Remote Provisioning Architecture for Embedded UICC Test Specification

Version 3.1

31 May 2016

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

Security Classification: Non-confidential

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

Copyright Notice

Disclaimer

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

Antitrust Notice

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

Table of Contents

1 Introduction

1.1 Overview

The main aim of the GSMA Embedded SIM Remote Provisioning Architecture [1] & [2] is to provide a technical description of the 'over the air' remote provisioning mechanism for machine-to-machine Devices.

This Test Plan provides a set of test cases to be used for testing the implementations of the GSMA Embedded SIM Remote Provisioning Architecture [1] & [2]. This document offers stakeholders a unified test strategy and ensures interoperability between different implementations.

1.2 Scope

This document is intended for:

- Test tools and platforms' suppliers
- Vendors (Device & eUICC Manufacturers)
- Operators

The Test Plan consists of a set of test cases relevant for testing all entities defined in the eUICC remote provisioning ecosystem. The testing scopes developed in this document are:

- Interface compliancy testing
- System behaviour testing

For each test case specified within this Test Plan, there is a reference to one or more requirements.

1.3 Definition of Terms

Term	Description
Actor	Physical entity (person, company or organization) that can assume a Role in the functional architecture. It is possible for an Actor to assume multiple Roles in the same functional architecture.
Connectivity Parameters	A set of data (e.g. SMS-C address) required by the eUICC to open a communication channel (e.g. SMS, HTTPS) on a dedicated network.
Device	Equipment into which an Embedded UICC and a communication module are inserted during assembly. Examples include Utility meter, car and camera.
Disabled (Profile)	The state of a Profile where all files and applications (e.g. NAA) present in the Profile are not selectable over the eUICC - Terminal interface.
Embedded UICC	A UICC which is not easily accessible or replaceable, is not intended to be removed or replaced in the Device, and enables the secure changing of Profiles.
Enabled (Profile)	The state of a Profile when its files and/or applications (e.g. NAA) are selectable over the UICC-Terminal interface.
Executable Load File	An on-card container of one or more application's executable code as defined in GlobalPlatform Card Specification [3].

Term	Description
Executable Module	The on-card executable code of a single application present within an Executable Load File as defined in GlobalPlatform Card Specification [3].
eUICC Certificate	A certificate issued by the EUM for a specific eUICC. This certificate can be verified using the EUM Certificate.
eUICC Manufacturer	Supplier of the eUICCs and resident software (e.g. firmware and operating system).
EUM Certificate	A certificate issued to a GSMA accredited EUM which can be used to verify eUICC Certificates. This certificate can be verified using the Root Certificate.
Integrated Circuit Card ID	Unique number to identify a Profile in an eUICC. Note: the ICCID throughout this specification is used to identify the Profile.
International Mobile Subscriber Identity	Unique identifier owned and issued by Mobile operators to (U)SIM applications to enable Devices to attach to a network and use services.
Issuer Security Domain	A security domain on the UICC as defined by GlobalPlatform Card Specification [3].
Mobile Network Operator	An entity providing access capability and communication services to its Customers through a mobile network infrastructure.
MNO-SD	Security domain part of the Profile, owned by the MNO, providing the Secured Channel to the MNO's OTA Platform. It is used to manage the content of a Profile once the Profile is Enabled.
Network Access Application	An application residing on a UICC which provides authorization to access a network e.g. a USIM application.
OTA Keys	The credentials included in the Profile, used in conjunction with OTA Platforms.
OTA Platform	An MNO platform for remote management of UICCs and the content of Enabled MNO Profiles on eUICCs.
PIX	Proprietary application Identifier eXtension, the value of which is part of the AID.
Platform Management	A set of functions related to the enabling, disabling and deletion of a Profile and the transport of Profile Management functions to an eUICC. Platform Management actions are protected by Platform Management Credentials shared between the SM-SR and the ISD-R. Platform Management does not affect the content of a Profile.
Profile Component	A Profile Component is an element of the Profile and may be one of the following: • An element of the file system like an MF, EF or DF • An Application, including NAA and Security Domain • POL1 • MNO-SD

Term	Description
Profile Element	A Profile Element is a part of the Profile Package representing one or several features of the Profile encoded using TLV structures based on ASN.1 description (as defined in SIMAlliance eUICC Profile Package specification [16]).
Profile Package	A personalised Profile using an interoperable description format transmitted to an eUICC in order to load and install a Profile (as defined in SIMAlliance eUICC Profile Package specification [16]).
Platform Management Credentials	Data required within an eUICC so that a secured communication can be set up between an external entity and the eUICC in order to enable, disable and delete Profiles on the eUICC and to transport Profile Management functions.
Policy	Principles reflected in a set of rules that governs the behaviour of eUICC and/or entities involved in the remote management of the eUICC.
Policy Rule	Defines the atomic action of a Policy and the conditions under which it is executed.
Profile	Combination of a file structure, data and applications to be provisioned onto, or present on, an eUICC and which allows, when Enabled, the access to a specific mobile network infrastructure.
Profile Management	A set of functions related to the downloading, installation and content update of a Profile in a dedicated ISD-P on the eUICC. Download and installation are protected by Profile Management Credentials shared between the SM-DP and the ISD-P.
Profile Management Credentials	Data required within an eUICC so that a Profile downloaded from an external entity can be decrypted and installed on the eUICC.
RID	Registered Application Provider Identifier, the value of which is part of the AID.
Roles	Roles are representing a logical grouping of functions.
Root Certificate	Self-signed certificate of the CI, used to authenticate certificates issued to other entities.
Subscriber	An entity (associated with one or more users) that is engaged in a Subscription with a Telecommunication Service Provider. The Subscriber is allowed to subscribe and unsubscribe to services, to register a user or a list of users authorized to use those services, and also to set the limits relative to the use that associated users make of those services.
Subscription	Describes the commercial relationship between the Subscriber and the Telecommunication Service Provider.
Subscription Address	A unique network address, such as MSISDN, IMSI or SIP-URI, of a mobile Subscription within a mobile network. It is used to route messages, e.g. SMS, to the eUICC.
Subscription Manager Data Preparation	Role that prepares the Profiles and manages the secure download and installation of these Profiles onto the eUICC.
Subscription Manager Secure Routing	Role that securely performs functions of Platform Management commands and the transport of Profile Management commands.

Term	Description
Telecommunication Service Provider	The organization through which the Subscriber obtains PLMN telecommunication services. This is usually the network operator or possibly a separate body.
Test Plan	Current document describing the test cases that allow testing the eUICC Remote Provisioning Architecture.

1.4 Abbreviations

Abbreviation	Description
ADF	Application Dedicated File
AES	Advanced Encryption Standard
AID	Application Identifier
AKA	Authentication and Key Agreement
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation One
ATR	Answer To Reset
ATS	Answer To Select
BIP	Bearer Independent Protocol
C-APDU	Command APDU
CASD	Controlling Authority Security Domain
CAT_TP	Card Application Toolkit Transport Protocol
CERT.DP.ECDSA	Certificate of the SM-DP for its ECDSA key
CERT.ECASD.ECKA	Certificate of the ECASD for its ECKA key
CERT.SR.ECDSA	Certificate of the SM-SR for its ECDSA key
CI	Certificate Issuer
CLA	Class byte of the command message
DER	Distinguished Encoding Rule
DF	Dedicated File
DGI	Data Grouping Identifier
DR	Derivation Random
DS	Device Simulator
ECASD	eUICC Controlling Authority Security Domain
ECDSA	Elliptic Curve cryptography Digital Signature Algorithm
ECKA	Elliptic Curve cryptography Key Agreement algorithm
EF	Elementary File
EID	eUICC-ID
EIS	eUICC Information Set
ePK.DP.ECKA	ephemeral Public Key of the SM-DP used for ECKA
ePK.SR.ECKA	ephemeral Public Key of the SM-SR used for ECKA
eSK.DP.ECKA	ephemeral Private Key of the SM-DP used for ECKA

Abbreviation	Description
eSK.SR.ECKA	ephemeral Private Key of the SM-SR used for ECKA
ETSI	European Telecommunications Standards Institute
eUICC	Embedded UICC
eUICC-UT	eUICC Under Test
EUM	eUICC Manufacturer
EUM-S	eUICC Manufacturer Simulator
EVT	Event
FFS	For Future Study
GSMA	GSM Association
HTTPS	HyperText Transfer Protocol Secure
ICCID	Integrated Circuit Card ID
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
INS	Instruction byte of the command message
ISD	Issuer Security Domain
ISD-P	Issuer Security Domain Profile
ISD-R	Issuer Security Domain Root
ISO	International Organization for Standardization
MAC	Message Authentication Code
MEID	Mobile Equipment IDentifier
MF	Master File
MNO	Mobile Network Operator
MNO-S	MNO Simulator
MSL	Minimum Security Level
NAN	Network Access Name
NPI	Numbering Plan Identifier
OID	Object IDentifier
OTA	Over The Air
P1	Reference control parameter 1
P2	Reference control parameter 2
PDU	Protocol Data Unit
PE	Profile Element
PIN	Personal Identification Number
PIX	Proprietary application Identifier eXtension
PK.CI.ECDSA	Public Key of the CI in the ECASD for verifying certificate signatures
PK.DP.ECDSA	Public Key of the SM-DP, part of the CERT.DP.ECDSA, for verifying his signatures
PK.ECASD.ECKA	Public Key of the ECASD used for ECKA

Abbreviation	Description
PK.SR.ECDSA	Public Key of the SM-SR part of the CERT.SR.ECDSA, for verifying his signatures
PLMN	Public Land Mobile Network
POL1	Policy Rules within the Profile
POL2	Policy Rules associated to a Profile and stored in the relevant EIS at the SM-SR
POR	Proof Of Receipt
PSK	Pre-Shared Key
PUK	PIN Unblocking Key
R-APDU	Response APDU
R-MAC	Response MAC
REQ	Requirement
RFM	Remote File Management
RPS	GSMA Embedded UICC Remote Provisioning messages
SCP	Secure Channel Protocol
SD	Security Domain
SDIN	Security Domain Image Number
SDU	Service Data Unit
ShS	Shared Secret
SIM	Subscriber Identity Module
SIN	Security Domain Provider Identification Number
SK.CI.ECDSA	Private key of the CI for signing certificates
SK.DP.ECDSA	Private Key of the of SM-DP for creating signatures
SK.ECASD.ECKA	Private Key of the ECASD used for ECKA
SK.SR.ECDSA	Private Key of the SM-SR for creating signatures
SM	Subscription Manager
SM-DP	Subscription Manager Data Preparation
SM-DP-S	Subscription Manager Data Preparation Simulator
SM-DP-UT	Subscription Manager Data Preparation Under Test
SMS-C	Short Message Service Centre
SM-SR	Subscription Manager Secure Routing
SM-SR-S	Subscription Manager Secure Routing Simulator
SM-SR-TP	Third Party Subscription Manager Secure Routing
SM-SR-UT	Subscription Manager Secure Routing Under Test
SSD	Supplementary Security Domain
SW	Status Word
TAR	Toolkit Application Reference
TLS	Transport Layer Security

Abbreviation	Description
TLV	Tag, Length, Value
TON	Type Of Number
URI	Uniform Resource Identifier
USIM	Universal Subscriber Identity Module
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1.5 Document Cross-references

Ref	Title
[1]	GSMA Embedded SIM Remote Provisioning Architecture v1.1
[2]	GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification v3.1
[3]	GlobalPlatform Card Specification v.2.2.1
[4]	ETSI TS 102 225 - Secured packet structure for UICC based applications; Release 12
[5]	3GPP TS 23.040 - Technical Specification Group Core Network and Terminals; Technical realization of the Short Message Service (SMS)
[6]	ETSI TS 102 226 - Remote APDU structure for UICC based applications; Release 9
[7]	ETSI TS 102 127 - Transport protocol for CAT applications; Release 6
[8]	RFC 5246 - The TLS Protocol – Version 1.2
[9]	RFC 5487 - Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode
[10]	ISO/IEC 7816-4 - Identification cards – Integrated circuit cards - Part 4: Organization, security and commands for interchange
[11]	GlobalPlatform Card Specification v.2.2 - Amendment D: Secure Channel Protocol 03 v1.1.1
[12]	GlobalPlatform Card Specification v.2.2 - Amendment E: Security Upgrade for Card Content Management v1.0.1
[13]	GlobalPlatform Card Specification v.2.2.1 - UICC Configuration v1.0.1
[14]	GlobalPlatform Card Specification v.2.2 - Amendment C: Contactless Services v1.1.1
[15]	RFC 4346 - The TLS Protocol – Version 1.1
[16]	SIMAlliance eUICC Profile Package: Interoperable Format Technical Specification Version 2.0

1.6 Conventions

Throughout this document, normative requirements are highlighted by use of key words as described below.

The key words "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "MAY" in this document are to be interpreted as follows:

SHALL - This word, or the term "REQUIRED", mean that the definition is a mandatory requirement of the specification.

SHALL NOT - This phrase means that the definition is a mandatory prohibition of the specification.

SHOULD - This word means that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT - This phrase means that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.

MAY - This word mean that an item is truly optional. One supplier may choose to include the item because a particular marketplace requires it or because the supplier feels that it enhances the product while another supplier may omit the same item.

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.
Support	The support columns are to be filled in by the supplier of the implementation. The following common notations are used for the support column: Y supported by the implementation. N not supported 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 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-SR, SM-DP or eUICC Entities under test that take in charge the functions used in the test case.
Applicability	See clause 2.1.3 'Applicability and Notations'.

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 4. Items indicated as O_XYZ (for example, O_HTTPS) refer to features supported by a Role.

Item	Option	Support	Mnemonic
1	Support of HTTPS		O_HTTPS
2	Support of CAT_TP		O_CAT_TP
3	HTTPS enabled on the default MNO-SD		O_MNO_HTTPS
4	Confidential setup of default Profile keys using scenario #2.B supported		O_MNO_SC2B
5	Confidential setup of default Profile keys using scenario #3 supported		O_MNO_SC3

Table 4: Options

All these features are related to the eUICC. As consequence, only the EUM is responsible for stating the support of these features.

Note that O_HTTPS and O_CAT_TP are linked. At least, one of these options shall be supported. The support of the optional feature O_MNO_HTTPS supposes that the O_HTTPS is also supported.

2.1.5 Applicability Table

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

Test case	Name	Roles	Applicability
Interfaces Compliancy Test Cases			
4.2.2.2.1	TC.TP.SMS.1:Transport_SMS	eUICC	M
4.2.2.2.2	TC.TP.CAT_TP.2:Transport_CAT_TP	eUICC	C2
4.2.2.2.3	TC.TP.HTTPS.3:Transport_HTTPS	eUICC	C1
4.2.3.2.1	TC.ES5.CISDP.1:CreateISDP_SMS	eUICC	M
4.2.3.2.2	TC.ES5.CISDP.2:CreateISDP_CAT_TP	eUICC	C2
4.2.3.2.3	TC.ES5.CISDP.3:CreateISDP_HTTPS	eUICC	C1
4.2.4.2.1	TC.ES5.EP.1:EnableProfile_SMS	eUICC	M
4.2.4.2.2	TC.ES5.EP.2:EnableProfile_CAT_TP	eUICC	C2
4.2.4.2.3	TC.ES5.EP.3:EnableProfile_HTTPS	eUICC	C1
4.2.5.2.1	TC.ES5.DISP.1:DisableProfile_SMS	eUICC	M
4.2.5.2.2	TC.ES5.DISP.2:DisableProfile_CAT_TP	eUICC	C2
4.2.5.2.3	TC.ES5.DISP.3:DisableProfile_HTTPS	eUICC	C1
4.2.6.2.1	TC.ES5.FB.1:SetFallbackAttribute_SMS	eUICC	M
4.2.6.2.2	TC.ES5.FB.2:SetFallbackAttribute_CAT_TP	eUICC	C2
4.2.6.2.3	TC.ES5.FB.3:SetFallbackAttribute_HTTPS	eUICC	C1
4.2.7.2.1	TC.ES5.DP.1:DeleteProfile_SMS	eUICC	M
4.2.7.2.2	TC.ES5.DP.2:DeleteProfile_CAT_TP	eUICC	C2
4.2.7.2.3	TC.ES5.DP.3:DeleteProfile_HTTPS	eUICC	C1

Test case	Name	Roles	Applicability
4.2.8.2.1	TC.ES5.ECA.1:eUICCCapabilityAudit_SMS	eUICC	M
4.2.8.2.2	TC.ES5.ECA.2:eUICCCapabilityAudit_CAT_TP	eUICC	C2
4.2.8.2.3	TC.ES5.ECA.3:eUICCCapabilityAudit_HTTPS	eUICC	C1
4.2.9.2.1	TC.ES5.MD.1:MasterDelete_SMS	eUICC	M
4.2.9.2.2	TC.ES5.MD.2:MasterDelete_CAT_TP	eUICC	C2
4.2.9.2.3	TC.ES5.MD.3:MasterDelete_HTTPS	eUICC	C1
4.2.10.2.1	TC.ES5.EISDRK.1:EstablishISDRKeyset_SMS	eUICC	M
4.2.10.2.2	TC.ES5.EISDRK.2:EstablishISDRKeyset_CAT_TP	eUICC	C2
4.2.10.2.3	TC.ES5.EISDRK.3:EstablishISDRKeyset_HTTPS	eUICC	C1
4.2.11.2.1	TC.ES5.FIH.1:FinaliseISDRHandover_SMS Test Sequence N°1	eUICC	C9
4.2.11.2.1	TC.ES5.FIH.1:FinaliseISDRHandover_SMS Test Sequence N°2, Test Sequence N°3	eUICC	M
4.2.11.2.2	TC.ES5.FIH.2:FinaliseISDRHandover_CAT_TP Test Sequence N°1	eUICC	C9
4.2.11.2.2	TC.ES5.FIH.2:FinaliseISDRHandover_CAT_TP Test Sequence N°2	eUICC	C8
4.2.11.2.3	TC.ES5.FIH.3:FinaliseISDRHandover_HTTPS	eUICC	C1
4.2.12.2.1	TC.ES5.USAP.1:UpdateSMSRAddrParam_SMS Test Sequence N°1	eUICC	M
4.2.12.2.1	TC.ES5.USAP.1:UpdateSMSRAddrParam_SMS Test Sequence N°2	eUICC	C10
4.2.12.2.1	TC.ES5.USAP.1:UpdateSMSRAddrParam_SMS Test Sequence N°3	eUICC	C9
4.2.12.2.2	TC.ES5.USAP.2:UpdateSMSRAddrParam_CAT_TP	eUICC	C2
0	TC.ES5.USAP.3:UpdateSMSRAddrParam_HTTPS	eUICC	C1
4.2.13.2.1	TC.ES5.NOTIFPE.1:Notification_SMS	eUICC	M
4.2.13.2.2	TC.ES5.NOTIFPE.2:Notification_CAT_TP	eUICC	C2
0	TC.ES5.NOTIFPE.3:Notification_HTTPS	eUICC	C1
4.2.14.2.1	TC.ES5.NOTIFPD.1:Notification_SMS	eUICC	M
4.2.14.2.2	TC.ES5.NOTIFPD.2:Notification_CAT_TP	eUICC	C2
0	TC.ES5.NOTIFPD.3:Notification_HTTPS	eUICC	C1
4.2.15.2.1	TC.ES6.UPOL1MNO.1:UpdatePOL1byMNO_SMS	eUICC	M
4.2.15.2.2	TC.ES6.UPOL1MNO.2:UpdatePOL1byMNO_CAT_TP	eUICC	C2
4.2.15.2.3	TC.ES6.UPOL1MNO.3:UpdatePOL1byMNO_HTTPS	eUICC	C5
4.2.16.2.1	TC.ES6.UCPMNO.1:UpdateConnectParamByMNO_SMS Test Sequence N°1	eUICC	M
4.2.16.2.1	TC.ES6.UCPMNO.1:UpdateConnectParamByMNO_SMS Test Sequence N°2	eUICC	C3
4.2.16.2.1	TC.ES6.UCPMNO.1:UpdateConnectParamByMNO_SMS Test Sequence N°3	eUICC	C4
4.2.17.2.1	TC.ES8.EISDPK.1:EstablishISDPKeyset_SMS	eUICC	M
4.2.17.2.2	TC.ES8.EISDPK.2:EstablishISDPKeyset_CAT_TP	eUICC	C2
4.2.17.2.3	TC.ES8.EISDPK.3:EstablishISDPKeyset_HTTPS	eUICC	C1
4.2.18.2.1	TC.ES8.DAI.1:DownloadAndInstallation_CAT_TP	eUICC	C2
4.2.18.2.2	TC.ES8.DAI.2:DownloadAndInstallation_HTTPS	eUICC	C1

Test case	Name	Roles	Applicability
4.2.19.2.1	TC.ES8.UCP.1:UpdateConnectivityParameters_SMS Test Sequence N°1	eUICC	M
4.2.19.2.1	TC.ES8.UCP.1:UpdateConnectivityParameters_SMS Test Sequence N°2, Test Sequence N°4	eUICC	C3
4.2.19.2.1	TC.ES8.UCP.1:UpdateConnectivityParameters_SMS Test Sequence N°3, Test Sequence N°5	eUICC	C4
4.2.19.2.2	TC.ES8.UCP.2:UpdateConnectivityParameters_CAT_TP	eUICC	C2
4.2.19.2.3	TC.ES8.UCP.3:UpdateConnectivityParameters_HTTPS	eUICC	C1
4.3.1.2.1	TC.ES1.REIS.1:RegisterEIS	SM-SR	M
4.3.2.2.1	TC.ES2.GEIS.1:GetEIS	SM-DP	M
4.3.3.2.1	TC.ES2.DP.1:DownloadProfile	SM-DP	M
4.3.4.2.1	TC.ES2.UPR.1:UpdatePolicyRules	SM-DP	M
4.3.5.2.1	TC.ES2.USA.1:UpdateSubscriptionAddress	SM-DP	M
4.3.6.2.1	TC.ES2.EP.1:EnableProfile	SM-DP	M
4.3.6.2.2	TC.ES2.EP.2:EnableProfileWithDeletion	SM-DP	M
4.3.7.2.1	TC.ES2.DISP.1:DisableProfile	SM-DP	M
4.3.8.2.1	TC.ES2.DP.1:DeleteProfile	SM-DP	M
4.3.9.2.1	TC.ES3.GEIS.1:GetEIS	SM-SR	M
4.3.10.2.1	TC.ES3.AEIS.1:AuditEIS	SM-SR	M
4.3.11.2.1	TC.ES3.CISDP.1:CreateISDP	SM-SR	M
4.3.12.2.1	TC.ES3.SDATA.1:SendData	SM-SR	M
4.3.13.2.1	TC.ES3.UPR.1:UpdatePolicyRules	SM-SR	M
4.3.14.2.1	TC.ES3.USA.1:UpdateSubscriptionAddress	SM-SR	M
4.3.15.2.1	TC.ES3.UCP.1:UpdateConnectivtyParameters	SM-SR	M
4.3.16.2.1	TC.ES3.EP.1:EnableProfile	SM-SR	M
4.3.17.2.1	TC.ES3.DISP.1:DisableProfile	SM-SR	M
4.3.18.2.1	TC.ES3.DISDP.1:DeleteISDP	SM-SR	M
4.3.19.2.1	TC.ES4.GEIS.1:GetEIS Test Sequence N°1	SM-SR	M
4.3.19.2.1	TC.ES4.GEIS.1:GetEIS Test Sequence N°2	SM-SR	N/A
4.3.20.2.1	TC.ES4.UPR.1:UpdatePolicyRules	SM-SR	M
4.3.21.2.1	TC.ES4.USA.1:UpdateSubscriptionAddress	SM-SR	M
4.3.22.2.1	TC.ES4.AEIS.1:AuditEIS	SM-SR	M
4.3.23.2.1	TC.ES4.EP.1:EnableProfile	SM-SR	M
4.3.24.2.1	TC.ES4.DISP.1:DisableProfile	SM-SR	M
4.3.25.2.1	TC.ES4.DP.1:DeleteProfile	SM-SR	M
4.3.26.2.1	TC.ES4.PSMSRC.1:PrepareSMSRChange	SM-SR	M
4.3.27.2.1	TC.ES4.SMSRC.1:SMSRChange	SM-SR	M
4.3.28.2.1	TC.ES7.HEUICC.1:HandoverEUICC	SM-SR	M
4.3.29.2.1	TC.ES7.ASMSR.1:AuthenticateSMSR	SM-SR	M
System Behaviour Test Cases			
5.2.1.2.1	TC.ECASD.1:EIDRetrieval	eUICC	M
5.2.2.2.1	TC.LOCKISDR.1:LockISDR	eUICC	M
5.2.2.2.2	TC.LOCKISDP.1:LockISDP	eUICC	M

