Procedures

	Procedure	PROC_TLS_INITIALIZATION_SERVER_AUTH_ESIPA	
	Description	Establishes the Transport Layer Security (TLS) v1.2 connection between the Client IPAd (S_IPAd) and the Server EIM using Server authentication mode on ES ipa with Variant O certificate.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	S_IPAd → EIM	MTD_TLS_CLIENT_HELLO(#TLS_VERSION_1_2, <TLS_CIPHER_SUITES>, #S_SESSION_ID_EMPTY, #S_EXT_SHA256_ECDSA)	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, <SEL_TLS_CIPHER_SUITE>, <SESSION_ID_RANDOM>, #CERT_SERVER_TLS)
2	EIM → S_IPAd	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)	MTD_TLS_SERVER_END(#CHANGE_CIPHER_SPEC, <SERVER_FINISHED>)

	Procedure	PROC_TLS_INITIALIZATION_SERVER_AUTH		
	Description	Establishes the Transport Layer Security (TLS) v1.2 connection between the Client EIM and (S_)SERVER using Server authentication mode on ES9+ or ES11.		
Step	**Direction**	**Sequence / Description**	**Expected result**	**REQ**
1	EIM → 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 → EIM	MTD_TLS_SERVER_HELLO_ETC(#TLS_VERSION_1_2, #S_TLS_CIPHER_SUITE, <S_SESSION_ID_SERVER>, #CERT_SERVER_TLS)	MTD_TLS_CLIENT_KEY_EXCH_ETC(<CLIENT_TLS_EPHEM_KEY>)	
3	S_SERVER → EIM	Finalize TLS Handshake (send Server ChangeCipherSpec and Finished messages)	HTTPS connection established	

	Procedure	PROC_ES9+'_HANDLE_NOTIF_EN1	
	Description	Handle Notification procedure.	
Step	**Direction**	**Sequence / Description**	**Expected result**
1	EIM→ 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+ → EIM	#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	EIM→ S_SM-DP+	Send ES9+'.HandleNotification method	MTD_HTTP_REQ(#TEST_DP_ADDRESS2, #PATH_HANDLE_NOTIF, MTD_HANDLE_NOTIF(#PENDING_NOTIF_DIS2))
2	S_SM-DP+ → EIM	#R_HTTP_204_OK	No error
NOTE 1: Other Notifications MAY be sent within the same HTTPS session.			

Annex D Commands And Responses

D.1 ES9+' Requests And Responses

D.1.1 ES9+' Requests

Name	Content

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_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
	}

D.1.2 ES9+' Responses

There are no ES9+' Responses defined in this version of the specification

D.2 ES11' Requests And Responses

D.2.1 ES11' Requests

There are no ES11' Requests defined in this version of the specification

D.2.2 ES11' Responses

There are no ES11' Responses defined in this version of the specification

D.3 ESipa Requests and Responses

D.3.1 ESipa Requests

Name	Content

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

R_EPR_EPR_OK_NOTIF_EN1	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult : ePRAndNotifications : {
	euiccPackageResult : euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : ok }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	},
	notificationList : {
	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
	}
	}
	}
	}

R_EPR_EPR_OK_NOTIF_EN1_NO_TRID	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult : ePRAndNotifications :
	{
	euiccPackageResult :
	euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : ok }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	},
	notificationList : {
	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
	}
	}
	}
	}

R_EPR_EPR_OK_NOTIF_EN1_DIS2	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult : ePRAndNotifications :
	{
	euiccPackageResult :
	euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId
	<EIM_TRANSACTION_ID>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : ok }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	},
	notificationList : {
	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
	},
	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_EPR_EPR_OK_NOTIF_EN1_DIS2_NO_TRID	Response ProvideEimPackageResult ::= {
	eidValue #EID1,
	eimPackageResult : ePRAndNotifications :
	{
	euiccPackageResult :
	euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : ok }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	},
	notificationList : {
	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
	},
	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_EPR_EPR_ERR_EID	Response ProvideEimPackageResult ::= {
	eidValue #EID2,
	eimPackageResult : ePRAndNotifications :
	{
	euiccPackageResult :
	euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId
	<EIM_TRANSACTION_ID>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : ok }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	},
	notificationList : {
	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
	}
	}
	}
	}