Test case	Name	Roles	Applicability
5.2.3.2.1	TC.CV.1:ComponentVisibility	eUICC	M
5.2.3.2.2	TC.CV.2:ISDRVisibility	eUICC	M
5.2.3.2.3	TC.CV.3:ISDPNotEnabled Test Sequence N°1, Test Sequence N°3	eUICC	C2
5.2.3.2.3	TC.CV.3:ISDPNotEnabled Test Sequence N°2, Test Sequence N°4	eUICC	C1
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°1	eUICC	C2
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°2	eUICC	C1
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°3	eUICC	M
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°1	eUICC	C2
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°2	eUICC	C1
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°3	eUICC	M
5.2.3.2.6	TC.CV.6:MNOSDDefinition	eUICC	M
5.2.4.2.1	TC.SAR.1:SecurityError_SMS	eUICC	M
5.2.4.2.1.2	TC.SAR.2:ISDRResponsibility	eUICC	M
0	TC.SAR.3:ReplayAttack	eUICC	M
5.2.4.2.4	TC.SAR.4:HTTPSRestrictions	eUICC	C1
5.2.4.2.5	TC.SAR.5:SCP03t_ErrorManagement	eUICC	M
5.2.5.2.1	TC.CSMNOSCK.1:Scenario#2.B	eUICC	C6
5.2.5.2.2	TC.CSMNOSCK.2:Scenario#3	eUICC	C7
5.2.6.2.1	TC.FPIP.1:ProfileDownloadAndEnabling Test Sequence N°1	eUICC	C2
5.2.6.2.1	TC.FPIP.1:ProfileDownloadAndEnabling Test Sequence N°2	eUICC	C1
5.3.1.2.1	TC.EUICCIC.1:eUICCEligibilitySMDP	SM-DP	M
5.3.1.2.2	TC.EUICCIC.2:eUICCEligibilitySMSR	SM-SR	M
5.3.2.2.1	TC.PROC.DIP.1:DownloadAndInstallProfile Test Sequence N°1	SM-DP, SM-SR	C3
5.3.2.2.1	TC.PROC.DIP.1:DownloadAndInstallProfile Test Sequence N°2	SM-DP, SM-SR	C4
5.3.2.2.2	TC.PROC.DIP.2:DownloadAndInstallProfileAndEnable	SM-DP, SM-SR	M
5.3.3.2.1	TC.PROC.PE.1.ProfileEnablingByMNO	SM-SR	M
5.3.3.2.2	TC.PROC.PE.2.ProfileEnablingBySMDP	SM-DP, SM-SR	M
5.3.4.2.1	TC.PROC.DIS.1:ProfileDisablingByMNO	SM-SR	M
5.3.4.2.2	TC.PROC.DIS.2:ProfileDisablingBySMDP	SM-DP, SM-SR	M
5.3.5.2.1	TC.PROC.DEL.1:ProfileDeletionByMNO	SM-SR	M
5.3.5.2.2	TC.PROC.DEL.2:ProfileDeletionBySMDP	SM-DP, SM-SR	M

Test case	Name	Roles	Applicability
5.3.7.2.1	TC.PROC.SMSRCH.1:SMSRChange	SM-DP, SM-SR	M
5.3.7.2.2	TC.PROC.SMSRCH.2:SMSRChange	SM-SR	M
5.3.7.2.3	TC.PROC.SMSRCH.3:SMSRChange	SM-SR	M
5.3.7.2.4	TC.PROC.SMSRCH.4:SMSRChange	SM-SR	M
5.3.8.2.1	TC.PROC.UCP.1:UpdateConnectivityParameters Test Sequence N°1	SM-SR	M
5.3.8.2.1	TC.PROC.UCP.1:UpdateConnectivityParameters Test Sequence N°2	SM-SR	C3
5.3.8.2.1	TC.PROC.UCP.1:UpdateConnectivityParameters Test Sequence N°3	SM-SR	C4

Table 5: Applicability of Tests

Conditional item	Condition
C1	IF (NOT O_CAT_TP OR O_HTTPS) THEN M ELSE N/A
C2	IF (NOT O_HTTPS OR O_CAT_TP) THEN M ELSE N/A
C3	IF (O_CAT_TP) THEN M ELSE N/A
C4	IF (O_HTTPS) THEN M ELSE N/A
C5	IF (O_HTTPS AND O_MNO_HTTPS) THEN M ELSE N/A
C6	IF (O_MNO_SC2B) THEN M ELSE N/A
C7	IF (O_MNO_SC3) THEN M ELSE N/A
C8	IF (O_HTTPS AND O_CAT_TP) THEN M ELSE N/A
C9	IF (NOT O_HTTPS) THEN M ELSE N/A
C10	IF (NOT O_CAT_TP) THEN M ELSE N/A

Table 6: Conditional Items Referenced by Table 5

2.2 General Consideration

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 I. 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.

2.2.1 Test Cases Format

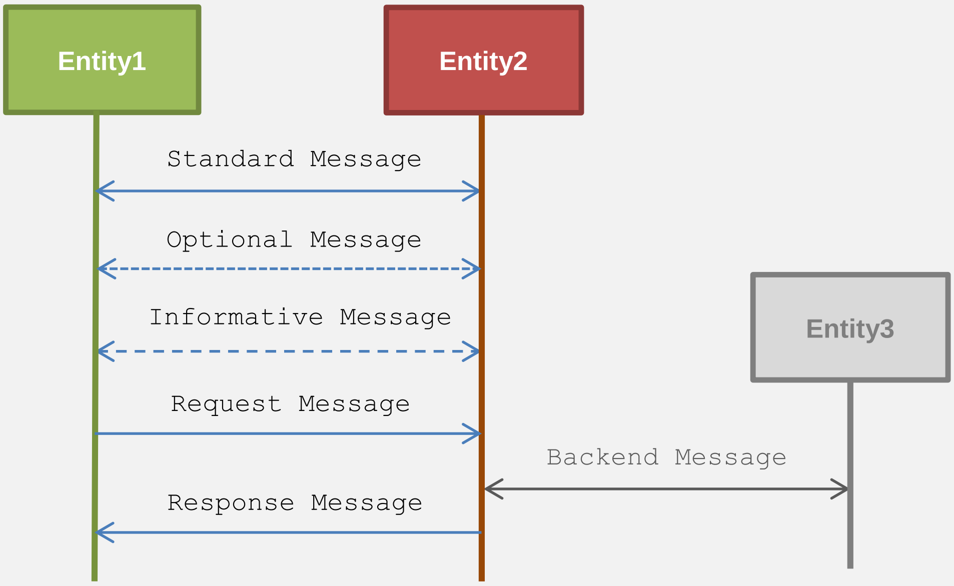
Here is an explanation of the way to define the test cases in chapters 4 and 5.

4.X.Y.Z Test Cases

General Initial Conditions

- Test cases - general condition 1
- Test cases - general condition 2

Test Environment



4.X.Y.Z.1 TC.TEST_NAME.1: TEST_TITLE

Test Purpose

Description of the aim of the test case TC.TEST_NAME.1

Referenced Requirements

- REQ1, REQ2

Initial Conditions

- Test case TC.TEST_NAME.1 - initial condition 1
- Test case TC.TEST_NAME.1 - initial condition 2

4.X.Y.Z.1.1 Test Sequence N°1

Initial Conditions

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

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

2	Entity2 → Entity3	Command or Message to send from Entity2 to Entity3		
Note: Global note for the test sequence N°1				

4.X.Y.Z.1.2 Test Sequence N°2

Initial Conditions

- None

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 (see Note 1)	REQ2
Note 1: Note about the expected result N°2.2				

4.X.Y.Z.2 TC.TEST_NAME.2: TEST_TITLE

…

The test cases TC.TEST_NAME.1:TEST_TITLE and TC.TEST_NAME.2:TEST_TITLE are referenced in Table 5 that allows indicating the applicability of the tests.

The test environment allows describing the different entities involved in the test sequences of the test case. Different types of messages are used:

- standard message: message exchanged between two entities (e.g. an APDU, a RPS Message) composed of a request and a response
- optional message: standard message that may be sent or not depending of the aim of the test
- informative message: message used to facilitate the understanding of the test case. It is not exchanged by any entities (e.g. messages between simulators)
- request message: message sent to an entity that may trigger messages to other entities to generate the corresponding response
- backend message: message exchanged between two entities that cannot be checked by the current test case
- response message: a response related to a request message

In the test case TC.TEST_NAME.1:TEST_TITLE, the requirements REQ1 and REQ2 are respectively covered by the test sequences N°1 and N°2.

The test sequence N°1 shall be executed if and only if these conditions are met:

- Test cases - general condition 1
- Test cases - general condition 2
- Test case TC.TEST_NAME.1 - initial condition 1
- Test case TC.TEST_NAME.1 - initial condition 2
- Test sequence N°1 - initial condition 1
- Test sequence N°1 - initial condition 2

The test sequence N°2 shall be executed if and only if these conditions are met:

- Test cases - general condition 1
- Test cases - general condition 2
- Test case TC.TEST_NAME.1 - initial condition 1
- Test case TC.TEST_NAME.1 - initial condition 2

In the test sequence N°1, 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 I) shall be implemented by the entity under test whatever the test cases to execute.

2.2.2 Using of Methods, Constants and Dynamic Content

In several test sequences described in this document, some methods, constants and dynamic values are used.

A constant is used as follow:
`#NAME_OF_THE_CONSTANT`: shall be replaced by the value of the corresponding constant defined in Annex B.

A dynamic content is described in Annex C and used as follow:
`{NAME_OF_THE_VARIABLE}`
A dynamic content is either generated by an entity under test or by a test tool provider.

A method is used as follow:
`NAME_OF_THE_METHOD(PARAM1, PARAM2…)`: the method and the parameters are described in Annex D.
The implementation of these methods is under the responsibility of the test tool providers.

2.2.3 Commands and Responses

In several test sequences described in this document, some commands and responses are used. These elements are explained in Annex E.

A reference to a command or a response is used as follow:
`[NAME_OF_THE_COMMAND_OR_RESPONSE]`: shall be replaced by the value defined in Annex E.

2.2.4 Referenced Requirements

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

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

2.2.5 Pass Criterion

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 during the steps indicated with a white background in the tables.

A test execution is considered as inconclusive when the pass criteria cannot be evaluated due to issues during the setup of the initial conditions or during the steps indicated with a pink background in the tables.

2.2.6 Future Study

Some of the test cases or test sequences described in this Test Plan are FFS (For Future Study). This means 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

Here are all the interfaces that are tested in this document.

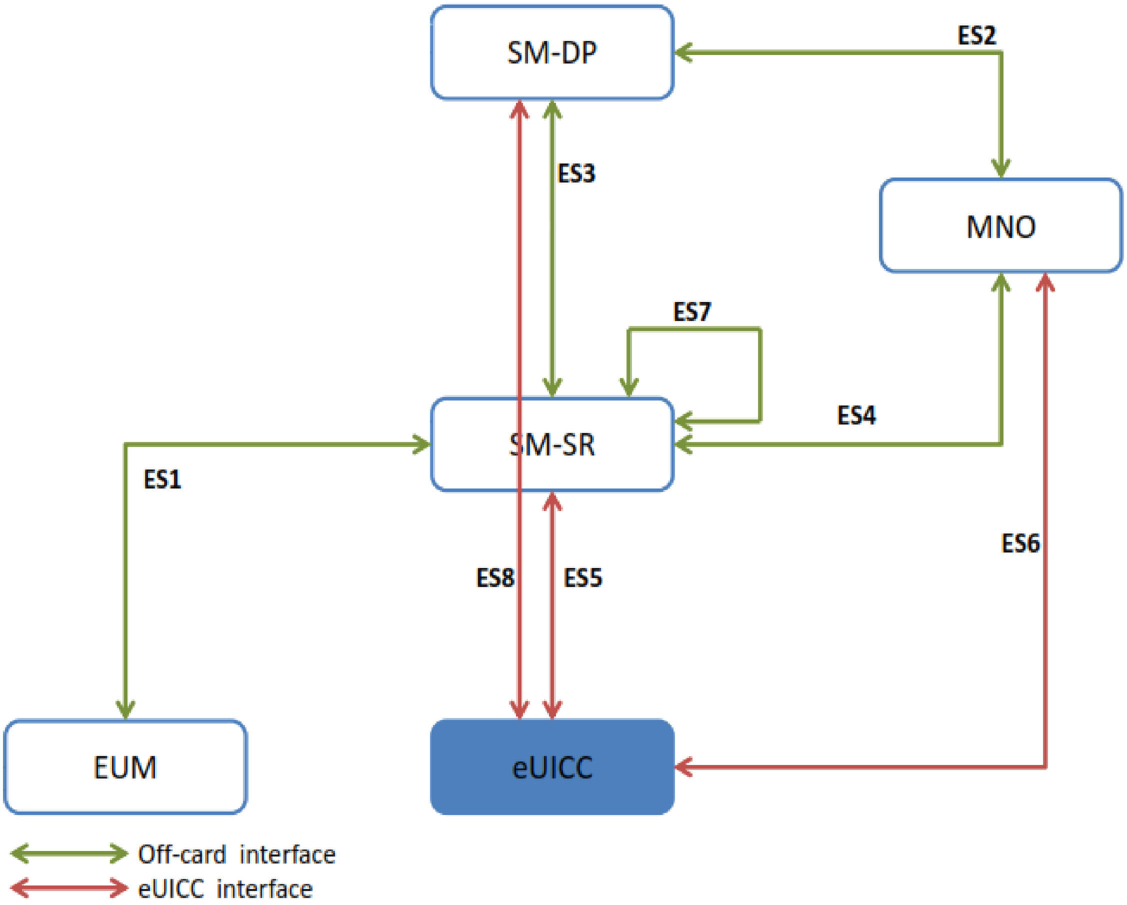


Figure 1: Scope of the Tests

Interface	Description
ES1	Interface between the EUM and the SM-SR that allows the registration of an eUICC within the SM-SR.
ES2	Interface between the MNO and the SM-DP that allows managing a Profile and to trigger Profile loading.
ES3	Interface between the SM-DP and the SM-SR that allows managing a Profile and to trigger Profile loading.
ES4	Interface between the MNO and the SM-SR that allows enabling, disabling and deleting Profiles.
ES5	Interface between the SM-SR and the eUICC that allows the OTA communication.
ES6	Interface between the MNO and the eUICC that allows managing the content of the MNO's Profile.
ES7	Interface between two SM-SR that allows managing the SM-SR change process.
ES8	Interface between the SM-DP and the eUICC that allows downloading of a Profile within the eUICC.

Table 7: Interfaces Descriptions

3.2 Testing Execution

This chapter aims to describe the different testing environments and equipment to allow executing the test cases.
To allow the execution of the different test cases described in this Test Plan, some simulators shall be used. Here are the different simulators that have been defined:

- DS: the Device simulator used to simulate the Device and to send some commands to the eUICC-UT using ISO/IEC 7816-4 [10] on the contact interface. The provisioning commands sent by the DS refer to commands sent by the system Actors (i.e. SM-SR, SM-DP and MNO)
- SM-DP-S: the SM-DP simulator used to simulate the SM-DP and to test a SM-SR
- SM-SR-S: the SM-SR simulator used to simulate the SM-SR and to test a SM-DP or a SM-SR
- MNO-S: the MNO simulator used to simulate the MNO and to test a SM-DP or a SM-SR
- EUM-S: the EUM simulator used to simulate the EUM and to test a SM-SR

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

3.2.1 Interfaces Compliancy

The aim of all the test cases related to the interfaces compliancy (see section 4) is to verify the compliancy of an Actor (i.e. eUICC, SM-DP, SM-SR).

3.2.1.1 eUICC Interfaces

Figure 2 shows the different entities used during the execution of the test cases related to the eUICC interfaces (see section 4.2).

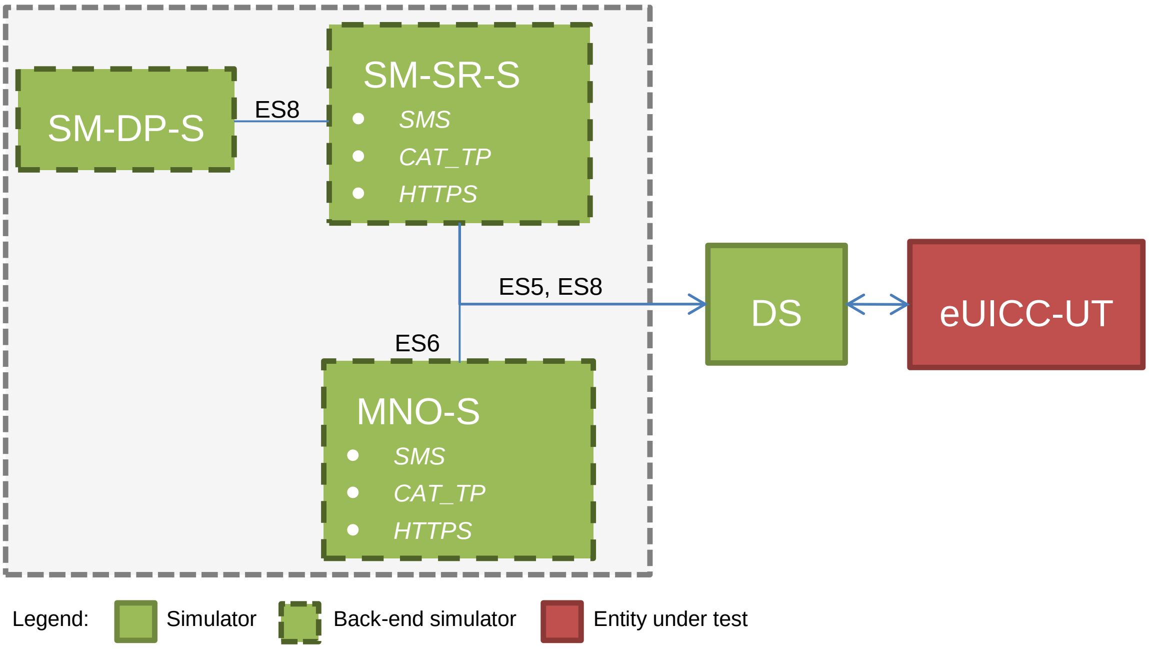


Figure 2: eUICC Interfaces Test Environment

The aim of the interface compliancy test cases, related to the interfaces ES5, ES6 and ES8, is to test the eUICC. The Device Simulator (DS) allows simulating the SM-SR, the SM-DP or the MNO. As consequence, the DS shall include SMS, HTTPS and CAT_TP entities to simulate the OTA communication with the eUICC (i.e. the SM-SR-S, SM-DP-S and MNO-S shall be considered as parts of the DS).

The CAT_TP entity generates CAT_TP PDUs according the Annex G.
The HTTPS entity generates TLS records according the 0.

3.2.1.2 Off-card Interfaces

The off-card test cases assume that all simulated platforms (i.e. EUM-S, MNO1-S, MNO2-S, SM-DP-S, SM-SR-S) identified by EUM_S_ID, MNO1_S_ID, MNO2_S_ID, SM_DP_S_ID, SM_SR_S_ID shall be well known to the platforms under test (i.e. SM-DP-UT, SM-SR-UT) as specified in the initial conditions of each test. All simulated platforms shall be compliant with the security level mandated by the platforms under test.

Figure 3 shows the different entities used during the execution of the test cases related to the off-card interfaces (see section 0).

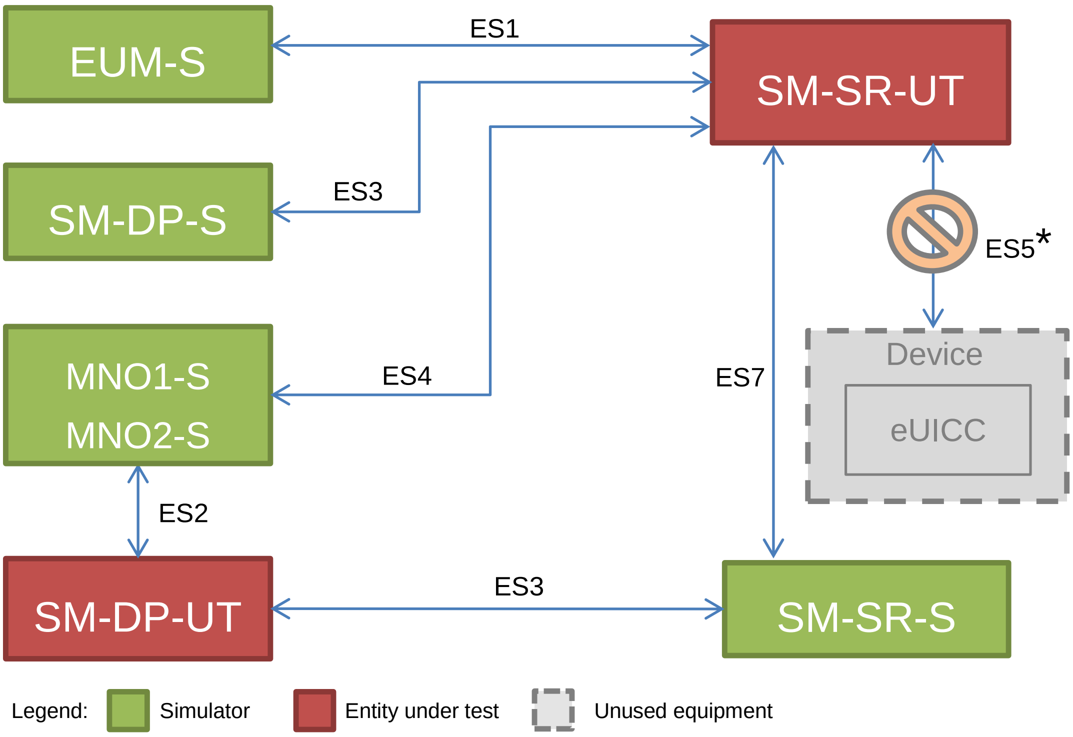


Figure 3: Off-card Interfaces Test Environment

** All OTA interfaces between the SM-SR-UT and an eUICC (ES5 or ES8 over ES5) are out of the scope defined for the off-card interfaces testing. The test cases involving the SM-SR-UT and an eUICC are defined in the section “5 -* System Behaviour Testing*”.*

3.2.2 System Behaviour

The aim of all the test cases related to the system behaviour (see section 5) is to verify the functional behaviour of the eUICC ecosystem composed of the following Actors:

- MNO
- eUICC
- SM-DP
- SM-SR

3.2.2.1 eUICC Behaviour

Figure 4 shows the different entities used during the execution of the test cases related to the eUICC behaviour (see section 5.2).

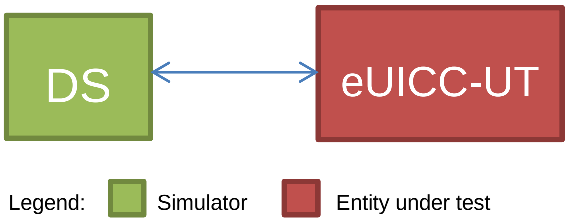


Figure 4: eUICC Behaviour Test Environment

3.2.2.2 Platform Behaviour

Figure 5 shows the different entities used during the execution of the test cases related to the platforms behaviour (see section 5.2.6).

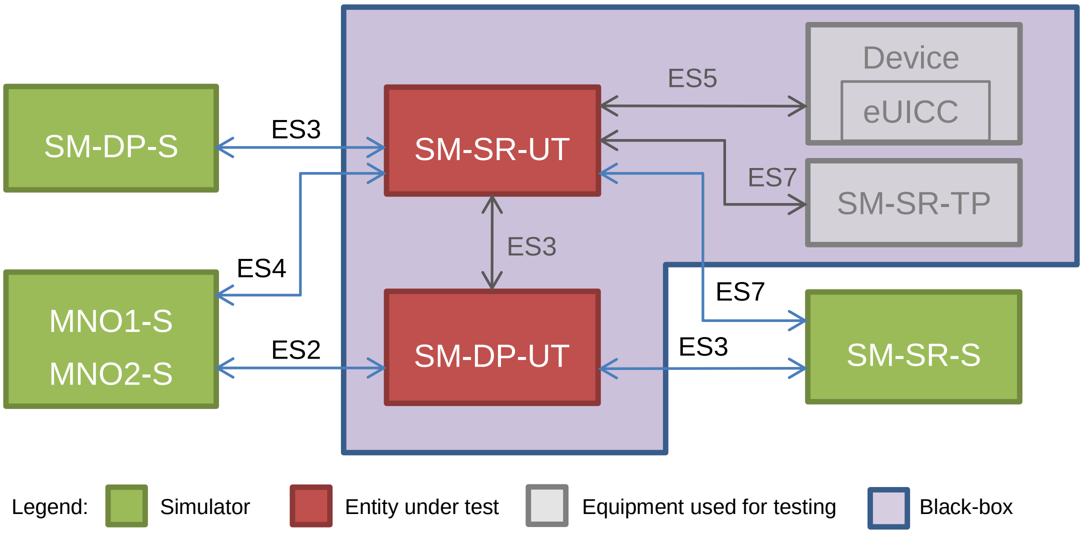


Figure 5: Platform Behaviour Test Environment

A black box testing method is used in order to ensure that the system functional scenarios are properly implemented. In this context, it is assumed that:

- The OTA communication between the SM-SR-UT and the Device equipment (i.e. ES5) shall be based on real wireless network provided by MNO (see Figure 7). OTA operations performed by the SM-SR-UT are not checked by test tool providers: the verification of the correctness of commands coming from the SM-SR-UT is performed by the eUICC/Device.
- The SM-DP-UT and the SM-SR-UT are well known to each other and the functions of the ES3 interface are individually tested in accordance with the test cases described in section 0.
- The Device used for testing shall support all mandatory requirements described in the GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification / Annex G [2].

- The functions of the eUICC interface (i.e. ES5 and ES8 over ES5) shall be supported by the eUICC.
- The entity SM-SR-TP shall be considered as a third party platform used to test the SM-SR-UT. As consequence, the functions of the ES7 interface shall be supported by this platform.

Figure 6 shows the eUICC configuration that shall be used to execute the test cases:

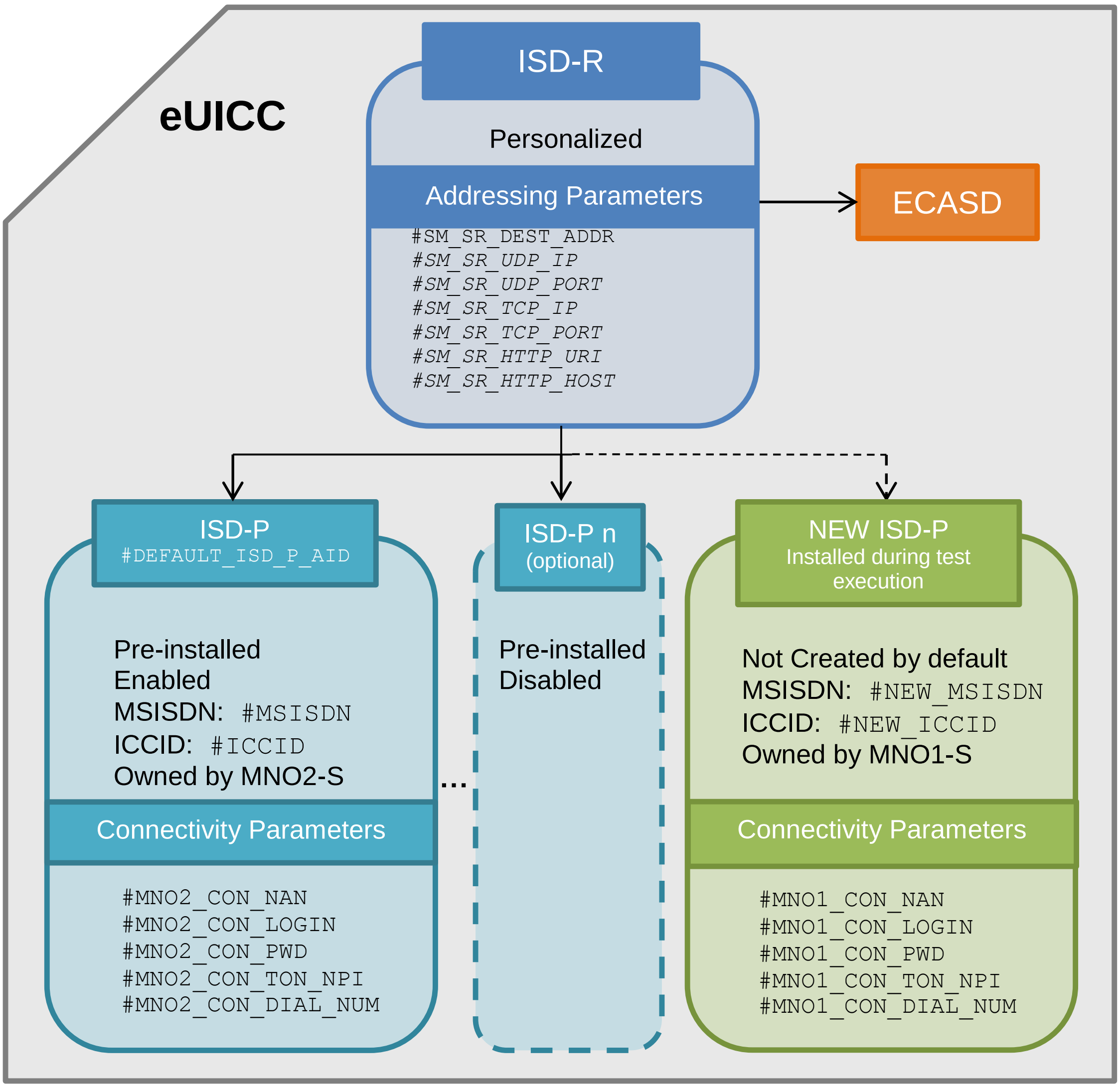


Figure 6: eUICC Configuration

The eUICC, used to execute the test cases defined in the section 5.2.6, shall be compliant with the figure above. A Profile, identified by #ICCID, shall be Enabled. Other pre-installed Profiles may be present (i.e. if present, they shall be Disabled). The Profile, identified by #NEW_ICCID, is dynamically downloaded during the test cases execution: as consequence, it shall not be pre-installed. It is implicitly assumed that all mandatory Profile Components shall be present in the Profiles identified by #ICCID and #NEW_ICCID to allow connectivity network (i.e. file system, NAA…).

Regarding the addressing parameters, except the #SM_SR_DEST_ADDR which is mandatory, the HTTPS and the CAT_TP settings are conditional depending on the eUICC implementation.

Note that the Subscription Addresses of the Profile dynamically downloaded during the tests (i.e. `#NEW_MSISDN` / `#NEW_ICCID`) and the pre-installed Profile (i.e. `#MSISDN` / `#ICCID`) shall be provided by real MNOs (named MNO1 and MNO2 in the Figure 7). It means that the SM-SR-UT is able to communicate with these MNOs' networks (as mentioned in the initial conditions of the test cases defined in section 5.2.6).

In the sections dealing with the platform behaviour testing, MNO1-S and MNO2-S stand for MNO platforms simulators which only allow sending requests to the SM-DP-UT and SM-SR-UT.

Figure 7 shows how the SM-SR-UT shall communicate OTA with the eUICC.

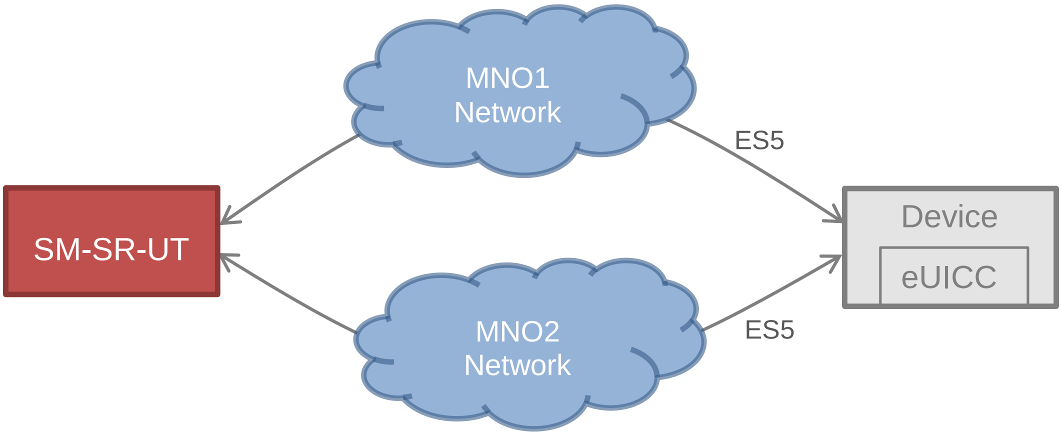


Figure 7: Required Network Access for SM-SR-UT

3.3 Void

4 Interface Compliancy Testing

4.1 General Overview

This section focuses on the implementation of the different interfaces according to the GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2]. The aim is to verify the compliancy of all interfaces within the system.

4.2 eUICC Interfaces

4.2.1 Generic Sub-sequences

This section describes some generic sub-sequences used in the eUICC interfaces compliancy test cases. These test sequences are part of test cases and shall not be executed in standalone mode.

4.2.1.1 Initialization Sequence

To initialize the communication between the DS and the eUICC, these commands shall be executed:

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization SW='9000'	
Note: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.				

4.2.1.2 Open CAT_TP Session on ISD-R

To open a CAT_TP session on the ISD-R, here are the different steps to execute:

Step	Direction	Sequence / Description	Expected result	REQ

1	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[OPEN_CHANNEL_FOR_BIP];`
		`[OPEN_CHANNEL_FOR_CATTP])`
				EUICC_REQ22,
				EUICC_REQ53,
				EUICC_REQ54

2	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
3	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to `#BEARER_DESCRIPTION` 2- The buffer size is equal to `#BUFFER_SIZE` 3- The NAN is equal to `#NAN_VALUE` 4- The port is equal to `#UDP_PORT` 5- The IP is equal to `#IP_VALUE`	EUICC_REQ13, EUICC_REQ18, EUICC_REQ53
5	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The CAT_TP PDU used here after shall be compliant with the Annex G.*				
6	eUICC-UT → DS	SYN	The identification data may contain the `#EID`	EUICC_REQ18
7	DS → eUICC-UT	SYN_ACK		
8	eUICC-UT → DS	ACK_NO_DATA	The CAT_TP session is open.	EUICC_REQ18
9	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The SCP80 status code is equal to '00' – POR OK	EUICC_REQ21
10	DS → eUICC-UT	TERMINAL RESPONSE		

This sub-sequence allows testing these requirements:

- EUICC_REQ13, EUICC_REQ18, EUICC_REQ21, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

4.2.1.3 Open CAT_TP Session on MNO-SD

To open a CAT_TP session on the `#MNO_SD_AID`, here are the different steps to execute:

Step	Direction	Sequence / Description	Expected result	REQ

1	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_SD_TAR,`
		`[OPEN_CHANNEL_FOR_BIP];`
		`[OPEN_CHANNEL_FOR_CATTP])`
		Use `#MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY`
				EUICC_REQ22

Step	Direction	Sequence / Description	Expected result	REQ
2	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
3	DS → eUICC-UT	FETCH		
4	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to `#BEARER_DESCRIPTION` 2- The buffer size is equal to `#BUFFER_SIZE` 3- The NAN is equal to `#NAN_VALUE` 4- The port is equal to `#UDP_PORT` 5- The IP is equal to `#IP_VALUE`	EUICC_REQ13, EUICC_REQ18
5	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The CAT_TP PDU used here after shall be compliant with the Annex G.*				
6	eUICC-UT → DS	SYN		EUICC_REQ18
7	DS → eUICC-UT	SYN_ACK		
8	eUICC-UT → DS	ACK_NO_DATA	The CAT_TP session is open.	EUICC_REQ18
9	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- The SCP80 status code is equal to ‘00’ – POR OK	
10	DS → eUICC-UT	TERMINAL RESPONSE		

This sub-sequence allows testing these requirements:

- EUICC_REQ13, EUICC_REQ18, EUICC_REQ22

4.2.1.4 Close CAT_TP Session

To close a CAT_TP session, here are the different steps to execute:

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RST		EUICC_REQ18
2	eUICC-UT → DS	*PROACTIVE COMMAND:* CLOSE CHANNEL	The CAT_TP session is closed.	EUICC_REQ18
3	DS → eUICC-UT	TERMINAL RESPONSE		

This sub-sequence allows testing this requirement:

- EUICC_REQ18

4.2.1.5 Open HTTPS Session on ISD-R

To open an HTTPS session on the ISD-R, here are the different steps to execute:

Step	Direction	Sequence / Description	Expected result	REQ

1	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[OPEN_SCP81_SESSION])`
				EUICC_REQ22,
				EUICC_REQ42,
				EUICC_REQ54

2	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
3	DS → eUICC-UT	FETCH		
4	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The SCP80 status code is equal to '00' – POR OK	EUICC_REQ21
5	DS → eUICC-UT	TERMINAL RESPONSE		
6	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
7	DS → eUICC-UT	FETCH		
8	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to `#BEARER_DESCRIPTION` 2- The buffer size is equal to `#BUFFER_SIZE` 3- The NAN is equal to `#NAN_VALUE` 4- The port is equal to `#TCP_PORT` 5- The IP is equal to `#IP_VALUE`	EUICC_REQ13, EUICC_REQ14, EUICC_REQ42
9	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The TLS records used here after shall be compliant with the 0.*				
10	eUICC-UT → DS	TLS_CLIENT_HELLO	The CLIENT_HELLO shall contain at least one of the cipher-suites accepted by the HTTPS server.	EUICC_REQ14, EUICC_REQ43
11	DS → eUICC-UT	TLS_SERVER_HELLO and TLS_SERVER_HELLO_DONE		

Step	Direction	Sequence / Description	Expected result	REQ
12	eUICC-UT → DS	TLS_CLIENT_KEY_EXCHANGE and TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED	The CLIENT_KEY_EXCHANGE shall contain the #PSK_ID	EUICC_REQ14, EUICC_REQ43, EUICC_REQ45
13	DS → eUICC-UT	TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED		
14	eUICC-UT → DS	TLS_APPLICATION with the first POST message	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The HTTP content is empty 3- The POST URI is equal to #POST_URI 4- The headers are equal to 1. #HOST 2. #X_ADMIN_PROTOCOL 3. #X_ADMIN_FROM_ISD_R	EUICC_REQ14, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47

This sub-sequence allows testing these requirements:

- EUICC_REQ13, EUICC_REQ14, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ54

4.2.1.6 Open HTTPS Session on MNO-SD

To open an HTTPS session on the #MNO_SD_AID, here are the different steps to execute:

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_SD_TAR, [OPEN_SCP81_MNO_SESSION]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22
2	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
3	DS → eUICC-UT	FETCH		
4	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- The SCP80 status code is equal to ‘00’ – POR OK	

Step	Direction	Sequence / Description	Expected result	REQ
5	DS → eUICC-UT	TERMINAL RESPONSE		
6	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
7	DS → eUICC-UT	FETCH		
8	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to #BEARER_DESCRIPTION 2- The buffer size is equal to #BUFFER_SIZE 3- The NAN is equal to #NAN_VALUE 4- The port is equal to #TCP_PORT 5- The IP is equal to #IP_VALUE	EUICC_REQ13, EUICC_REQ14
9	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The TLS records used here after shall be compliant with the 0.*				
10	eUICC-UT → DS	TLS_CLIENT_HELLO	The CLIENT_HELLO shall contain at least one of the cipher-suites accepted by the HTTPS server.	EUICC_REQ14, EUICC_REQ43
11	DS → eUICC-UT	TLS_SERVER_HELLO and TLS_SERVER_HELLO_DONE		
12	eUICC-UT → DS	TLS_CLIENT_KEY_EXCHANGE and TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED	The CLIENT_KEY_EXCHANGE shall contain the #MNO_PSK_ID	EUICC_REQ14, EUICC_REQ43
13	DS → eUICC-UT	TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED		

Step	Direction	Sequence / Description	Expected result	REQ
14	eUICC-UT → DS	TLS_APPLICATION with the first POST message	1- Decrypt the TLS record with the #MNO_SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The HTTP content is empty 3- The POST URI is equal to #POST_URI 4- The headers are equal to 4. #HOST 5. #X_ADMIN_PROTOCOL 6. #X_ADMIN_FROM_MNO	EUICC_REQ14, EUICC_REQ43

This sub-sequence allows testing these requirements:

- EUICC_REQ13, EUICC_REQ14, EUICC_REQ22, EUICC_REQ43

4.2.1.7 Close HTTPS Session

To close an HTTPS session, here are the different steps to execute:

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	TLS_APPLICATION with the HTTP code equal to #HTTP_CODE_204. The header X-Admin-Protocol shall be present and equal to #X_ADMIN_PROTOCOL.		
2	eUICC-UT → DS	TLS_ALERT_CLOSE_NOTIFY		EUICC_REQ14, EUICC_REQ43
3	eUICC-UT → DS	*PROACTIVE COMMAND:* CLOSE CHANNEL	The HTTP session is closed.	EUICC_REQ14
4	DS → eUICC-UT	TERMINAL RESPONSE		

This sub-sequence allows testing these requirements:

- EUICC_REQ14, EUICC_REQ43

4.2.2 OTA Transport Protocols

4.2.2.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

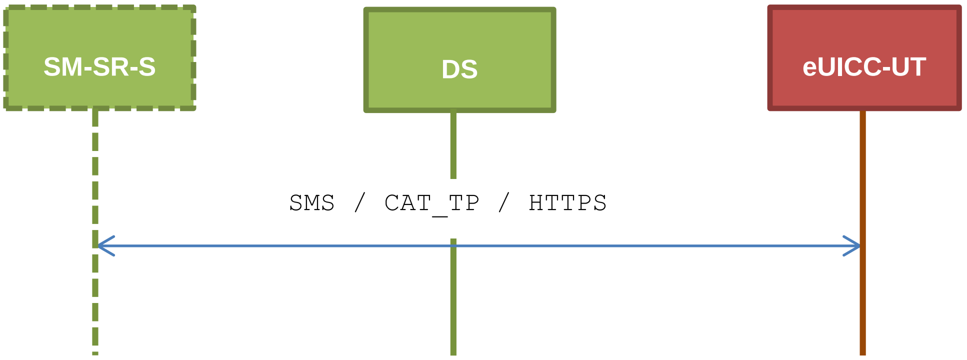
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ53, EUICC_REQ54

4.2.2.2 Test Cases

General Initial Conditions

- None

Test Environment



4.2.2.2.1 TC.TP.SMS.1: Transport_SMS

Test Purpose

To ensure remote application management is possible using SMS. The aim is to send an APDU (GET STATUS) over SMS. The compliance of the GET STATUS response is not verified during these tests.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54

Initial Conditions

- None

4.2.2.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DEFAULT_ISDP])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 7. `#SCP80_AUTH_KEY` 3- The response data is in expanded format with definite length	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.2.2.2 TC.TP.CAT_TP.2: Transport_CAT_TP

Test Purpose

To ensure remote application management is possible using CAT_TP. The aim is to send an APDU (GET STATUS) over CAT_TP. The compliance of the GET STATUS response is not verified during these tests.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ18, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.2.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

3	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DEFAULT_ISDP])`
				EUICC_REQ54

4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using 3. `#SCP80_AUTH_KEY` 4- The response data is in expanded format with definite length	EUICC_REQ13, EUICC_REQ18

Step	Direction	Sequence / Description	Expected result	REQ
5	Close CAT_TP session as described in section 4.2.1.4			

4.2.2.2.3 TC.TP.HTTPS.3: Transport_HTTPS

Test Purpose

To ensure remote application management is possible using HTTPS. The aim is to send an APDU (GET STATUS) command over HTTPS. The compliance of the GET STATUS response is not verified during these tests.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ14, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.2.2.3.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[GET_DEFAULT_ISDP])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 9. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 10. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data in expanded format with indefinite length	EUICC_REQ14, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ48
5	Close HTTPS session as described in section 4.2.1.7			

4.2.3 ES5 (SM-SR – eUICC): CreateISDP

4.2.3.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

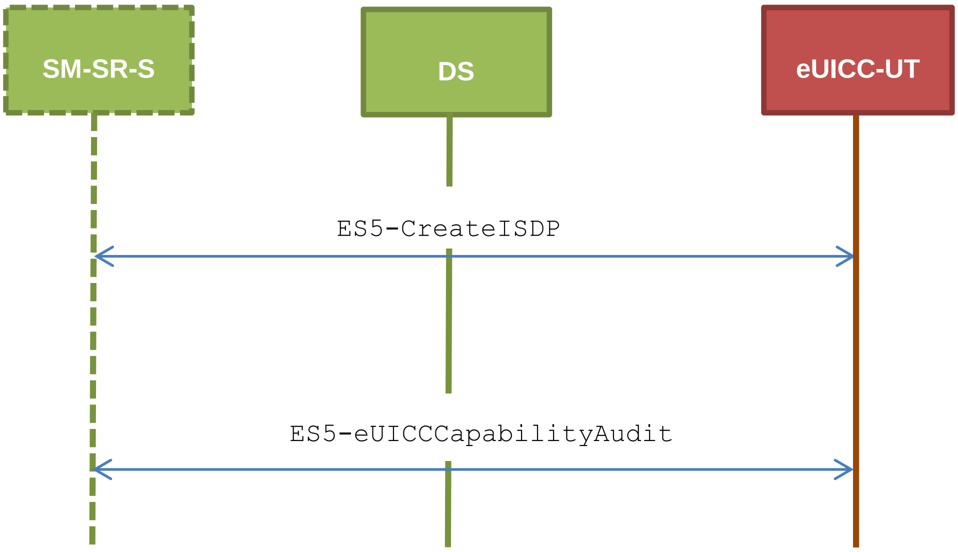
- PF_REQ3, PF_REQ7
- EUICC_REQ4, EUICC_REQ12, EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.3.2 Test Cases

General Initial Conditions

- ISD-P #ISD_P_AID1 not present on the eUICC

Test Environment



4.2.3.2.1 TC.ES5.CISDP.1: CreateISDP_SMS

Test Purpose

To ensure the ISD-P creation process is well implemented on the eUICC using SMS. Several INSTALL commands with different parameters are sent. After ISD-P creation, the lifecycle state of the security domain is checked (shall be SELECTABLE).