R_EPR_EPR_ERR_EID_NO_TRID	Response ProvideEimPackageResult ::= {
	eidValue #EID2,
	eimPackageResult : ePRAndNotifications :
	{
	euiccPackageResult :
	euiccPackageResultSigned : {
	euiccPackageResultDataSigned {
	eimId #EIM_ID1,
	counterValue <COUNTER_EIM>,
	seqNumber <SEQ_NUMBER>,
	euiccResult {
	EnableProfileResult : ok }
	},
	euiccSignEPR <EUICC_SIGN_EPR_EPR>
	},
	notificationList : {
	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
	}
	}
	}
	}

D.3.2 ESipa Responses

Name	Content

GET_EIM_PACKAGE_ENABLE_PROFILE_1	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest":
	#ENABLE_PROFILE1
	}

ENABLE_PROFILE1	value1 EuiccPackageRequest ::= {

	euiccPackageSigned {
	eimId #EIM_ID,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	eimTransactionId <EIM_TRANSACTION_ID>,
	euiccPackage psmoList : {
	enable {
	iccid #ICCID_OP_PROF1
	}
	}
	},
	eimSignature <EIM_SIGNATURE>
	}

GET_EIM_PACKAGE_ENABLE_PROFILE_1_NO_TRID	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	"euiccPackageRequest": #ENABLE_PROFILE_1_NO_TRID
	}

ENABLE_PROFILE_1_NO_TRID	value1 EuiccPackageRequest ::= {
	euiccPackageSigned {
	eimId #EIM_ID,
	eidValue #EID1,
	counterValue <COUNTER_EIM>,
	euiccPackage psmoList : {
	enable {
	iccid #ICCID_OP_PROF1
	}
	}
	},

	eimSignature <EIM_SIGNATURE> }

EIM_ACKNOWLEDGEMENT_EN1	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	
	"eimAcknowledgements":
	#EIM_ACK_EN1
	
	}

EIM_ACK_EN1	value1 EimAcknowledgements ::= {
	
	<SEQ_NUMBER>,<SEQ_NUMBER>
	
	}

EIM_ACKNOWLEDGEMENT_EN1_DIS2	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	
	"eimAcknowledgements":
	#EIM_ACK_EN1_DIS2
	
	}

EIM_ACK_EN1_DIS2	value1 EimAcknowledgements ::= {
	
	<SEQ_NUMBER>,<SEQ_NUMBER>,<SEQ_NUMBER>
	
	}

EIM_PK_RES_ERR_WRONG_EID	{
	"header" : {
	"functionExecutionStatus" : {
	"status" : "Executed-Success"
	}
	},
	
	"provideEimPackageResultError":
	eidNotFound
	}

D.4 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 VOID

Annex F IUT Settings

F.1 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.

F.2 Platforms Settings

In order to execute the test cases defined in this document, the Platform (eIM) Provider SHALL deliver following settings:

SM-DP+ Setting name	Description
IUT_EIM_ADDRESS	FQDN of the eIM Under Test.
IUT_EIM_ID	Unique identifier of the eIM Under Test. Depending on its setting, it can be an OID, a FQDN or a proprietary identifier.
IUT_TLS_VERSION	Highest TLS protocol version supported by the eIM Under Test, at least v1.2. By versions higher than TLS v1.2 backwards compatibility is assumed.

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 eIM

The eIM SHALL be configured with #CERT_EIM_TLS for NIST and optionally for BRP unless it is specified differently to verify specific configuration (e.g. test cases dedicated for NIST or BRP only).

The eIM provider SHALL provide the capability to provision or prepare eUICC Packages, as required by the specific test cases.

G.2 Device

G.2.1 Device (default)

The Device is “powered on”.

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

The LPAd has access to the root CI key #CERT_CI_ECDSA (or the CI public key) for verification of the TLS certificates of SM-DP+ or SM-DS. No CRL is loaded.

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

G.2.2 Companion Device connected to a Primary Device

The Companion Device is connected to the Primary Device as defined by the Device vendor.

Companion Device and the connected Primary Device are “powered on”.

The Companion Device and Primary Device are in the normal execution mode (NOT in the boot-up mode).

The LPAd of the Companion Device has access to the root CI #CERT_CI_ECDSA (or the CI public key) for verification of the TLS certificates of SM-DP+ or SM-DS. No CRL is loaded.

The Companion Device contains a Test eUICC preconfigured as defined below in G.1.3.

G.2.3 Test eUICC Settings

Depending on the test cases and on the supported options, the Test eUICC SHALL be configured according to the following Initial States.

- The Test eUICC is configured with the ISD-R AID #ISD_R_AID and the EID #EID1.
- The Test eUICC does not contain any Profile.
- The Test eUICC is configured with the default SM-DS address #TEST_ROOT_DS_ADDRESS.
- The Test eUICC contains #TEST_DP_ADDRESS1 as default SM-DP+ address.

The ECASD is configured with at least the following Keys and Certificates based on NIST P-256 [11] or on brainpoolP256r1 [8] for this version of the SGP.23:

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

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

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

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

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

For devices supporting a removable eUICC but not supporting O_D_REMOVABLE_DOWNLOAD_PPR, the Test eUICC can be configured with any RAT.

For devices supporting a non-removable eUICC:

- For some combinations of device options, RAT configurations with certain constraints are required for some sequences, as specified below. These constraints can be satisfied using any valid RAT table; for example, Allowed Operators can be specified explicitly or using wildcards.

Device option(s) supported	RAT configuration of Test eUICC
O_D_EMB_ALLOWS_PPR1_EUC_REQ	PPR1 is allowed and End User Consent is required for #MCC_MNC4 with gid1 and gid2 absent.
O_D_EMB_ALLOWS_PPR2_EUC_REQ	PPR2 is allowed and End User Consent is required for #MCC_MNC2 with gid1 and gid2 absent.
NOT O_D_EMB_ALLOWS_PPR1_EUC_REQ AND O_D_EMB_ALLOWS_PPR1_EUC_NOT_REQ	PPR1 is allowed and End User Consent is not required for #MCC_MNC4 with gid1 and gid2 absent.

NOT O_D_EMB_ALLOWS_PPR2_EUC_REQ AND O_D_EMB_ALLOWS_PPR2_EUC_NOT_REQ	PPR2 is allowed and End User Consent is not required for #MCC_MNC2 with gid1 and gid2 absent.

- If none of the constraints above apply, the Test eUICC can be configured with any RAT.
- Note: in the current version of this document, it is possible to satisfy the relevant constraints above with a single RAT configuration. It is recommended to supply a single device for testing with the RAT configuration satisfying all of the relevant constraints above, rather than to supply multiple devices.

A separate Test eUICC needs to be provided for each additional RAT configuration (not used in this version of the test specification). In case the Test eUICC is non-removable the additional Device SHALL contain the same software and hardware except the Test eUICC configuration.

Annex H VOID

Annex I Document Management

I.1 Document History

Version	Date	CR Number	Brief Description of Change	Approval Authority	Editor / Company
SGP.33-3 v1.2	27 January 2025		Initial version of SGP.33-2 coming from SGP.23 v1.14	ISAG	Yolanda Sanz, GSMA
		NA	Revised version during the eSIMWG3.117		Yolanda Sanz, GSMA
			Following eSIMWG116_2 discussion version updated to v1.2 (note v1.0, v1.1 never published, updated to align numbering with Core Specifications. Integrated CR00001R02		Stephen Packer, GSMA

I.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.