Referenced Requirements

- PF_REQ3, PF_REQ7
- EUICC_REQ4, EUICC_REQ12, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ54

Initial Conditions

- None

4.2.3.2.1.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_ISDP])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		

Step	Direction	Sequence / Description	Expected result	REQ
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_009000]`	PF_REQ3, EUICC_REQ12, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_07]`	PF_REQ3, PF_REQ7, EUICC_REQ4, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.3.2.1.2 Test Sequence N°2 - Nominal Case: Memory Quota Set

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_ISDP_MEM])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 1.#SCP80_AUTH_KEY 3- The response data is equal to [R_AB_009000]	PF_REQ3, EUICC_REQ12, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_07]	PF_REQ3, PF_REQ7, EUICC_REQ4, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.3.2.2 TC.ES5.CISDP.2: CreateISDP_CAT_TP

Test Purpose

To ensure the ISD-P creation process is well implemented on the eUICC using CAT_TP. After ISD-P creation, the lifecycle state of the security domain is checked (shall be SELECTABLE).

Referenced Requirements

- PF_REQ3, PF_REQ7
- EUICC_REQ4, EUICC_REQ12, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ23, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.3.2.2.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

3	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_ISDP])`
				EUICC_REQ54

4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `.2.#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_009000]`	PF_REQ3, EUICC_REQ12, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ23

5	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ54

6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `.3.#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_E3_ISDP1_07]`	PF_REQ3, PF_REQ7, EUICC_REQ4, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
7	Close CAT_TP session as described in section 4.2.1.4			

4.2.3.2.3 TC.ES5.CISDP.3: CreateISDP_HTTPS

Test Purpose

To ensure the ISD-P creation process is well implemented on the eUICC using HTTPS. After ISD-P creation, the lifecycle state of the security domain is checked (shall be SELECTABLE).

Referenced Requirements

- PF_REQ3, PF_REQ7
- EUICC_REQ4, EUICC_REQ12, EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.3.2.3.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([INSTALL_ISDP])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 4. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 5. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_009000]	PF_REQ3, EUICC_REQ12, EUICC_REQ14, EUICC_REQ16, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_ISDP1])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 6. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 7. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP1_07]	PF_REQ3, PF_REQ7, EUICC_REQ4, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
7	Close HTTPS session as described in section 4.2.1.7			

4.2.4 ES5 (SM-SR – eUICC): EnableProfile

4.2.4.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

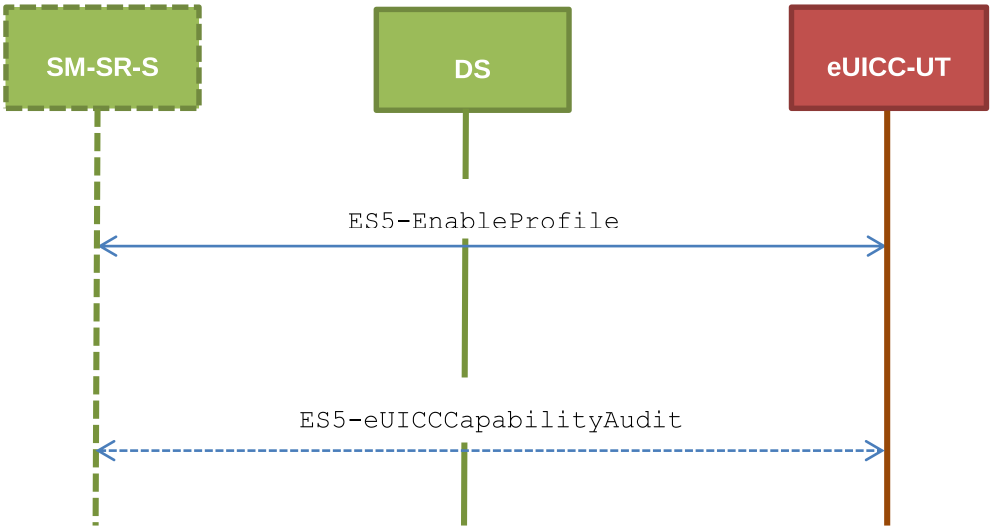
- PF_REQ4, PF_REQ7
- SEC_REQ14
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.4.2 Test Cases

General Initial Conditions

- #ISD_P_AID1 present on the eUICC
- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

Test Environment



4.2.4.2.1 TC.ES5.EP.1: EnableProfile_SMS

Test Purpose

To ensure the Profile enabling process is well implemented on the eUICC using SMS. Some error cases due to incompatible initial conditions are also defined. In these error cases, the lifecycle state of the corresponding ISD-P is checked to make sure that it remains unchanged.

Note: As the update of the lifecycle states of the Profiles may become effective after the REFRESH command, the check of the lifecycle states cannot be performed in this test case.

Referenced Requirements

- PF_REQ4, PF_REQ7
- SEC_REQ14
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54

Initial Conditions

- None

4.2.4.2.1.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- #ISD_P_AID1 in Disabled state
- No POL1 is defined on the #DEFAULT_ISD_P_AID

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [ENABLE_ISDP1])		EUICC_REQ22, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 18. #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_9000]	PF_REQ4, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE		
7	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 1	
8	DS → eUICC-UT	FETCH		
9	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PF_REQ4
10	DS → eUICC-UT	RESET	ATR returned by eUICC	
Note 1: Before sending the REFRESH command, the eUICC may wait for several STATUS events. In this case, the eUICC shall issue the REFRESH command within a maximum time interval of 10 STATUS events.				

4.2.4.2.1.2 Test Sequence N°2 - Error Case: ISD-P Not Disabled

Initial Conditions

- #ISD_P_AID1 in SELECTABLE state
- No POL1 is defined on the #DEFAULT_ISD_P_AID

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [ENABLE_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 19. `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6985]`	PF_REQ4, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_07]`	PF_REQ4, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.4.2.1.3 Test Sequence N°3 - Error Case: ISD-P with Incompatible POL1

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- `#DEFAULT_ISD_P_AID` contains the POL1 "Disabling of the Profile not allowed"

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[ENABLE_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 20. `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_69E1]`	PF_REQ4, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, SEC_REQ14
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_1F]`	PF_REQ4, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.4.2.2 TC.ES5.EP.2: EnableProfile_CAT_TP

Test Purpose

To ensure the Profile enabling process is well implemented on the eUICC using CAT_TP.

Note: As the update of the lifecycle states of the Profiles may become effective after the REFRESH command, the check of the lifecycle states cannot be performed in this test case.

Referenced Requirements

- PF_REQ4
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.4.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- No POL1 is defined on the `#DEFAULT_ISD_P_AID`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

3	DS → eUICC-UT	ACK_DATA containing the result of `SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[ENABLE_ISDP1])`
				EUICC_REQ54

4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using 21.`#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_9000]`	PF_REQ4, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
5	Close CAT_TP session as described in section 4.2.1.4 see Note 1			
6	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 2	
7	DS → eUICC-UT	FETCH		
8	eUICC-UT → DS	*PROACTIVE COMMAND: REFRESH*		PF_REQ4
9	DS → eUICC-UT	RESET	ATR returned by eUICC	
Note 1: The closing of the CAT_TP session may be performed automatically by the eUICC by sending the RST. *Note 2: Before sending the REFRESH command, the eUICC may wait for several STATUS events. In this case, the eUICC shall issue the REFRESH command within a maximum time interval of 10 STATUS events.*				

4.2.4.2.3 TC.ES5.EP.3: EnableProfile_HTTPS

Test Purpose

To ensure the Profile enabling process is well implemented on the eUICC using HTTPS.

Note: As the update of the lifecycle states of the Profiles may become effective after the REFRESH command, the check of the lifecycle states cannot be performed in this test case.

Referenced Requirements

- PF_REQ4
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: #PSK_ID
    - PSK value: #SCP81_PSK

4.2.4.2.3.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- #ISD_P_AID1 in Disabled state
- No POL1 is defined on the #DEFAULT_ISD_P_AID

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([ENABLE_ISDP1])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 2. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 3. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_9000]	PF_REQ4, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
5	Close HTTPS session as described in section 4.2.1.7 see Note 1			
6	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 2	
7	DS → eUICC-UT	FETCH		
8	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PF_REQ4

Step	Direction	Sequence / Description	Expected result	REQ
9	DS → eUICC-UT	RESET	ATR returned by eUICC	
Note 1: The closing of the HTTPS session may be performed automatically by the eUICC by sending the TLS_ALERT_CLOSE_NOTIFY *Note 2: Before sending the REFRESH command, the eUICC may wait for several STATUS events. In this case, the eUICC shall issue the REFRESH command within a maximum time interval of 10 STATUS events.*				

4.2.5 ES5 (SM-SR – eUICC): DisableProfile

4.2.5.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

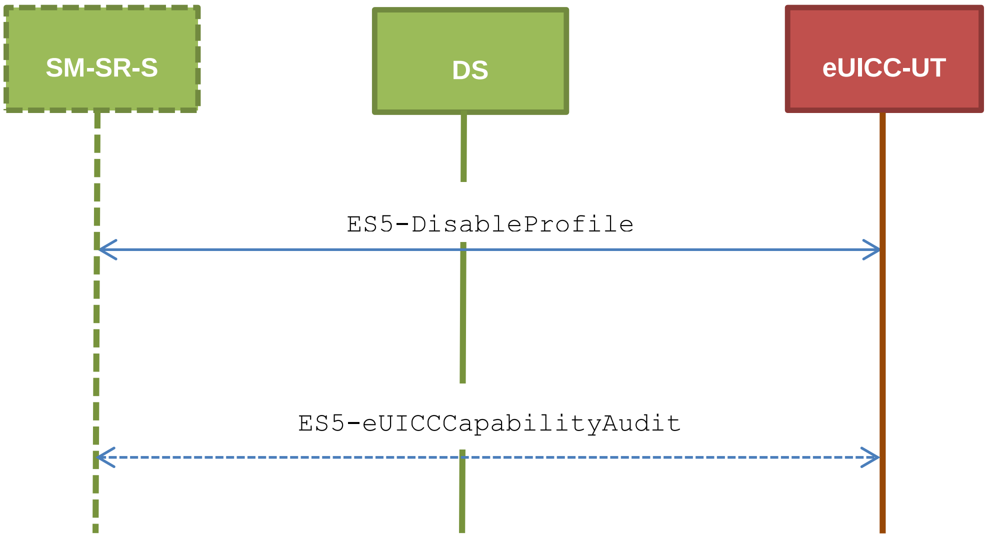
- PF_REQ5, PF_REQ7
- SEC_REQ14
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.5.2 Test Cases

General Initial Conditions

- None

Test Environment



4.2.5.2.1 TC.ES5.DISP.1: DisableProfile_SMS

Test Purpose

To ensure the Profile disabling process is well implemented on the eUICC using SMS. Some error cases due to incompatible initial conditions are also defined. In these error cases, the lifecycle state of the corresponding ISD-P is checked to make sure that it remains unchanged.

Note: As the update of the lifecycle states of the Profiles may become effective after the REFRESH command, the check of the lifecycle states cannot be performed in this test case.

Referenced Requirements

- PF_REQ5, PF_REQ7
- SEC_REQ14
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54

Initial Conditions

- `#ISD_P_AID1` present on the eUICC

4.2.5.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#ISD_P_AID1` in Enabled state
- `#DEFAULT_ISD_P_AID` in Disabled state
- No POL1 is defined on the `#ISD_P_AID1`
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [DISABLE_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 24.#SCP80_AUTH_KEY 3- The response data is equal to [R_AB_9000]	PF_REQ5, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE		
7	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 1	
8	DS → eUICC-UT	FETCH		
9	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PF_REQ5
10	DS → eUICC-UT	RESET	ATR returned by eUICC	
Note 1: Before sending the REFRESH command, the eUICC may wait for several STATUS events. In this case, the eUICC shall issue the REFRESH command within a maximum time interval of 10 STATUS events.				

4.2.5.2.1.2 Test Sequence N°2 – Error Case: ISD-P Not Enabled

Initial Conditions

- #ISD_P_AID1 in SELECTABLE state
- #DEFAULT_ISD_P_AID in Enabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [DISABLE_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		

Step	Direction	Sequence / Description	Expected result	REQ
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 25. #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6985]	PF_REQ5, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_07]	PF_REQ5, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.5.2.1.3 Test Sequence N°3 – Error Case: ISD-P with the Fall-back Attribute Set

Initial Conditions

- #ISD_P_AID1 in Enabled state
- #DEFAULT_ISD_P_AID in Disabled state
- No POL1 is defined on the #ISD_P_AID1
- #ISD_P_AID1 is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [DISABLE_ISDP1])		EUICC_REQ22, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 26. `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6985]`	PF_REQ5, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22, EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_3F]`	PF_REQ5, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.5.2.1.4 Test Sequence N°4 – Error Case: ISD-P with Incompatible POL1

Initial Conditions

- `#ISD_P_AID1` in Enabled state
- `#DEFAULT_ISD_P_AID` in Disabled state
- `#ISD_P_AID1` contains the POL1 "Disabling of the Profile not allowed"
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[DISABLE_ISDP1])`
				EUICC_REQ22, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 27. `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_69E1]`	PF_REQ5, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, SEC_REQ14
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22, EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_3F]`	PF_REQ5, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.5.2.2 TC.ES5.DISP.2: DisableProfile_CAT_TP

Test Purpose

To ensure the Profile disabling process is well implemented on the eUICC using CAT_TP.

Note: As the update of the lifecycle states of the Profiles may become effective after the REFRESH command, the check of the lifecycle states cannot be performed in this test case.

Referenced Requirements

- PF_REQ5
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.5.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#ISD_P_AID1` in Enabled state
- `#DEFAULT_ISD_P_AID` in Disabled state
- No POL1 is defined on the `#ISD_P_AID1`
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [DISABLE_ISDP1])		EUICC_REQ54
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 28.#SCP80_AUTH_KEY 4- The response data is equal to [R_AB_9000]	PF_REQ5, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
5	Close CAT_TP session as described in section 4.2.1.4 see Note 1			
6	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 2	
7	DS → eUICC-UT	FETCH		
8	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PF_REQ5
9	DS → eUICC-UT	RESET	ATR returned by eUICC	
Note 1: The closing of the CAT_TP session may be performed automatically by the eUICC by sending the RST. *Note 2: Before sending the REFRESH command, the eUICC may wait for several STATUS events. In this case, the eUICC shall issue the REFRESH command within a maximum time interval of 10 STATUS events.*				

4.2.5.2.3 TC.ES5.DISP.3: DisableProfile_HTTPS

Test Purpose

To ensure the Profile disabling process is well implemented on the eUICC using HTTPS.

Note: As the update of the lifecycle states of the Profiles may become effective after the REFRESH command, the check of the lifecycle states cannot be performed in this test case.

Referenced Requirements

- PF_REQ5
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.5.2.3.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#ISD_P_AID1` in Enabled state
- `#DEFAULT_ISD_P_AID` in Disabled state
- No POL1 is defined on the `#ISD_P_AID1`
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[DISABLE_ISDP1])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ

4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			29. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			30. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_9000]`
				PF_REQ5,
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ43,
				EUICC_REQ46,
				EUICC_REQ47,
				EUICC_REQ48,
				EUICC_REQ52

5	Close HTTPS session as described in section 4.2.1.7 see Note 1			
6	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 2	
7	DS → eUICC-UT	FETCH		
8	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PF_REQ5
9	DS → eUICC-UT	RESET	ATR returned by eUICC	
Note 1: The closing of the HTTPS session may be performed automatically by the eUICC by sending the TLS_ALERT_CLOSE_NOTIFY. *Note 2: Before sending the REFRESH command, the eUICC may wait for several STATUS events. In this case, the eUICC shall issue the REFRESH command within a maximum time interval of 10 STATUS events.*				

4.2.6 ES5 (SM-SR – eUICC): SetFallbackAttribute

4.2.6.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

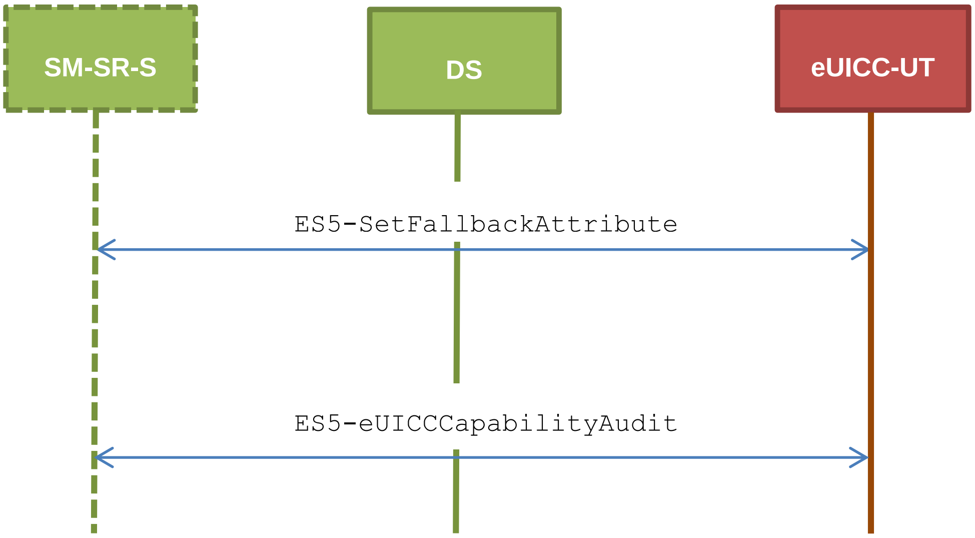
- PF_REQ7, PF_REQ9
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.6.2 Test Cases

General Initial Conditions

- #ISD_P_AID1 present on the eUICC
- #ISD_P_AID1 in Disabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute Set

Test Environment



4.2.6.2.1 TC.ES5.FB.1: SetFallbackAttribute_SMS

Test Purpose

To ensure it is possible to set the Fall-back Attribute on the eUICC using SMS. After changing the security domain with the Fall-back Attribute, a GET STATUS command is sent to make sure that the attribute is set on the targeted ISD-P.

Referenced Requirements

- PF_REQ7, PF_REQ9
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54

Initial Conditions

- None

4.2.6.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[SET_FALLBACK])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 31. `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_9000]`	PF_REQ9, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_FALLBACK])`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_E1]`	PF_REQ7, PF_REQ9, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.6.2.2 TC.ES5.FB.2: SetFallbackAttribute_CAT_TP

Test Purpose

To ensure it is possible to set the Fall-back Attribute on the eUICC using CAT_TP. After changing the security domain with the Fall-back Attribute, a GET STATUS command is sent to make sure that the attribute is set on the targeted ISD-P.

Referenced Requirements

- PF_REQ7, PF_REQ9

- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.6.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [SET_FALLBACK])		EUICC_REQ54
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 32. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_9000]	PF_REQ9, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
5	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_FALLBACK])		EUICC_REQ54
6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 33. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E3_ISDP1_E1]	PF_REQ7, PF_REQ9, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
7	Close CAT_TP session as described in section 4.2.1.4			

4.2.6.2.3 TC.ES5.FB.3: SetFallbackAttribute_HTTPS

Test Purpose

To ensure it is possible to set the Fall-back Attribute on the eUICC using HTTPS. After changing the security domain with the Fall-back attribute, a GET STATUS command is sent to make sure that the attribute is set on the targeted ISD-P.

Referenced Requirements

- PF_REQ7, PF_REQ9
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.6.2.3.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[SET_FALLBACK])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 34. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 35. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_9000]	PF_REQ9, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_FALLBACK])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 36. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 37. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP1_E1]	PF_REQ7, PF_REQ9, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
7	Close HTTPS session as described in section 4.2.1.7			

4.2.7 ES5 (SM-SR – eUICC): DeleteProfile

4.2.7.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

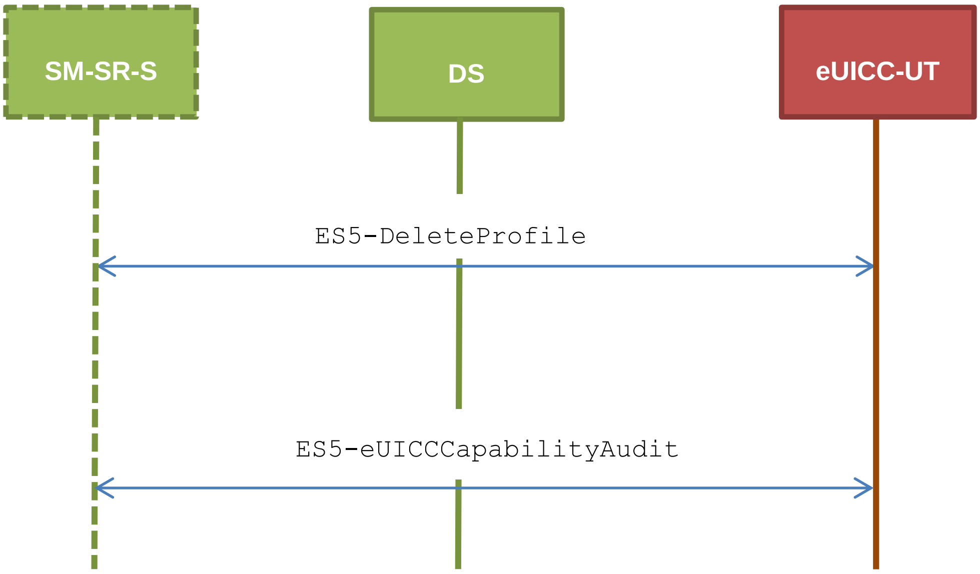
- PF_REQ6, PF_REQ7
- SEC_REQ12, SEC_REQ14
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.7.2 Test Cases

General Initial Conditions

- `#ISD_P_AID1` present on the eUICC

Test Environment



4.2.7.2.1 TC.ES5.DP.1: DeleteProfile_SMS

Test Purpose

To ensure the Profile deletion process is well implemented on the eUICC using SMS. After ISD-P deletion, a GET STATUS command is sent to make sure that the security domain is no longer present on the eUICC. Some error cases due to incompatible initial conditions are also defined.

Referenced Requirements

- PF_REQ6, PF_REQ7
- SEC_REQ12, SEC_REQ14
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54

Initial Conditions

- None

4.2.7.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- #ISD_P_AID1 in Disabled state
- No POL1 defined on #ISD_P_AID1
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [DELETE_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_009000]	PF_REQ6, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6A88]	PF_REQ6, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, SEC_REQ12
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.7.2.1.2 Test Sequence N°2 – Error Case: ISD-P Not Disabled

Initial Conditions

- `#ISD_P_AID1` in Enabled state
- No POL1 defined on `#ISD_P_AID1`
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[DELETE_ISDP1])`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6985]`	PF_REQ6, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22, EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_3F]`	PF_REQ6, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.7.2.1.3 Test Sequence N°3 – Error Case: ISD-P with the Fall-back Attribute Set

Initial Conditions

- #ISD_P_AID1 in Disabled state
- No POL1 defined on #ISD_P_AID1
- #ISD_P_AID1 is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [DELETE_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6985]	PF_REQ6, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_1F]	PF_REQ6, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.7.2.1.4 Test Sequence N°4 – Error Case: ISD-P with Incompatible POL1

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- `#ISD_P_AID1` contains the POL1 “Deletion of the Profile not allowed”
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[DELETE_ISDP1])`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_69E1]`	PF_REQ6, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, SEC_REQ14
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22, EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_1F]`	PF_REQ6, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.7.2.2 TC.ES5.DP.2: DeleteProfile_CAT_TP

Test Purpose

To ensure the Profile deletion process is well implemented on the eUICC using CAT_TP. After ISD-P deletion, a GET STATUS command is sent to make sure that the security domain is no longer present on the eUICC.

Referenced Requirements

- PF_REQ6, PF_REQ7
- SEC_REQ12
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.7.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- No POL1 defined on `#ISD_P_AID1`
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

3	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[DELETE_ISDP1])`
				EUICC_REQ54

4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using 38. `#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_009000]`	PF_REQ6, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18

5	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 39. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_6A88]	PF_REQ6, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, SEC_REQ12
7	Close CAT_TP session as described in section 4.2.1.4			

4.2.7.2.3 TC.ES5.DP.3: DeleteProfile_HTTPS

Test Purpose

To ensure the Profile deletion process is well implemented on the eUICC using HTTPS. After ISD-P deletion, a GET STATUS command is sent to make sure that the security domain is no longer present on the eUICC.

Referenced Requirements

- PF_REQ6, PF_REQ7
- SEC_REQ12
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: #PSK_ID
    - PSK value: #SCP81_PSK

4.2.7.2.3.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- #ISD_P_AID1 in Disabled state
- No POL1 is defined on the #ISD_P_AID1
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

Step	Direction	Sequence / Description	Expected result	REQ
3	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([DELETE_ISDP1])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 10. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 11. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_009000]	PF_REQ6, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_ISDP1])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 12. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 13. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_6A88]	PF_REQ6, PF_REQ7, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, SEC_REQ12
7	Close HTTPS session as described in section 4.2.1.7			

4.2.8 ES5 (SM-SR – eUICC): eUICCCapabilityAudit

4.2.8.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

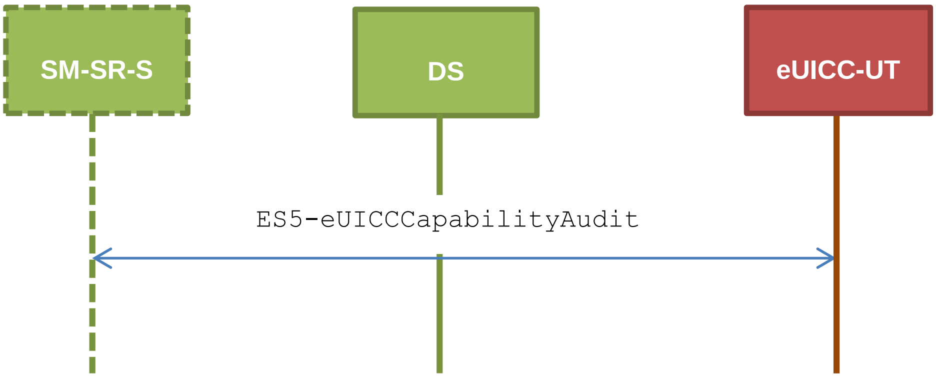
- PF_REQ7
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.8.2 Test Cases

General Initial Conditions

- None

Test Environment



4.2.8.2.1 TC.ES5.ECA.1: eUICCCapabilityAudit_SMS

Test Purpose

To ensure it is possible to audit the eUICC using SMS. GET STATUS and GET DATA commands are sent to retrieve the ISD-P list, the ECASD certificate, the eUICC recognition data and the card resources information.

Referenced Requirements

- PF_REQ7
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54

Initial Conditions

- None

4.2.8.2.1.1 Test Sequence N°1 – Nominal Case: Retrieve all ISD-P

Initial Conditions

- `#ISD_P_AID1` in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_LIST])`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP_LIST3]` 44. (see Note 1)	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: If more than one Profile is pre-installed on the eUICC, this response shall be adapted in consequence (in addition of the Enabled ISD-P identified by the AID `#DEFAULT_ISD_P_AID` and the ISD-P identified by the AID `#ISD_P_AID1`, other Profiles may be present).				

4.2.8.2.1.2 Test Sequence N°2 – Nominal Case: Retrieve Default Enabled ISD-P

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_ENABLED])`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP_3F]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.8.2.1.3 Test Sequence N°3 – Nominal Case: Retrieve Disabled ISD-P

Initial Conditions

- `#ISD_P_AID1` in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_DISABLED])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2-Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_1F]` 45. (see Note 1)	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: If more than one Profile is pre-installed on the eUICC (i.e. several Disabled Profiles exist), this response shall be adapted in consequence (in addition of the ISD-P identified by the AID `#ISD_P_AID1`, other Profiles may be present).				

4.2.8.2.1.4 Test Sequence N°4 – Nominal Case: Retrieve Card Resources Information

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_FF21])`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_FF21]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.8.2.1.5 Test Sequence N°5 – Nominal Case: Retrieve ECASD Recognition Data

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_BF30_REC])`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_BF30_REC]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.8.2.1.6 Test Sequence N°6 – Nominal Case: Retrieve ECASD Certificate Store

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_BF30_CERT])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 16. #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_BF30_ECASD] 4- The #PK_ECASD_ECKA is equal to the content of the TAG '7F49' 5- The signature (i.e. TAG '5F37') shall be verified using the #EUM_PK_ECDSA 6- TAG '95' is equal to #KEY_USAGE 7- TAG '73' contains the TLV 'C0', 'C1' and 'C2' 8- TAG '5F20' contains the #EID	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.8.2.1.7 Test Sequence N°7 – Nominal Case: Retrieve ISD-P with Memory Information

Initial Conditions

- `#ISD_P_AID1` in SELECTABLE state and created using the command `[INSTALL_ISDP_MEM]`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1_MEM])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2-Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_MEM]	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.8.2.1.8 Void

4.2.8.2.2 TC.ES5.ECA.2: eUICCCapabilityAudit_CAT_TP

Test Purpose

To ensure it is possible to audit the eUICC using CAT_TP. GET STATUS and GET DATA commands are sent to retrieve the ISD-P list, the ECASD certificate, the eUICC recognition data and the card resources information.

Referenced Requirements

- PF_REQ7
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.8.2.2.1 Test Sequence N°1 – Nominal Case: Retrieve all Information

Initial Conditions

- #ISD_P_AID1 in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

Step	Direction	Sequence / Description	Expected result	REQ
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_LIST])		EUICC_REQ54
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 47. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E3_ISDP_LIST3] 48. (see Note 1)	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
5	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED])		EUICC_REQ54
6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 49. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E3_ISDP_3F]	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
7	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_DISABLED])		EUICC_REQ54
8	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 50. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E3_ISDP1_1F] 51. (see Note 2)	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18

Step	Direction	Sequence / Description	Expected result	REQ

9	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_FF21])`
				EUICC_REQ54

10	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `52.#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_FF21]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18

11	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_BF30_REC])`
				EUICC_REQ54

12	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `53.#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_BF30_REC]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18

13	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_BF30_CERT])`
				EUICC_REQ54

14	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `54.#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_BF30_ECASD]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
15	Close CAT_TP session as described in section 4.2.1.4			

Step	Direction	Sequence / Description	Expected result	REQ
Note 1: If more than one Profile is pre-installed on the eUICC, this response shall be adapted in consequence (in addition of the Enabled ISD-P identified by the AID `#DEFAULT_ISD_P_AID` and the ISD-P identified by the AID `#ISD_P_AID1`, other Profiles may be present). *Note 2: If more than one Profile is pre-installed on the eUICC (i.e. several Disabled Profiles exist), this response shall be adapted in consequence (in addition of the ISD-P identified by the AID `#ISD_P_AID1`).*				

4.2.8.2.3 TC.ES5.ECA.3: eUICCCapabilityAudit_HTTPS

Test Purpose

To ensure it is possible to audit the eUICC using HTTPS. GET STATUS and GET DATA commands are sent to retrieve the ISD-P list, the ECASD certificate, the eUICC recognition data and the card resources information.

Referenced Requirements

- PF_REQ7
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.8.2.3.1 Test Sequence N°1 – Nominal Case: Retrieve all Information

Initial Conditions

- `#ISD_P_AID1` in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`    [GET_ISDP_LIST])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 55. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 56. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP_LIST3] 57. (see Note 1)	PF_REQ7, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_ISDP_ENABLED])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 58. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 59. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP_3F]	PF_REQ7, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
7	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_ISDP_DISABLED])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
8	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 50. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 51. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP1_1F] 52. (see Note 2)	PF_REQ7, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
9	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_DATA_FF21])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
10	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 53. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 54. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_FF21]	PF_REQ7, EUICC_REQ14, EUICC_REQ16, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
11	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_DATA_BF30_REC])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ

12	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			55. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			56. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_BF30_REC]`
				PF_REQ7,
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ43,
				EUICC_REQ46,
				EUICC_REQ47,
				EUICC_REQ48,
				EUICC_REQ52

13	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[GET_DATA_BF30_CERT])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

14	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			57. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			58. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_BF30_CERT]`
				PF_REQ7,
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ43,
				EUICC_REQ46,
				EUICC_REQ47,
				EUICC_REQ48,
				EUICC_REQ52

15	Close HTTPS session as described in section 4.2.1.7			

Note 1: If more than one Profile is pre-installed on the eUICC, this response shall be adapted in consequence (in addition of the Enabled ISD-P identified by the AID `#DEFAULT_ISD_P_AID` and the ISD-P identified by the AID `#ISD_P_AID1`, other Profiles may be present).

Note 2: If more than one Profile is pre-installed on the eUICC (i.e. several Disabled Profiles exist), this response shall be adapted in consequence (in addition of the ISD-P identified by the AID `#ISD_P_AID1`).

4.2.9 ES5 (SM-SR – eUICC): MasterDelete

4.2.9.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ7, PF_REQ8
- SEC_REQ12, SEC_REQ14
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.9.2 Test Cases

General Initial Conditions

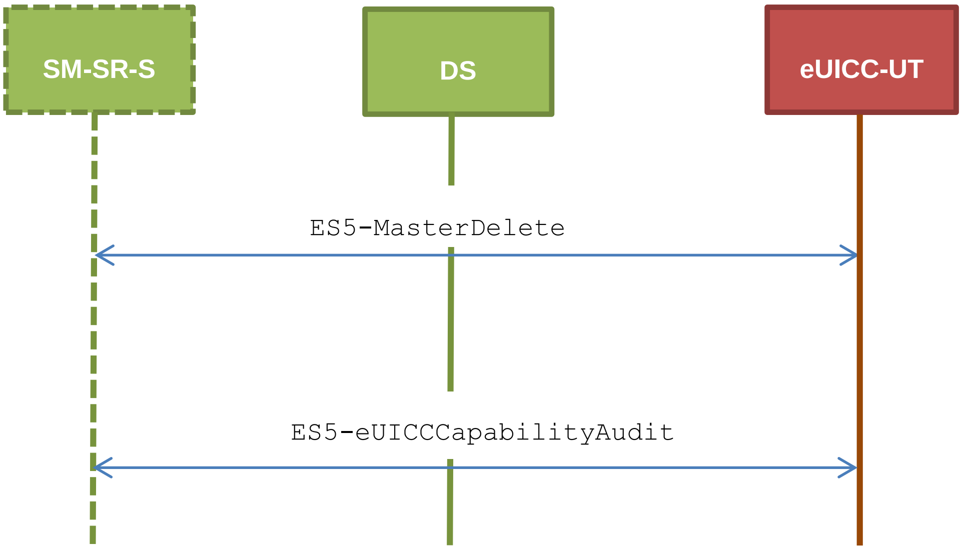
- #ISD_P_AID1 present on the eUICC and personalized with SCP03 keys
  - The process *ES8-EstablishISDPKeySet* has been used
  - {SCP_KENC}, {SCP_KMAC}, {SCP_KDEK} have been set
- #ISD_P_AID1 contains a keyset '70' with an AES key (16 bytes long)
  - A PUT KEY command as defined in the GlobalPlatform Card Specification [3] should be used to initialize the {TOKEN_KEY}
  - The value of the {TOKEN_KEY} can be freely chosen by the test tool
- #ISD_P_AID1 contains the SDIN value #ISD_P_SDIN*
- #ISD_P_AID1 contains the SIN value #ISD_P_SIN*
- #ISD_P_AID1 contains the Application Provider Identifier value #ISD_P_PROV_ID*

** To set the SDIN, SIN and the Application Provider Identifier, the sequence below shall be executed just after the establishment of the ISD-P keysets:*

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_P_TAR1, SCP03_SCRIPT(#SCP03_KVN, [STORE_SDIN]; [STORE_SIN]; [STORE_PROV_ID])) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		

Step	Direction	Sequence / Description	Expected result	REQ
2	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
3	DS → eUICC-UT	FETCH		
4	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- For each R-APDU received: a. SW='9000' or '6108' 69.	70.
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Test Environment



4.2.9.2.1 TC.ES5.MD.1: MasterDelete_SMS

Test Purpose

To ensure the master deletion process is well implemented on the eUICC using SMS. After ISD-P deletion, a GET STATUS command is sent to make sure that the security domain is no longer present on the eUICC. Some error cases due to incompatible initial conditions or incorrect values in commands are also defined.

Referenced Requirements

- PF_REQ7, PF_REQ8
- SEC_REQ12, SEC_REQ14
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54

Initial Conditions

- None

4.2.9.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- #ISD_P_AID1 in Disabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute Set
- No POL1 defined on #ISD_P_AID1

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [MASTER_DEL_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_009000]	PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6A88]	PF_REQ7, PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, SEC_REQ12
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.9.2.1.2 Test Sequence N°2 – Nominal Case: ISD-P with POL1 “Deletion not allowed”

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set
- `#ISD_P_AID1` contains the POL1 “Deletion of the Profile not allowed”

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [MASTER_DEL_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_009000]	PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, SEC_REQ14
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6A88]	PF_REQ7, PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, SEC_REQ12
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.9.2.1.3 Test Sequence N°3 – Error Case: ISD-P Not Disabled

Initial Conditions

- #ISD_P_AID1 in Enabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [MASTER_DEL_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6985]	PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_3F]	PF_REQ7, PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.9.2.1.4 Test Sequence N°4 – Error Case: ISD-P with the Fall-back Attribute Set

Initial Conditions

- #ISD_P_AID1 in Disabled state
- #ISD_P_AID1 is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [MASTER_DEL_ISDP1])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6985]	PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_1F]	PF_REQ7, PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.9.2.1.5 Test Sequence N°5 – Error Case: Wrong Token Value

Initial Conditions

- #ISD_P_AID1 in Disabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[BAD_MASTER_DEL_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1-Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6985]` 1. (see Note 1)	PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_1F]`	PF_REQ7, PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW may be also '6A80' or '6982'				

4.2.9.2.2 TC.ES5.MD.2: MasterDelete_CAT_TP

Test Purpose

To ensure the master deletion process is well implemented on the eUICC using CAT_TP. After ISD-P deletion, a GET STATUS command is sent to make sure that the security domain is no longer present on the eUICC.

Referenced Requirements

- PF_REQ7, PF_REQ8

- SEC_REQ12
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.9.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [MASTER_DEL_ISDP1])		EUICC_REQ54
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 72.#SCP80_AUTH_KEY 4- The response data is equal to [R_AB_009000]	PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
5	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ54
6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 73.#SCP80_AUTH_KEY 4- The response data is equal to [R_AB_6A88]	PF_REQ7, PF_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, SEC_REQ12
7	Close CAT_TP session as described in section 4.2.1.4			

4.2.9.2.3 TC.ES5.MD.3: MasterDelete_HTTPS

Test Purpose

To ensure the master deletion process is well implemented on the eUICC using HTTPS. After ISD-P deletion, a GET STATUS command is sent to make sure that the security domain is no longer present on the eUICC.

Referenced Requirements

- PF_REQ7, PF_REQ8
- SEC_REQ12
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.9.2.3.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[MASTER_DEL_ISDP1])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ

4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			74. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			75. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_009000]`
				PF_REQ8,
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ43,
				EUICC_REQ46,
				EUICC_REQ47,
				EUICC_REQ48,
				EUICC_REQ52

5	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[GET_ISDP1])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			76. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			77. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_6A88]`
				PF_REQ7,
				PF_REQ8,
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ43,
				EUICC_REQ46,
				EUICC_REQ47,
				EUICC_REQ48,
				EUICC_REQ52,
				SEC_REQ12

7	Close HTTPS session as described in section 4.2.1.7			

4.2.10 ES5 (SM-SR – eUICC): EstablishISDRKeySet

4.2.10.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

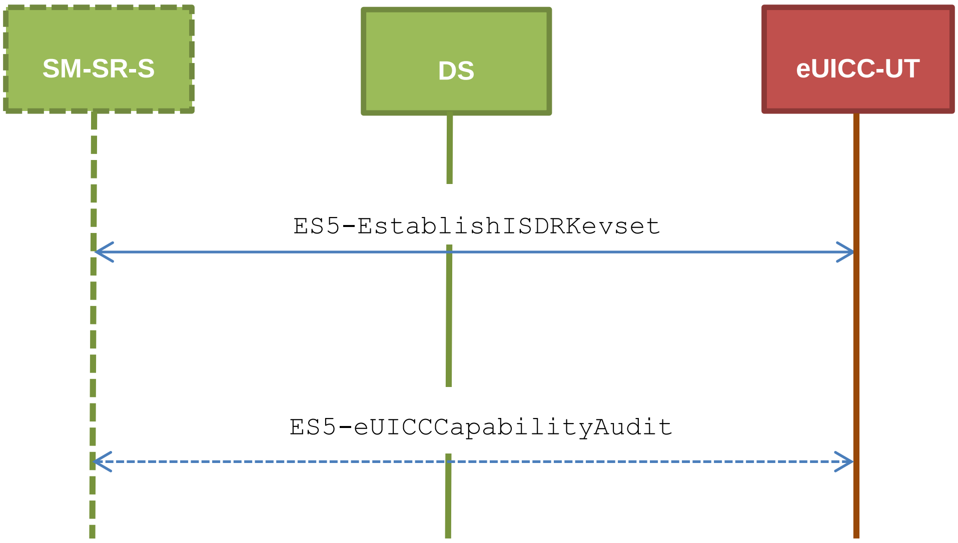
- PF_REQ7
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.10.2 Test Cases

General Initial Conditions

- None

Test Environment



4.2.10.2.1 TC.ES5.EISDRK.1: EstablishISDRKeyset_SMS

Test Purpose

To ensure the ISD-R keyset establishment process is well implemented on the eUICC using SMS. After SCP80 keys initialization on ISD-R, a new secure channel session is opened to make sure that the new keys have been set. During the key establishment, different parameters are used (DR, HostID) to make sure that all configurations are supported on the eUICC. An error case is defined to test that an incorrect SM-SR certificate is rejected.

Referenced Requirements

- PF_REQ7
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24, EUICC_REQ54

Initial Conditions

- None

4.2.10.2.1.1 Test Sequence N°1 – Nominal Case: No DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_SR_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RC]` 4- Retrieve the `{RC}`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`STORE_ISDR_KEYS(`
		`#SC3_NO_DR;`
		`{RC}),`
		`#LAST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RECEIPT]` 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tag 'A6')	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

12	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_ENABLED])`
		Use `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
				EUICC_REQ22,
				EUICC_REQ54

13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `{SCP_KENC}` 2- Verify the cryptographic checksum using `{SCP_KMAC}` 3- The response data is equal to `[R_AB_E3_ISDP_3F]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.10.2.1.2 Test Sequence N°2 – Nominal case: DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_SR_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RC]` 4- Retrieve the `{RC}`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`STORE_ISDR_KEYS(`
		`#SC3_DR;`
		`{RC}),`
		`#LAST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_02RECEIPT_DR] 4- Calculate ShS from #SM_ESK_ECKA and #PK_ECASD_ECKA 5- Derive keyset from ShS and {DR} and retrieve the {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK} 6- Verify the {RECEIPT} (i.e. it shall be generated by calculating a MAC across the tags ‘A6’ and ‘85’)	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
11	DS → eUICC-UT	TERMINAL RESPONSE	SW=’9000’	
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED]) Use {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		EUICC_REQ22, EUICC_REQ54
13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the {SCP_KENC} 2- Verify the cryptographic checksum using {SCP_KMAC} 3- The response data is equal to [R_AB_E3_ISDP_3F]	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
16	DS → eUICC-UT	TERMINAL RESPONSE	SW=’9000’	

4.2.10.2.1.3 Test Sequence N°3 – Nominal Case: DR, Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_SR_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RC]` 4- Retrieve the `{RC}`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`STORE_ISDR_KEYS(`
		`#SC3_DR_HOST;`
		`{RC}),`
		`#LAST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RECEIPT_DR]` 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS (using `{DR}`, `#HOST_ID`, `#ISD_R_SIN` and `#ISD_R_SDIN)` and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tags 'A6' and '85')	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

12	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_ENABLED])`
		Use `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
				EUICC_REQ22,
				EUICC_REQ54

13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `{SCP_KENC}` 2- Verify the cryptographic checksum using `{SCP_KMAC}` 3- The response data is equal to `[R_AB_E3_ISDP_3F]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.10.2.1.4 Test Sequence N°4 – Error Case: Invalid SM-SR Certificate

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_INVALID_SR_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_026982]` (see Note)	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: The SW may be also '6A80'				

4.2.10.2.2 TC.ES5.EISDRK.2: EstablishISDRKeyset_CAT_TP

Test Purpose

To ensure the ISD-R keyset establishment process is well implemented on the eUICC using CAT_TP. After ISD-R keys initialization, a new secure channel is opened to make sure that the new keys have been set.

Referenced Requirements

- PF_REQ7
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ24, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.10.2.2.1 Test Sequence N°1 – Nominal Case: No DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [STORE_SR_CERTIF], #FIRST_SCRIPT)		EUICC_REQ54
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 78. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_02RC] 5- Retrieve the {RC}	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ24
5	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, STORE_ISDR_KEYS(#SC3_NO_DR; {RC}), #LAST_SCRIPT)		EUICC_REQ54
6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 79. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_02RECEIPT] 5- Calculate ShS from #SM_ESK_ECKA and #PK_ECASD_ECKA 6- Derive keyset from ShS and retrieve the {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK} 7- Verify the {RECEIPT} (i.e. it shall be generated by calculating a MAC across the tag 'A6')	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ24

Step	Direction	Sequence / Description	Expected result	REQ

7	DS → eUICC-UT	ACK_DATA containing the result of `SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_ENABLED])`
		Use `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
				EUICC_REQ54

8	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `{SCP_KENC}` 3- Verify the cryptographic checksum using `{SCP_KMAC}` 4- The response data is equal to `[R_AB_E3_ISDP_3F]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ24
9	Close CAT_TP session as described in section 4.2.1.4			

4.2.10.2.3 TC.ES5.EISDRK.3: EstablishISDRKeyset_HTTPS

Test Purpose

To ensure the ISD-R keyset establishment process is well implemented on the eUICC using HTTPS. After ISD-R keys initialization, a new secure channel is opened to make sure that the new keys have been set.

Referenced Requirements

- PF_REQ7
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ24, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.10.2.3.1 Test Sequence N°1 – Nominal Case: No DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`    [STORE_SR_CERTIF])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			30. `#HOST`
			31. `#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			32. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_RC]`
			5- Retrieve the `{RC}`
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ24,
				EUICC_REQ43,
				EUICC_REQ46,
				EUICC_REQ47,
				EUICC_REQ48,
				EUICC_REQ52

5	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`    STORE_ISDR_KEYS(`
		`        #SC3_NO_DR;`
		`        {RC}))`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 33. #HOST 34. #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 35. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_RECEIPT] 5- Calculate ShS from #SM_ESK_ECKA and #PK_ECASD_ECKA 6- Derive keyset from ShS and retrieve the {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK} 7- Verify the {RECEIPT} (i.e. it shall be generated by calculating a MAC across the tag 'A6')	EUICC_REQ14, EUICC_REQ16, EUICC_REQ24, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
7	Close HTTPS session as described in section 4.2.1.7			
8	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED]) Use {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		EUICC_REQ54
9	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
10	DS → eUICC-UT	FETCH		
11	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the {SCP_KENC} 2- Verify the cryptographic checksum using {SCP_KMAC} 3- The response data is equal to [R_AB_E3_ISDP_3F]	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24
12	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.11 ES5 (SM-SR – eUICC): FinaliseISDRhandover

4.2.11.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

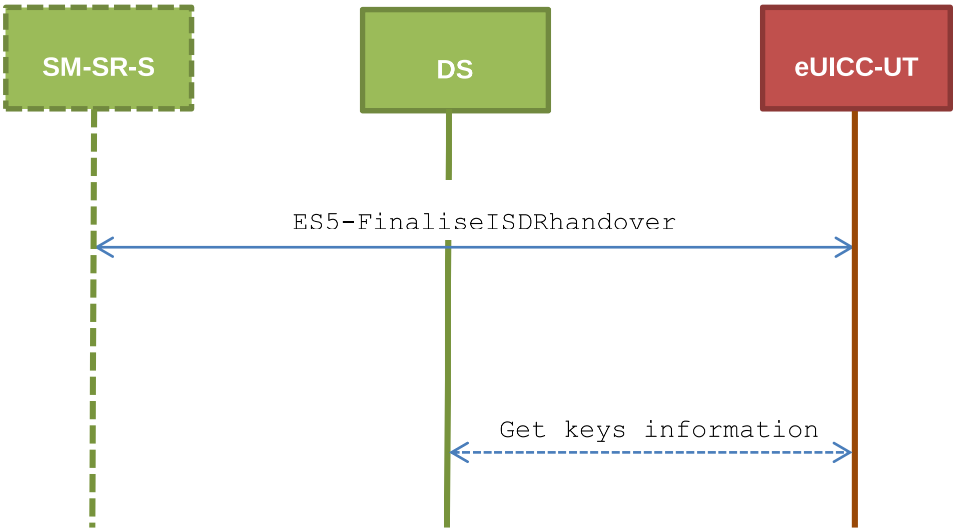
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.11.2 Test Cases

General Initial Conditions

- An additional keyset with the key version number `#SCP80_NEW_KVN` is initialized on the ISD-R

Test Environment



4.2.11.2.1 TC.ES5.FIH.1: FinaliseISDRhandover_SMS

Test Purpose

To ensure it is possible to delete ISD-R keys on the eUICC using SMS. After keysets deletion, a GET DATA (TAG 'E0' – key information template) is sent to retrieve all the keysets present on the ISD-R to make sure that the range of keyset has been deleted correctly. Some error cases due to inconsistent values in commands are also defined.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25, EUICC_REQ54

Initial Conditions

- None

4.2.11.2.1.1 Test Sequence N°1 – Nominal Case: Delete All Keys except SCP80 Keys

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[DELETE1_KEYSETS])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_009000]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_E0])`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E0_SCP80]` (i.e. no `#SCP80_NEW_KVN` returned)	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.11.2.1.2 Test Sequence N°2 – Nominal Case: Delete All Keys except SCP80 and SCP81 Keys

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [DELETE2_KEYSETS])		EUICC_REQ22, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_009000]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_DATA_E0])		EUICC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E0_SCP80_SCP81] (i.e. no #SCP80_NEW_KVN returned)	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.11.2.1.3 Test Sequence N°3 – Error Case: Delete All SCP80 Keys

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[DELETE_SCP80_KEYSETS])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6985]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.11.2.2 TC.ES5.FIH.2: FinaliseISDRhandover_CAT_TP

Test Purpose

To ensure it is possible to delete ISD-R keys on the eUICC using CAT_TP. After keysets deletion, a GET DATA (TAG 'E0' – key information template) is sent to retrieve all the keysets present on the ISD-R to make sure that the range of keyset has been deleted correctly.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22, EUICC_REQ25, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.11.2.2.1 Test Sequence N°1 – Nominal Case: Delete All Keys except SCP80 Keys

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

Step	Direction	Sequence / Description	Expected result	REQ

3	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[DELETE1_KEYSETS])`
				EUICC_REQ54

4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using 36. `#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_009000]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ25

5	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DATA_E0])`
				EUICC_REQ54

6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using 37. `#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_E0_SCP80]` (i.e. no `#SCP80_NEW_KVN` 38. returned)	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ25
7	Close CAT_TP session as described in section 4.2.1.4			

4.2.11.2.2.2 Test Sequence N°2 – Nominal Case: Delete All Keys except SCP80 and SCP81 Keys

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

Step	Direction	Sequence / Description	Expected result	REQ
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [DELETE2_KEYSETS])		EUICC_REQ54
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 39. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_009000]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ25
5	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_DATA_E0])		EUICC_REQ54
6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 90. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E0_SCP80_SCP81] (i.e. no #SCP80_NEW_KVN returned)	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ25
7	Close CAT_TP session as described in section 4.2.1.4			

4.2.11.2.3 TC.ES5.FIH.3: FinaliseISDRhandover_HTTPS

Test Purpose

To ensure it is possible to delete ISD-R keys on the eUICC using HTTPS. After keysets deletion, a GET DATA (TAG 'E0' – key information template) is sent to retrieve all the keysets present on the ISD-R to make sure that the range of keyset has been deleted correctly.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ25, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.11.2.3.1 Test Sequence N°1 – Nominal Case: Delete All Keys except SCP80 and SCP81 Keys

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[DELETE2_KEYSETS])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			01. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			02. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_009000]`
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ25,
				EUICC_REQ43,
				EUICC_REQ46,
				EUICC_REQ47,
				EUICC_REQ48,
				EUICC_REQ52

5	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[GET_DATA_E0])`
				EUICC_REQ49,
				EUICC_REQ50,
				EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 3. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 4. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E0_SCP80_SCP81 (i.e. no #SCP80_NEW_KVN returned)	EUICC_REQ14, EUICC_REQ16, EUICC_REQ25, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
7	Close HTTPS session as described in section 4.2.1.7			

4.2.12 ES5 (SM-SR – eUICC): UpdateSMSRAddressingParameters

4.2.12.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

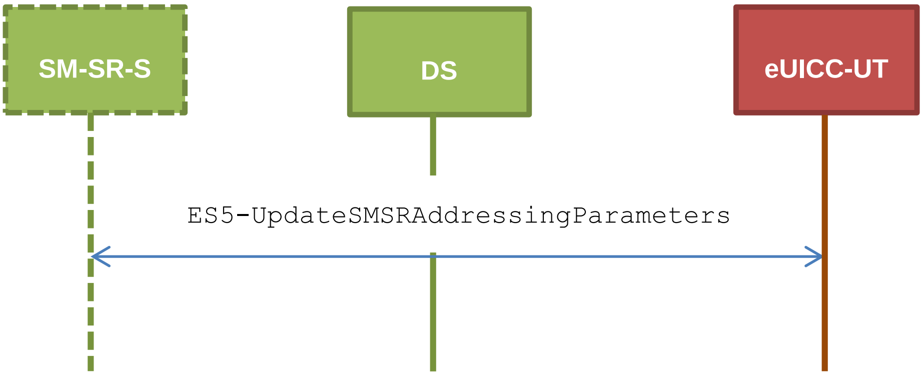
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.12.2 Test Cases

General Initial Conditions

- None

Test Environment



4.2.12.2.1 TC.ES5.USAP.1: UpdateSMSRAddrParam_SMS

Test Purpose

To ensure it is possible to update SM-SR addressing parameters on the eUICC using SMS. Some error cases due to inconsistent values in commands are also defined.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26, EUICC_REQ54

Initial Conditions

- None

4.2.12.2.1.1 Test Sequence N°1 – Nominal Case: Update SMS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_SMS_PARAM])`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_9000]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.12.2.1.2 Test Sequence N°2 – Error Case: Update CAT_TP Parameters when CAT_TP Not Supported

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_CATTP_PARAM])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6A80]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.12.2.1.3 Test Sequence N°3 – Error Case: Update HTTPS Parameters when HTTPS Not Supported

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_HTTPS_PARAM])`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6A80]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.12.2.2 TC.ES5.USAP.2: UpdateSMSRAddrParam_CAT_TP

4.2.12.2.2.1 Test Sequence N°1 – Nominal Case: Update CAT_TP Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

3	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_CATTP_PARAM])`
				EUICC_REQ54

4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using 05. `#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_9000]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ26
5	Close CAT_TP session as described in section 4.2.1.4			

4.2.12.2.3 TC.ES5.USAP.3: UpdateSMSRAddrParam_HTTPS

Test Purpose

To ensure it is possible to update SM-SR addressing parameters on the eUICC using HTTPS.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ26, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: #PSK_ID
    - PSK value: #SCP81_PSK

4.2.12.2.3.1 Test Sequence N°1 – Nominal Case: Update HTTPS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([STORE_HTTPS_PARAM])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 6. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 7. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_9000]	EUICC_REQ14, EUICC_REQ16, EUICC_REQ26, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
5	Close HTTPS session as described in section 4.2.1.7			

4.2.13 ES5 (SM-SR – eUICC): Notification on Profile Enabling

4.2.13.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

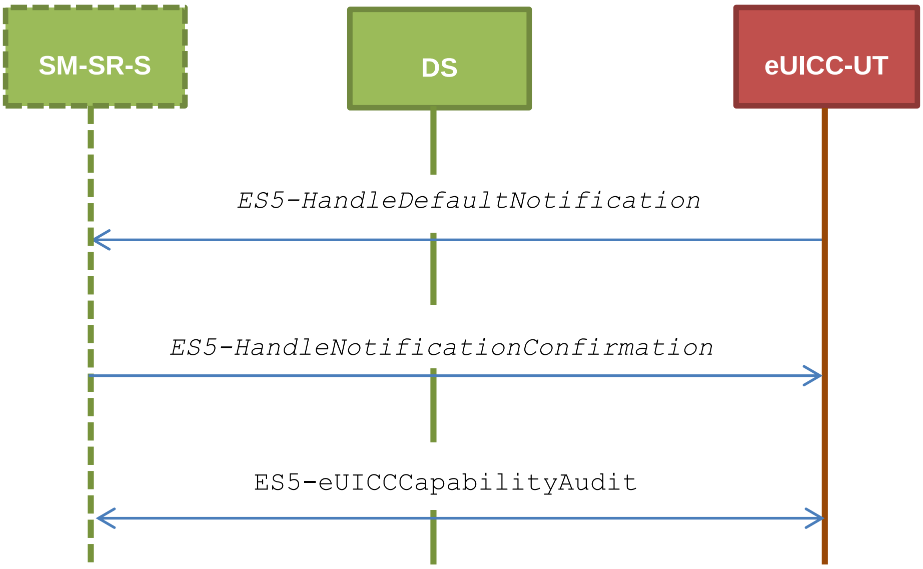
- PF_REQ4, PF_REQ7
- PM_REQ3, PM_REQ4
- PROC_REQ6, PROC_REQ8, PROC_REQ20, PROC_REQ21
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ54

4.2.13.2 Test Cases

General Initial Conditions

- The #ISD_P_AID1 has just been Enabled
  - REFRESH proactive command has been sent by the eUICC
  - To Enable this Profile, the Profile enabling process shall be used (i.e. the test sequence defined in section 4.2.4.2.1.1 may be executed)

Test Environment


SM-SR-S
DS
eUICC-UT
ES5-HandleDefaultNotification
ES5-HandleNotificationConfirmation
ES5-eUICCCapabilityAudit

4.2.13.2.1 TC.ES5.NOTIFPE.1: Notification_SMS

Test Purpose

To ensure SMS notification procedure is well implemented when a Profile is Enabled.

Note: As the update of the lifecycle states may become effective after the REFRESH command, the check of the lifecycle states of the Profiles is performed in this test case.

Referenced Requirements

- PF_REQ4, PF_REQ7
- PM_REQ3, PM_REQ4
- PROC_REQ6, PROC_REQ8, PROC_REQ20
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ54

Initial Conditions

- The SMS mode is the default way (priority order 1) to send the notification
- TP-Destination-Address has been set on #ISD_R_AID with #DEST_ADDR
- SMS-C parameters have been set on #ISD_P_AID1 with #TON_NPI and #DIALING_NUMBER

4.2.13.2.1.1 Test Sequence N°1 – Nominal Case: No Follow-up Activities

Initial Conditions

- No POL1 defined in the previous Enabled ISD-P (i.e. #DEFAULT_ISD_P_AID)

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- The TP-Destination-Address is equal to #DEST_ADDR 2- The SMS-C address is equal to #TON_NPI + #DIALING_NUMBER 3- The SPI is equal to #SPI_NOTIF 4- Verify the cryptographic checksum using #SCP80_AUTH_KEY 5- The secured data shall only contain the TLV #NOTIF_PROFILE_CHANGE (see Note 1) 6- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ20,

Step	Direction	Sequence / Description	Expected result	REQ
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [NOTIF_CONFIRMATION])		PROC_REQ20, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 98. #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_NOTIF]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ20
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])		EUICC_REQ22, EUICC_REQ54
13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP_LIST1]	PM_REQ3, PM_REQ4, PF_REQ4, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Step	Direction	Sequence / Description	Expected result	REQ
Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification. *Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.* *Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).*				

4.2.13.2.1.2 Test Sequence N°2 – Nominal Case: Follow-up Activity

Initial Conditions

- The previous Enabled ISD-P's (i.e. `#DEFAULT_ISD_P_AID`) POL1 contains the rule "Profile deletion is mandatory when it is disabled"

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- The TP-Destination-Address is equal to `#DEST_ADDR` 2- The SMS-C address is equal to `#TON_NPI + #DIALING_NUMBER` 3- The SPI is equal to `#SPI_NOTIF` 4- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 5- The secured data shall only contain the TLV `#NOTIF_PROFILE_CHANGE` (see Note 1) 6- Extract the `{NOTIF_NUMBER}`	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ20
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[NOTIF_CONFIRMATION])`
				PROC_REQ20, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 99.`#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_NOTIF1]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ20
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

12	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DEFAULT_ISDP])`
				EUICC_REQ54

13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using 100.`#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6A88]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification.

Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.

Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).

4.2.13.2.1.3 Test Sequence N°3 – Error Case: SM-SR Unreachable

Initial Conditions

- No POL1 defined in the previous Enabled ISD-P (i.e. `#DEFAULT_ISD_P_AID`)

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- The TP-Destination-Address is equal to `#DEST_ADDR` 2- The SMS-C address is equal to `#TON_NPI + #DIALING_NUMBER` 3- The SPI is equal to `#SPI_NOTIF` 4- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 5- The secured data shall only contain the TLV `#NOTIF_PROFILE_CHANGE` `01.` (see Note 1) 6- Extract the `{NOTIF_NUMBER}`	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ20
6	DS → eUICC-UT	TERMINAL RESPONSE		
Loop while maximum retries number is not reached				
7	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* TIMER MANAGEMENT		
8	DS → eUICC-UT	FETCH		
9	eUICC-UT → DS	*PROACTIVE COMMAND:* TIMER MANAGEMENT		EUICC_REQ27, PROC_REQ6, PROC_REQ8, PROC_REQ20
10	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
11	DS → eUICC-UT	ENVELOPE TIMER EXPIRATION		
12	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
13	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
14	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- The TP-Destination-Address is equal to `#DEST_ADDR` 2- The SMS-C address is equal to `#TON_NPI + #DIALING_NUMBER` 3- The SPI is equal to `#SPI_NOTIF` 4- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 5- The secured data shall only contain the TLV `#NOTIF_PROFILE_CHANGE` 102. (see Note 1) 6- Extract the `{NOTIF_NUMBER}`: it shall be the same as the previous one	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ6, PROC_REQ8, PROC_REQ20
15	DS → eUICC-UT	TERMINAL RESPONSE		
End loop				
16	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* TIMER MANAGEMENT		
17	DS → eUICC-UT	FETCH		
18	eUICC-UT → DS	*PROACTIVE COMMAND:* TIMER MANAGEMENT		EUICC_REQ27, PROC_REQ6, PROC_REQ8, PROC_REQ20
19	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
20	DS → eUICC-UT	ENVELOPE TIMER EXPIRATION		
21	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH		
22	DS → eUICC-UT	FETCH		
23	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PM_REQ3, PROC_REQ6, PROC_REQ8
24	DS → eUICC-UT	RESET	ATR returned by eUICC	
25	Initialization sequence as described in section 4.2.1.1			
26	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
27	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
28	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- The TP-Destination-Address is equal to `#DEST_ADDR` 2- The SPI is equal to `#SPI_NOTIF` 3- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 4- The secured data shall only contain the TLV `#NOTIF_ROLL_BACK` 03. (see Note 1) 5- Extract the `{NOTIF_NUMBER}`: it shall not be the same as the previous one	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ6, PROC_REQ8
29	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

30	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[NOTIF_CONFIRMATION])`
				PROC_REQ6, PROC_REQ8, EUICC_REQ54

31	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
32	DS → eUICC-UT	FETCH		
33	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `04.#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_NOTIF]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ6, PROC_REQ8
34	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

35	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_ENABLED])`
				EUICC_REQ54

36	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
37	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
38	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 05.#SCP80_AUTH_KEY 3- The response data is equal to 06. [R_AB_E3_ISDP_3F]	PM_REQ3, PM_REQ4, PF_REQ7, PROC_REQ6, PROC_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29
39	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification.

Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.

Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).

4.2.13.2.2 TC.ES5.NOTIFPE.2: Notification_CAT_TP

Test Purpose

To ensure CAT_TP notification procedure is well implemented when a Profile is Enabled.

Note: As the update of the lifecycle states may become effective after the REFRESH command, the check of the lifecycle states of the Profiles is performed in this test case.

Referenced Requirements

- PF_REQ4, PF_REQ7
- PM_REQ3, PM_REQ4
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ54

Initial Conditions

- The CAT_TP mode is the default way (priority order 1) to send the notification

4.2.13.2.2.1 Test Sequence N°1 – Nominal Case: No Follow-up Activities

Initial Conditions

- No POL1 defined in the previous Enabled ISD-P (i.e. #DEFAULT_ISD_P_AID)
- CAT_TP Connectivity Parameters have been set on #ISD_R_AID with #UDP_PORT, #CAT_TP_PORT and #IP_VALUE
- CAT_TP Connectivity Parameters have been set on #ISD_P_AID1 with #BEARER_DESCRIPTION, #NAN_VALUE, #LOGIN and #PWD

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL	107.	
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to #BEARER_DESCRIPTION 2- The NAN is equal to #NAN_VALUE 3- The port is equal to #UDP_PORT 4- The IP is equal to #IP_VALUE 5- The login/password are equal to #LOGIN/#PWD	EUICC_REQ18, EUICC_REQ27
6	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The CAT_TP PDU used here after shall be compliant with the Annex G.*				
7	eUICC-UT → DS	SYN	The identification data may contain the #EID	EUICC_REQ18
8	DS → eUICC-UT	SYN_ACK		
9	eUICC-UT → DS	ACK_NO_DATA	The CAT_TP session is open.	EUICC_REQ18
10	eUICC-UT → DS	ACK_DATA containing the notification	1- The ACK_DATA contains a command packet 2- The SPI is equal to #SPI_NOTIF 3- Verify the cryptographic checksum using #SCP80_AUTH_KEY 4- The secured data shall only contain the TLV #NOTIF_PROFILE_CHANGE 108. (see Note 1) 5- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ18, EUICC_REQ27, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ

11	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[NOTIF_CONFIRMATION])`
				EUICC_REQ54

12	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `09.#SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_NOTIF]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ29
13	Close CAT_TP session as described in section 4.2.1.4			

14	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1];`
		`[GET_DEFAULT_ISDP])`
			10.	EUICC_REQ22,
				EUICC_REQ54

15	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
16	DS → eUICC-UT	FETCH	11.	
17	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP_LIST1]`	PM_REQ3, PM_REQ4, PF_REQ4, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
18	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification.

Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.

Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).

4.2.13.2.3 TC.ES5.NOTIFPE.3: Notification_HTTPS

Test Purpose

To ensure HTTPS notification procedure is well implemented when a Profile is Enabled.

Note: As the update of the lifecycle states may become effective after the REFRESH command, the check of the lifecycle states of the Profiles is performed in this test case.

Referenced Requirements

- PF_REQ4, PF_REQ7
- PM_REQ3, PM_REQ4
- PROC_REQ21
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS mode is the default way (priority order 1) to send the notification
- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: #PSK_ID
    - PSK value: #SCP81_PSK

4.2.13.2.3.1 Test Sequence N°1 – Nominal Case: No Follow-up Activities

Initial Conditions

- No POL1 defined in the previous Enabled ISD-P (i.e. #DEFAULT_ISD_P_AID)
- HTTPS Connectivity Parameters have been set on #ISD_R_AID with #TCP_PORT, #IP_VALUE, #ADMIN_HOST, #AGENT_ID, #PSK_ID, #SCP81_KVN, #SCP81_KEY_ID and #ADMIN_URI
- HTTPS Connectivity Parameters have been set on #ISD_P_AID1 with #BEARER_DESCRIPTION, #NAN_VALUE, #LOGIN and #PWD

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL	112.	
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to `#BEARER_DESCRIPTION` 2- The NAN is equal to `#NAN_VALUE` 3- The port is equal to `#TCP_PORT` 4- The IP is equal to `#IP_VALUE` 5- The login/password are equal to `#LOGIN/#PWD`	EUICC_REQ13, EUICC_REQ14, PROC_REQ21
6	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The TLS records used here after shall be compliant with the 0.*				
7	eUICC-UT → DS	TLS_CLIENT_HELLO	The CLIENT_HELLO shall contain at least one of the cipher-suites accepted by the HTTPS server.	EUICC_REQ14, EUICC_REQ43, PROC_REQ21
8	DS → eUICC-UT	TLS_SERVER_HELLO and TLS_SERVER_HELLO_DONE		PROC_REQ21
9	eUICC-UT → DS	TLS_CLIENT_KEY_EXCHANGE and TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED	The CLIENT_KEY_EXCHANGE shall contain the `#PSK_ID`	EUICC_REQ14, EUICC_REQ43, EUICC_REQ45, PROC_REQ21
10	DS → eUICC-UT	TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED		PROC_REQ21
11	eUICC-UT → DS	TLS_APPLICATION with the first POST message	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake 2- The HTTP content is empty 13. The POST URI is equal to `#POST_URI_NOTIF` (see Note 1) 3- The headers are equal to 114.`#HOST` 115.`#X_ADMIN_PROTOCOL` 116. `#X_ADMIN_FROM_ISD_R`	EUICC_REQ14, EUICC_REQ27, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, PROC_REQ21

Step	Direction	Sequence / Description	Expected result	REQ
12	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([NOTIF_CONFIRMATION])	17.	EUICC_REQ29, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, PROC_REQ21
13	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 18. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 19. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_NOTIF]	EUICC_REQ14, EUICC_REQ16, EUICC_REQ29, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, PROC_REQ21
14	Close HTTPS session as described in section 4.2.1.7			
15	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])	20.	EUICC_REQ22, EUICC_REQ54
16	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE	21.	
17	DS → eUICC-UT	FETCH	22.	
18	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP_LIST1]	PM_REQ3, PM_REQ4, PF_REQ4, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
19	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Step	Direction	Sequence / Description	Expected result	REQ
Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification. *Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.* *Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).*				

4.2.14 ES5 (SM-SR – eUICC): Notification on Profile Disabling

4.2.14.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

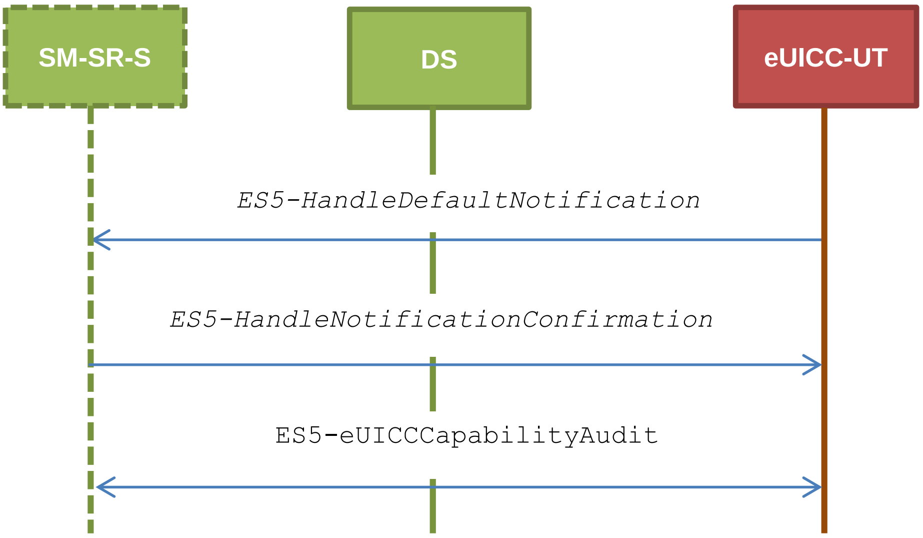
- PF_REQ5, PF_REQ7
- PM_REQ3, PM_REQ4
- PROC_REQ20, PROC_REQ21, PROC_REQ22
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ54

4.2.14.2 Test Cases

General Initial Conditions

- The `#ISD_P_AID1` has just been Disabled
  - REFRESH proactive command has been sent by the eUICC
  - To Disable this Profile, the Profile disabling process shall be used (i.e. the test sequence defined in section 4.2.5.2.1.1 may be executed)
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute Set

Test Environment


SM-SR-S
DS
eUICC-UT
ES5-HandleDefaultNotification
ES5-HandleNotificationConfirmation
ES5-eUICCCapabilityAudit

4.2.14.2.1 TC.ES5.NOTIFPD.1: Notification_SMS

Test Purpose

To ensure SMS notification procedure is well implemented when a Profile is Disabled.

Note: As the update of the lifecycle states may become effective after the REFRESH command, the check of the lifecycle states of the Profiles is performed in this test case (the ISD-P with the Fall-back Attribute Set shall be Enabled).

Referenced Requirements

- PF_REQ5, PF_REQ7
- PM_REQ3, PM_REQ4
- PROC_REQ20, PROC_REQ22
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ54

Initial Conditions

- The SMS mode is the default way (priority order 1) to send the notification
- TP-Destination-Address has been set on #ISD_R_AID with #DEST_ADDR
- SMS-C parameters have been set on #DEFAULT_ISD_P_AID with #TON_NPI and #DIALING_NUMBER

4.2.14.2.1.1 Test Sequence N°1 – Nominal Case: No Follow-up Activities

Initial Conditions

- No POL1 defined in the previous Enabled ISD-P (i.e. #ISD_P_AID1)

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	

Step	Direction	Sequence / Description	Expected result	REQ
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- The TP-Destination-Address is equal to #DEST_ADDR 2- The SMS-C address is equal to #TON_NPI + #DIALING_NUMBER 3- The SPI is equal to #SPI_NOTIF 4- Verify the cryptographic checksum using #SCP80_AUTH_KEY 5- The secured data shall only contain the TLV #NOTIF_PROFILE_CHANGE2 (see Note 1) 6- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ20, PROC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [NOTIF_CONFIRMATION])		PROC_REQ20, PROC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 23.#SCP80_AUTH_KEY 3- The response data is equal to [R_AB_NOTIF]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ20, PROC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])		EUICC_REQ22, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP_LIST2]`	PM_REQ3, PM_REQ4, PF_REQ5, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PROC_REQ22
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification.

Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.

Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).

4.2.14.2.1.2 Test Sequence N°2 – Nominal Case: Follow-up Activity

Initial Conditions

- The previous Enabled ISD-P's (i.e. `#ISD_P_AID1`) POL1 contains the rule "Profile deletion is mandatory when it is disabled"

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- The TP-Destination-Address is equal to #DEST_ADDR 2- The SMS-C address is equal to #TON_NPI + #DIALING_NUMBER 3- The SPI is equal to #SPI_NOTIF 4- Verify the cryptographic checksum using #SCP80_AUTH_KEY 5- The secured data shall only contain the TLV #NOTIF_PROFILE_CHANGE2 (see Note 1) 6- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ20, PROC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [NOTIF_CONFIRMATION])		PROC_REQ20, PROC_REQ22, EUICC_REQ54
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using 24.#SCP80_AUTH_KEY 3- The response data is equal to [R_AB_NOTIF2]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ20, PROC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ54
13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `125.#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6A88]`	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ22
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification.

Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.

Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).

4.2.14.2.2 TC.ES5.NOTIFPD.2: Notification_CAT_TP

Test Purpose

To ensure CAT_TP notification procedure is well implemented when a Profile is Disabled.

Note: As the update of the lifecycle states may become effective after the REFRESH command, the check of the lifecycle states of the Profiles is performed in this test case (the ISD-P with the Fall-back Attribute Set shall be Enabled).

Referenced Requirements

- PF_REQ5, PF_REQ7
- PM_REQ3, PM_REQ4
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ54
- PROC_REQ22

Initial Conditions

- The CAT_TP mode is the default way (priority order 1) to send the notification

4.2.14.2.2.1 Test Sequence N°1 – Nominal Case: No Follow-up Activities

Initial Conditions

- No POL1 defined in the previous Enabled ISD-P (i.e. `#ISD_P_AID1)`
- CAT_TP Connectivity Parameters have been set on `#ISD_R_AID` with `#UDP_PORT`, `#CAT_TP_PORT` and `#IP_VALUE`
- CAT_TP Connectivity Parameters have been set on `#DEFAULT_ISD_P_AID` with `#BEARER_DESCRIPTION`, `#NAN_VALUE`, `#LOGIN` and `#PWD`

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to #BEARER_DESCRIPTION 2- The NAN is equal to #NAN_VALUE 3- The port is equal to #UDP_PORT 4- The IP is equal to #IP_VALUE 5- The login/password are equal to #LOGIN/#PWD	EUICC_REQ18, EUICC_REQ27, PROC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The CAT_TP PDU used here after shall be compliant with the Annex G.*				
7	eUICC-UT → DS	SYN	The identification data may contain the #EID	EUICC_REQ18, PROC_REQ22
8	DS → eUICC-UT	SYN_ACK		PROC_REQ22
9	eUICC-UT → DS	ACK_NO_DATA	The CAT_TP session is open.	EUICC_REQ18, PROC_REQ22
10	eUICC-UT → DS	ACK_DATA containing the notification	1- The ACK_DATA contains a command packet 2- The SPI is equal to #SPI_NOTIF 3- Verify the cryptographic checksum using #SCP80_AUTH_KEY 4- The secured data shall only contain the TLV #NOTIF_PROFILE_CHANGE2 26. (see Note 1) 5- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ18, EUICC_REQ27, EUICC_REQ54, PROC_REQ22
11	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [NOTIF_CONFIRMATION])		PROC_REQ22, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
12	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 27.#SCP80_AUTH_KEY 4- The response data is equal to [R_AB_NOTIF]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ29, PROC_REQ22
13	Close CAT_TP session as described in section 4.2.1.4			
14	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])		EUICC_REQ22, EUICC_REQ54
15	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
16	DS → eUICC-UT	FETCH		
17	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP_LIST2]	PM_REQ3, PM_REQ4, PF_REQ5, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PROC_REQ22
18	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification.

Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.

Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).

4.2.14.2.3 TC.ES5.NOTIFPD.3: Notification_HTTPS

Test Purpose

To ensure HTTPS notification procedure is well implemented when a Profile is Disabled.

Note: As the update of the lifecycle states may become effective after the REFRESH command, the check of the lifecycle states of the Profiles is performed in this test case (the ISD-P with the Fall-back Attribute Set shall be Enabled).

Referenced Requirements

- PF_REQ5, PF_REQ7
- PM_REQ3, PM_REQ4
- PROC_REQ21, PROC_REQ22
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS mode is the default way (priority order 1) to send the notification
- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: #PSK_ID
    - PSK value: #SCP81_PSK

4.2.14.2.3.1 Test Sequence N°1 – Nominal Case: No Follow-up Activities

Initial Conditions

- No POL1 defined in the previous Enabled ISD-P (i.e. #ISD_P_AID1)
- HTTPS Connectivity Parameters have been set on #ISD_R_AID with #TCP_PORT, #IP_VALUE, #ADMIN_HOST, #AGENT_ID, #PSK_ID, #SCP81_KVN, #SCP81_KEY_ID and #ADMIN_URI
- HTTPS Connectivity Parameters have been set on #DEFAULT_ISD_P_AID with #BEARER_DESCRIPTION, #NAN_VALUE, #LOGIN and #PWD

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by eUICC	
2	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to `#BEARER_DESCRIPTION` 2- The NAN is equal to `#NAN_VALUE` 3- The port is equal to `#TCP_PORT` 4- The IP is equal to `#IP_VALUE` 5- The login/password are equal to `#LOGIN/#PWD`	EUICC_REQ13, EUICC_REQ14, PROC_REQ21, PROC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE		
For readability reason, the proactive commands are not fully specified in the next steps. *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The TLS records used here after shall be compliant with the 0.*				
7	eUICC-UT → DS	TLS_CLIENT_HELLO	The CLIENT_HELLO shall contain at least one of the cipher-suites accepted by the HTTPS server.	EUICC_REQ14, EUICC_REQ43, PROC_REQ21, PROC_REQ22
8	DS → eUICC-UT	TLS_SERVER_HELLO and TLS_SERVER_HELLO_DONE		PROC_REQ21, PROC_REQ22
9	eUICC-UT → DS	TLS_CLIENT_KEY_EXCHANGE and TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED	The CLIENT_KEY_EXCHANGE shall contain the `#PSK_ID`	EUICC_REQ14, EUICC_REQ43, EUICC_REQ45, PROC_REQ21, PROC_REQ22
10	DS → eUICC-UT	TLS_CHANGE_CIPHER_SPEC and TLS_FINISHED		PROC_REQ21, PROC_REQ22
11	eUICC-UT → DS	TLS_APPLICATION with the first POST message	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake 2- The HTTP content is empty 28. The POST URI is equal to `#POST_URI_NOTIF2` (see Note 1) 3- The headers are equal to 129.`#HOST` 130.`#X_ADMIN_PROTOCOL` 131. `#X_ADMIN_FROM_ISD_R`	EUICC_REQ14, EUICC_REQ27, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, PROC_REQ21, PROC_REQ22

Step	Direction	Sequence / Description	Expected result	REQ

12	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[NOTIF_CONFIRMATION])`
				EUICC_REQ29, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, PROC_REQ21, PROC_REQ22

13	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			32. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			33. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_NOTIF]`
				EUICC_REQ14, EUICC_REQ16, EUICC_REQ29, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, PROC_REQ21, PROC_REQ22

14	Close HTTPS session as described in section 4.2.1.7			

15	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1];`
		`[GET_DEFAULT_ISDP])`
				EUICC_REQ22, EUICC_REQ54

16	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
17	DS → eUICC-UT	FETCH		
18	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP_LIST2]`	PM_REQ3, PM_REQ4, PF_REQ5, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PROC_REQ22
19	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Step	Direction	Sequence / Description	Expected result	REQ
Note 1: The tag '14' (or '94') with the IMEI value and the tag '6D' (or 'ED') with the MEID provided in the TERMINAL RESPONSE(PROVIDE LOCAL INFORMATION) sent during the toolkit initialization process may be also present in the notification. *Note 2: It is assumed that some proactive commands may be sent by the eUICC after sending the TERMINAL PROFILE (i.e. SET UP EVENT LIST, POLL INTERVAL, PROVIDE LOCAL INFORMATION…). In this case, the DS shall send the corresponding FETCH and TERMINAL RESPONSE(successfully performed) commands.* *Note 3: Depending on the implementation, it may be necessary to send an ENVELOPE (EVENT DOWNLOAD - Location status) indicating "normal service" (i.e. '00') in order to trigger the sending of the eUICC notification. This envelope shall be sent only if this event (i.e. encoded with the value '03') is present in the SET UP EVENT LIST sent by the eUICC. Moreover, the eUICC may also wait for several STATUS events before issuing the notification (within a maximum time interval of 10 STATUS events).*				

4.2.15 ES6 (MNO – eUICC): UpdatePOL1byMNO

4.2.15.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

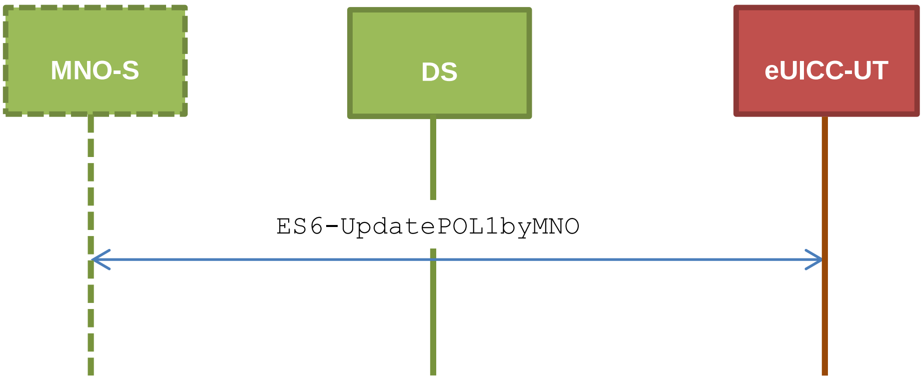
- PM_REQ6
- PROC_REQ17
- EUICC_REQ7, EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ43, EUICC_REQ48, EUICC_REQ52

4.2.15.2 Test Cases

General Initial Conditions

- None

Test Environment



4.2.15.2.1 TC.ES6.UPOL1MNO.1: UpdatePOL1byMNO_SMS

Test Purpose

To ensure MNO can update POL1 on the eUICC using SMS. Some error cases due to inconsistent values in commands are also defined.

Referenced Requirements

- PM_REQ6
- PROC_REQ17
- EUICC_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

4.2.15.2.1.1 Test Sequence N°1 – Nominal Case: No Rule

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`[INSTALL_PERSO_RES_ISDP];`
		`[STORE_POL1_NO_RULE])`
		Use `#MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY`
				EUICC_REQ22, PROC_REQ17

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_029000]`	PM_REQ6, PROC_REQ17, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.15.2.1.2 Test Sequence N°2 – Nominal Case: Disabling Not Allowed

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_PERSO_RES_ISDP]; [STORE_POL1_DIS]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22, PROC_REQ17
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_029000]	PM_REQ6, PROC_REQ17, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.15.2.1.3 Test Sequence N°3 – Nominal Case: Deletion and Disabling Not Allowed

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_PERSO_RES_ISDP]; [STORE_POL1_DEL_DIS]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22, PROC_REQ17
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_029000]	PM_REQ6, PROC_REQ17, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.15.2.1.4 Test Sequence N°4 – Nominal Case: Delete when Disabled

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_PERSO_RES_ISDP]; [STORE_POL1_DEL_AUTO]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22, PROC_REQ17
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_029000]	PM_REQ6, PROC_REQ17, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.15.2.1.5 Test Sequence N°5 – Error Case: Bad POL1 Value

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_PERSO_RES_ISDP]; [BAD_STORE_POL1]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22, PROC_REQ17
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_026A80]	PM_REQ6, PROC_REQ17, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.15.2.1.6 Test Sequence N°6 – Error Case: Associated ISD-P Not Enabled

⚠ This test case is defined as FFS pending further clarification in the GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2].

4.2.15.2.2 TC.ES6.UPOL1MNO.2: UpdatePOL1byMNO_CAT_TP

Test Purpose

To ensure MNO can update POL1 on the eUICC using CAT_TP.

Referenced Requirements

- PM_REQ6
- PROC_REQ17
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ18, EUICC_REQ22

Initial Conditions

- None

4.2.15.2.2.1 Test Sequence N°1 – Nominal Case: No Rule

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on MNO-SD as described in section 4.2.1.3			

3	DS → eUICC-UT	ACK_DATA containing the result of `SCP80_PACKET(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`[INSTALL_PERSO_RES_ISDP];`
		`[STORE_POL1_NO_RULE])`
		Use `#MNO_SCP80_ENC_KEY,` `#MNO_SCP80_AUTH_KEY,` `#MNO_SCP80_DATA_ENC_KEY`
				PROC_REQ17

4	eUICC-UT → DS	ACK_DATA with POR	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_029000]`	PM_REQ6, PROC_REQ17, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
5	Close CAT_TP session as described in section 4.2.1.4			

4.2.15.2.3 TC.ES6.UPOL1MNO.3: UpdatePOL1byMNO_HTTPS

Test Purpose

To ensure MNO can update POL1 on the eUICC using HTTPS.

Referenced Requirements

- PM_REQ6
- PROC_REQ17
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ22, EUICC_REQ43, EUICC_REQ48, EUICC_REQ52

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#MNO_PSK_ID`
    - PSK value: `#MNO_SCP81_PSK`

4.2.15.2.3.1 Test Sequence N°1 – Nominal Case: No Rule

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on MNO-SD as described in section 4.2.1.6			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[INSTALL_PERSO_RES_ISDP];`
		`[STORE_POL1_NO_RULE])`
				PROC_REQ17

4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the `#MNO_SCP81_PSK` using the cipher-suite negotiated during the TLS handshake
			2- The POST URI is equal to `#POST_URI`
			3- The different headers are equal to
			34. `#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_MNO`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			35. `#X_ADMIN_STATUS_OK`
			4- The HTTP content contains a response data equal to `[R_AF_029000]`
				PM_REQ6,
				PROC_REQ17,
				EUICC_REQ14,
				EUICC_REQ16,
				EUICC_REQ43,
				EUICC_REQ48,
				EUICC_REQ52

5	Close HTTPS session as described in section 4.2.1.7			

4.2.16 ES6 (MNO – eUICC): UpdateConnectivityParametersByMNO

4.2.16.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

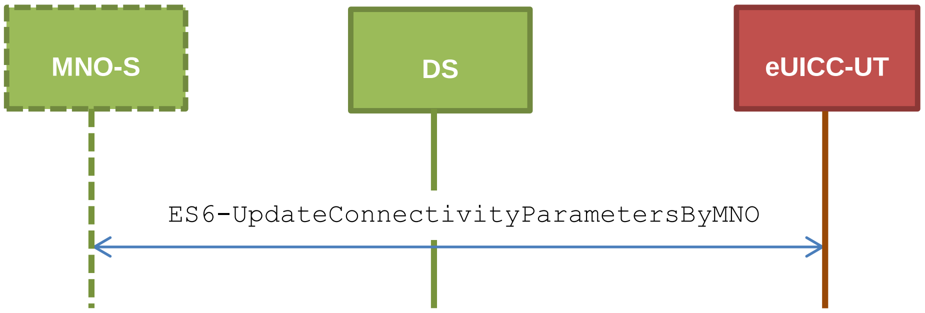
- PM_REQ7
- PROC_REQ18
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

4.2.16.2 Test Cases

General Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)

Test Environment



4.2.16.2.1 TC.ES6.UCPMNO.1: UpdateConnectParamByMNO_SMS

Test Purpose

To ensure MNO can update the Connectivity Parameters on the eUICC using SMS.

Referenced Requirements

- PM_REQ7
- PROC_REQ18
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

4.2.16.2.1.1 Test Sequence N°1 – Nominal Case: Update SMS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`[INSTALL_PERSO_RES_ISDP];`
		`[STORE_SMS_PARAM_MNO])`
		Use `#MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY`
				EUICC_REQ22, PROC_REQ18

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_029000]`	PM_REQ7, PROC_REQ18, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.16.2.1.2 Test Sequence N°2 – Nominal Case: Update CAT_TP Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`[INSTALL_PERSO_RES_ISDP];`
		`[STORE_CATTP_PARAM_MNO])`
		Use `#MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY`
				EUICC_REQ22,
				PROC_REQ18

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_029000]`	PM_REQ7, PROC_REQ18, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.16.2.1.3 Test Sequence N°3 – Nominal Case: Update HTTPS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_PERSO_RES_ISDP]; [STORE_HTTPS_PARAM_MNO]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22, PROC_REQ18
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_029000]	PM_REQ7, PROC_REQ18, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.17 ES8 (SM-DP – eUICC): EstablishISDPKeySet

4.2.17.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

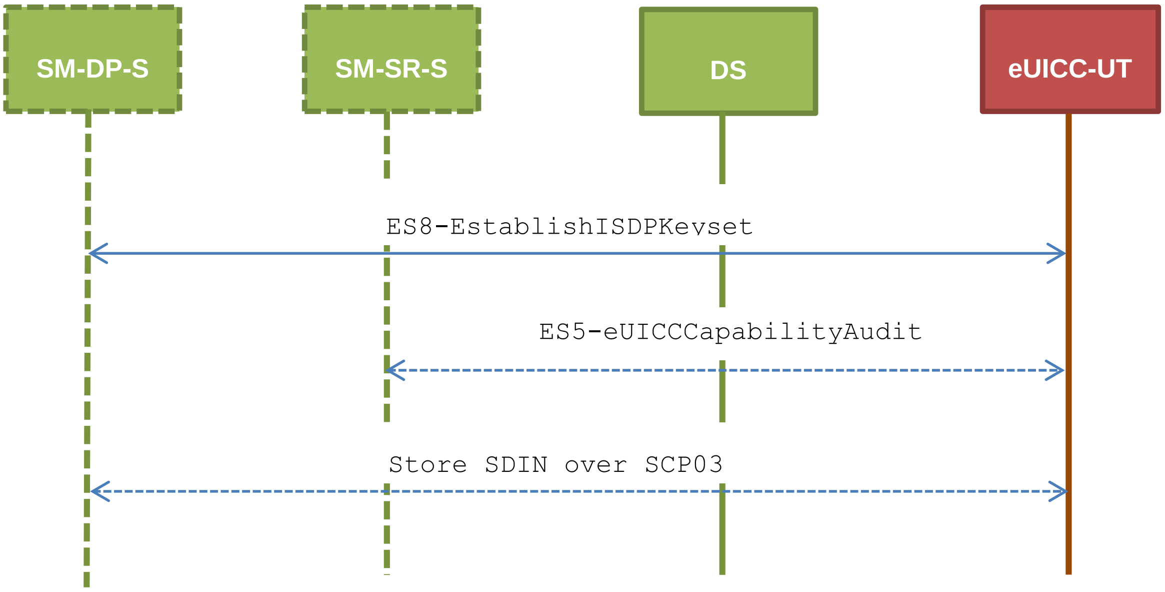
- PF_REQ7
- PM_REQ8
- EUICC_REQ5, EUICC_REQ13, EUICC_REQ14, EUICC_REQ15, EUICC_REQ17, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ51, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

4.2.17.2 Test Cases

General Initial Conditions

- #ISD_P_AID1 present on the eUICC
- #ISD_P_AID1 in SELECTABLE state

Test Environment



4.2.17.2.1 TC.ES8.EISDPK.1: EstablishISDPKeyset_SMS

Test Purpose

To ensure the ISD-P keyset establishment process is well implemented on the eUICC using SMS. After ISD-P SCP03 keys initialization, the lifecycle state of the ISD-P is checked (shall be PERSONALIZED) and a new secure channel session is opened to make sure that the new keys have been set. During the key establishment, different parameters are used (DR, HostID) to make sure that all configurations are supported on the eUICC. An error case is defined to test that an incorrect SM-DP certificate is rejected.

Referenced Requirements

- PF_REQ7
- PM_REQ8
- EUICC_REQ5, EUICC_REQ13, EUICC_REQ15, EUICC_REQ17, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ54

Initial Conditions

- None

4.2.17.2.1.1 Test Sequence N°1 – Nominal Case: No DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_PERSO_ISDP1];`
		`[STORE_DP_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_03RC]` 4- Retrieve the `{RC}`	PM_REQ8, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`STORE_ISDP_KEYS(`
		`#SC3_NO_DR;`
		`{RC}),`
		`#LAST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RECEIPT]` 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tag ‘A6’)	PM_REQ8, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW=’9000’	

17	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

18	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
19	DS → eUICC-UT	FETCH		
20	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_0F]`	PF_REQ7, PM_REQ8, EUICC_REQ5, EUICC_REQ13, EUICC_REQ15, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
21	DS → eUICC-UT	TERMINAL RESPONSE	SW=’9000’	

22	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03_SCRIPT(`
		`#SCP03_KVN,`
		`[STORE_SDIN]))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
				EUICC_REQ17,
				EUICC_REQ54

23	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		

Step	Direction	Sequence / Description	Expected result	REQ
24	DS → eUICC-UT	FETCH		
25	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- No SCP03 security error is raised in the response data (i.e. INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed)	EUICC_REQ19, EUICC_REQ21, EUICC_REQ23
26	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.17.2.1.2 Test Sequence N°2 – Nominal Case: DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_PERSO_ISDP1];`
		`[STORE_DP_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_03RC]` 4- Retrieve the `{RC}`	PM_REQ8, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Step	Direction	Sequence / Description	Expected result	REQ

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`STORE_ISDP_KEYS(`
		`#SC3_DR;`
		`{RC}),`
		`#LAST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RECEIPT_DR]` 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS and `{DR}` and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tags ‘A6’ and ‘85’)	PM_REQ8, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ54
11	DS → eUICC-UT	TERMINAL RESPONSE	SW=’9000’	

12	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_0F]`	PF_REQ7, PM_REQ8, EUICC_REQ5, EUICC_REQ13, EUICC_REQ15, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

17	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03_SCRIPT(`
		`#SCP03_KVN,`
		`[STORE_SDIN]))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
				EUICC_REQ17,
				EUICC_REQ54

18	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
19	DS → eUICC-UT	FETCH		
20	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- No SCP03 security error is raised in the response data (i.e. INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed)	EUICC_REQ19, EUICC_REQ21, EUICC_REQ23
21	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.17.2.1.3 Test Sequence N°3 – Nominal Case: DR, Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_PERSO_ISDP1];`
		`[STORE_DP_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_03RC]` 4- Retrieve the `{RC}`	PM_REQ8, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`STORE_ISDP_KEYS(`
		`#SC3_DR_HOST;`
		`{RC}),`
		`#LAST_SCRIPT)`
				EUICC_REQ22,
				EUICC_REQ54

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_02RECEIPT_DR]` 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS (using `{DR}`, `#HOST_ID`, `#ISD_R_SIN` and `#ISD_R_SDIN)` and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tags 'A6' and '85')	PM_REQ8, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

12	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

13	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
14	DS → eUICC-UT	FETCH		
15	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_0F]`	PF_REQ7, PM_REQ8, EUICC_REQ5, EUICC_REQ13, EUICC_REQ15, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
16	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

17	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03_SCRIPT(`
		`#SCP03_KVN,`
		`[STORE_SDIN]))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
				EUICC_REQ17,
				EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
18	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
19	DS → eUICC-UT	FETCH		
20	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- No SCP03 security error is raised in the response data (i.e. INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed)	EUICC_REQ19, EUICC_REQ21, EUICC_REQ23
21	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.17.2.1.4 Test Sequence N°4 – Error Case: Invalid SM-DP Certificate

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_PERSO_ISDP1];`
		`[STORE_INVALID_DP_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_036982]` (see Note)	PM_REQ8, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: The SW may be also '6A80'				

4.2.17.2.2 TC.ES8.EISDPK.2: EstablishISDPKeyset_CAT_TP

Test Purpose

To ensure the ISD-P keyset establishment process is well implemented on the eUICC using CAT_TP. After ISD-P SCP03 keys initialization, the lifecycle state of the ISD-P is checked (shall be PERSONALIZED) and a new secure channel session is opened to make sure that the new keys have been set.

Referenced Requirements

- PF_REQ7
- PM_REQ8
- EUICC_REQ5, EUICC_REQ13, EUICC_REQ15, EUICC_REQ17, EUICC_REQ18, EUICC_REQ22, EUICC_REQ23, EUICC_REQ53, EUICC_REQ54

Initial Conditions

- None

4.2.17.2.2.1 Test Sequence N°1 – Nominal Case: No DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

3	DS → eUICC-UT	ACK_DATA containing the result of `SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[INSTALL_PERSO_ISDP1];`
		`[STORE_DP_CERTIF],`
		`#FIRST_SCRIPT)`
				EUICC_REQ54

4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `36. #SCP80_AUTH_KEY` 4- The response data is equal to `[R_AB_03RC]` 5- Retrieve the `{RC}`	PM_REQ8, EUICC_REQ13, EUICC_REQ18

Step	Direction	Sequence / Description	Expected result	REQ
5	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, STORE_ISDP_KEYS(#SC3_NO_DR; {RC}), #LAST_SCRIPT)		EUICC_REQ54
6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 37. #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_02RECEIPT] 5- Calculate ShS from #SM_ESK_ECKA and #PK_ECASD_ECKA 6- Derive keyset from ShS and retrieve the {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK} 7- Verify the {RECEIPT} (i.e. it shall be generated by calculating a MAC across the tag 'A6')	PM_REQ8, EUICC_REQ13, EUICC_REQ18
7	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ54
8	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E3_ISDP1_0F]	PF_REQ7, EUICC_REQ5, EUICC_REQ13, EUICC_REQ15, EUICC_REQ18

Step	Direction	Sequence / Description	Expected result	REQ

9	DS → eUICC-UT	ACK_DATA containing the result of `SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03_SCRIPT(`
		`#SCP03_KVN,`
		`[STORE_SDIN]))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
			38.	EUICC_REQ17, EUICC_REQ54

10	eUICC-UT → DS	ACK_DATA with POR	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- No SCP03 security error is raised in the response data (i.e. INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed)	EUICC_REQ18, EUICC_REQ23
11	Close CAT_TP session as described in section 4.2.1.4			

4.2.17.2.3 TC.ES8.EISDPK.3: EstablishISDPKeyset_HTTPS

Test Purpose

To ensure the ISD-P keyset establishment process is well implemented on the eUICC using HTTPS. After ISD-P SCP03 keys initialization, the lifecycle state of the ISD-P is checked (shall be PERSONALIZED) and a new secure channel session is opened to make sure that the new keys have been set.

Referenced Requirements

- PF_REQ7
- PM_REQ8
- EUICC_REQ5, EUICC_REQ13, EUICC_REQ14, EUICC_REQ15, EUICC_REQ17, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ51, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.17.2.3.1 Test Sequence N°1 – Nominal Case: No DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([INSTALL_PERSO_ISDP1]; [STORE_DP_CERTIF])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 39. #HOST 40. #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 41. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_02RC] 5- Retrieve the {RC}	PM_REQ8, EUICC_REQ14, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT(STORE_ISDP_KEYS(#SC3_NO_DR; {RC}))		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 42. #HOST 43. #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 44. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_RECEIPT] 5- Calculate ShS from #SM_ESK_ECKA and #PK_ECASD_ECKA 6- Derive keyset from ShS and retrieve the {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK} 7- Verify the {RECEIPT} (i.e. it shall be generated by calculating a MAC across the tag 'A6')	PM_REQ8, EUICC_REQ14, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
7	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_ISDP1])		EUICC_REQ49, EUICC_REQ50, EUICC_REQ52
8	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 45. #HOST 46. #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 47. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP1_0F]	PF_REQ7, PM_REQ8, EUICC_REQ5, EUICC_REQ14, EUICC_REQ15, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52

Step	Direction	Sequence / Description	Expected result	REQ
9	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP(#ISD_P_AID1 SCP03_SCRIPT(#SCP03_KVN, [STORE_SDIN])) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK})		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52
10	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 48. #HOST 49. #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 50. #X_ADMIN_STATUS_OK 4- No SCP03 security error is raised in the response data (i.e. INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed)	EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
11	Close HTTPS session as described in section 4.2.1.7			

4.2.18 ES8 (SM-DP – eUICC): DownloadAndInstallation

4.2.18.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ7
- PM_REQ3, PM_REQ9
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ17, EUICC_REQ18, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61

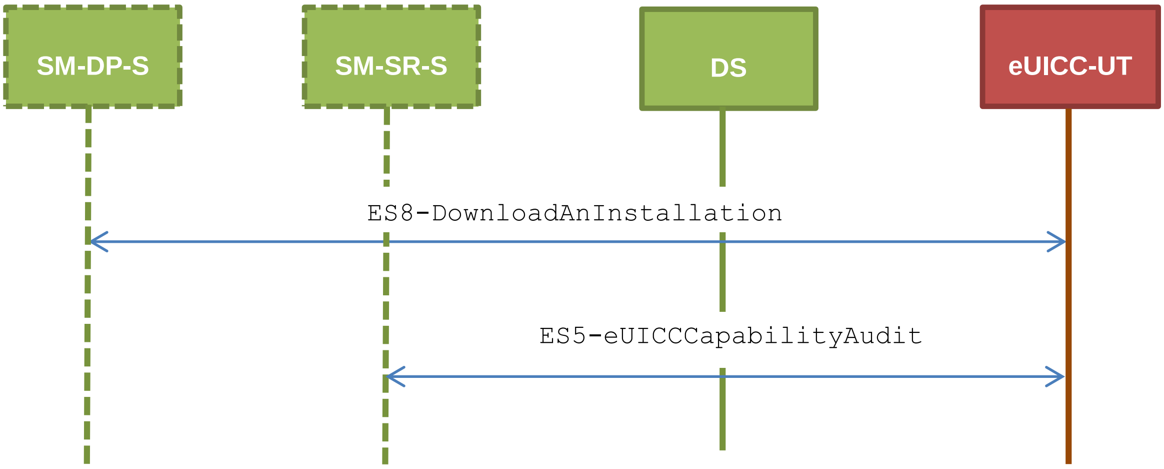
- SEC_REQ23

4.2.18.2 Test Cases

General Initial Conditions

- `#ISD_P_AID1` present on the eUICC and personalized with SCP03 keys
  - The process *ES8-EstablishISDPKeySet* has been used
  - `{SCP_KENC}`, `{SCP_KMAC}`, `{SCP_KDEK}` have been set

Test Environment



4.2.18.2.1 TC.ES8.DAI.1: DownloadAndInstallation_CAT_TP

Test Purpose

To ensure Profile download is possible on the eUICC using CAT_TP. A generic Profile is downloaded and script chaining, as defined in ETSI TS 102 226 [6], is used in this sequence. After the execution of the download process, an audit is sent to make sure that the new Profile is Disabled. An error case is also defined to check that the ISD-P lifecycle state remains unchanged when the Profile is not fully downloaded.

Referenced Requirements

- PF_REQ7
- PM_REQ3, PM_REQ9
- EUICC_REQ13, EUICC_REQ17, EUICC_REQ18, EUICC_REQ22, EUICC_REQ23, EUICC_REQ53, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61
- SEC_REQ23

Initial Conditions

- None

4.2.18.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The #PROFILE_PACKAGE shall be split in several parts named from {PROFILE_PART1} to {PROFILE_PARTn} in this sequence (n = the last index of the sub part). Each Profile part contains a list of PEs.

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_P_TAR1, SCP03T_SCRIPT(#SCP03_KVN, {PROFILE_PART1}), #FIRST_SCRIPT) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		EUICC_REQ17, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 51.#SCP80_AUTH_KEY 4- The response data is formatted in an expanded remote command structure with definite length coding 5- The response to the INITIALIZE UPDATE TLV command (i.e. TAG '84') shall be equal to [R_SCP03T_INITUP_OK] 6- The response to the EXTERNAL AUTHENTICATE TLV command (i.e. TAG '85') shall be equal to [R_SCP03T_EXTAUTH_OK] 7- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned	PM_REQ9, EUICC_REQ13, EUICC_REQ18, EUICC_REQ23, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61
Loop until the Profile part index (named i) is equal to n-1				

Step	Direction	Sequence / Description	Expected result	REQ

5	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03T_SUB_SCRIPT(`
		`{PROFILE_PARTi}),`
		`#SUB_SCRIPT)`
				EUICC_REQ17,
				EUICC_REQ54,
				EUICC_REQ57,
				EUICC_REQ58

6	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the `#SCP80_ENC_KEY` 3- Verify the cryptographic checksum using `52.#SCP80_AUTH_KEY` 4- The response data is formatted in an expanded remote command structure with definite length coding 5- For each SCP03t TLV command sent (i.e. TAG '86'), a response `[R_SCP03T_EMPTY]` is returned	PM_REQ9, EUICC_REQ13, EUICC_REQ18, EUICC_REQ23, EUICC_REQ61
End loop				

7	DS → eUICC-UT	ACK_DATA containing the result of
		`SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03T_SUB_SCRIPT(`
		`{PROFILE_PARTn}),`
		`#LAST_SCRIPT)`
				EUICC_REQ17,
				EUICC_REQ54,
				EUICC_REQ57,
				EUICC_REQ58

Step	Direction	Sequence / Description	Expected result	REQ
8	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 53. #SCP80_AUTH_KEY 4- The response data is formatted in an expanded remote command structure with definite length coding 5- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned (except for the last one) 6- Decrypt the last SCP03t response using the SCP03 session key and check the R-MAC 7- The content of the last SCP03t response data is equal to #R_PROF_PKG_OK	PM_REQ9, EUICC_REQ13, EUICC_REQ18, EUICC_REQ23, EUICC_REQ61, SEC_REQ23
9	Close CAT_TP session as described in section 4.2.1.4			
10	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
11	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
12	DS → eUICC-UT	FETCH		
13	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_1F]	PF_REQ7, PM_REQ3, EUICC_REQ13, EUICC_REQ22
14	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.18.2.1.2 Test Sequence N°2 – Error Case: Profile Downloading Interrupted

Initial Conditions

- The #PROFILE_PACKAGE shall be split in several parts named from {PROFILE_PART1} to {PROFILE_PARTn} in this sequence (n = the last index of the sub part). Each Profile part contains a list of PEs. Note that only the {PROFILE_PART1} needs to be sent in the following test.

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_P_TAR1, SCP03T_SCRIPT(#SCP03_KVN, {PROFILE_PART1}), #FIRST_SCRIPT) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		EUICC_REQ17, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- Verify the cryptographic checksum using 54.#SCP80_AUTH_KEY 4- The response data is formatted in an expanded remote command structure with definite length coding 5- The response to the INITIALIZE UPDATE TLV command (i.e. TAG '84') shall be equal to [R_SCP03T_INITUP_OK] 6- The response to the EXTERNAL AUTHENTICATE TLV command (i.e. TAG '85') shall be equal to [R_SCP03T_EXTAUTH_OK] 7- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned	PM_REQ9, EUICC_REQ13, EUICC_REQ18, EUICC_REQ23, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61
5	Close CAT_TP session as described in section 4.2.1.4 (the other Profile Elements shall not be sent)			

Step	Direction	Sequence / Description	Expected result	REQ

6	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP1])`
				EUICC_REQ22,
				EUICC_REQ54

7	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
8	DS → eUICC-UT	FETCH		
9	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_0F]`	PF_REQ7, PM_REQ3, EUICC_REQ13, EUICC_REQ22
10	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.18.2.2 TC.ES8.DAI.2: DownloadAndInstallation_HTTPS

Test Purpose

To ensure Profile download is possible on the eUICC using HTTP. A generic Profile is downloaded and script chaining, as defined in ETSI TS 102 226 [6], is used in this sequence. After the execution of the download process, an audit is sent to make sure that the new Profile is Disabled. An error case is also defined to check that the ISD-P lifecycle state remains unchanged when the Profile is not fully downloaded.

Referenced Requirements

- PF_REQ7
- PM_REQ3, PM_REQ9
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ17, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61
- SEC_REQ23

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

4.2.18.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The #PROFILE_PACKAGE shall be split in several parts named from {PROFILE_PART1} to {PROFILE_PARTn} in this sequence (n = the last index of the sub part). Each Profile part contains a list of PEs.

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP(#ISD_P_AID1, SCP03T_SCRIPT(#SCP03_KVN, {PROFILE_PART1})) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 55. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 56. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data formatted in an expanded remote command structure with indefinite length coding 5- The response to the INITIALIZE UPDATE TLV command (i.e. TAG '84') shall be equal to [R_SCP03T_INITUP_OK] 6- The response to the EXTERNAL AUTHENTICATE TLV command (i.e. TAG '85') shall be equal to [R_SCP03T_EXTAUTH_OK] 7- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned	PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61
Loop until the Profile part index (named i) is equal to n-1				
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP(#ISD_P_AID1, SCP03T_SUB_SCRIPT({PROFILE_PARTi}))		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58

Step	Direction	Sequence / Description	Expected result	REQ
6	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 57.#HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 58.#X_ADMIN_STATUS_OK 4- The HTTP content contains a response data formatted in an expanded remote command structure with indefinite length coding 5- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned	PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ61, SEC_REQ23
End loop				
7	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP(#ISD_P_AID1, SCP03T_SUB_SCRIPT({PROFILE_PARTn}))		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58

Step	Direction	Sequence / Description	Expected result	REQ
8	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 59. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 60. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data formatted in an expanded remote command structure with indefinite length coding 5- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned (except for the last one) 6- Decrypt the last SCP03t response using the SCP03 session key and check the R-MAC 7- The content of the last SCP03t response data is equal to #R_PROF_PKG_OK	PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ61
9	Close HTTPS session as described in section 4.2.1.7			
10	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
11	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
12	DS → eUICC-UT	FETCH		
13	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- Verify the cryptographic checksum using #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_E3_ISDP1_1F]	PF_REQ7, PM_REQ3, EUICC_REQ13, EUICC_REQ22
14	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.18.2.2.2 Test Sequence N°2 – Error Case: Profile Downloading Interrupted

Initial Conditions

- The #PROFILE_PACKAGE shall be split in several parts named from {PROFILE_PART1} to {PROFILE_PARTn} in this sequence (n = the last index of the sub part). Each Profile part contains a list of PEs. Note that only the {PROFILE_PART1} needs to be sent in the following test.

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

3	DS → eUICC-UT	TLS_APPLICATION containing the result of
		HTTPS_CONTENT_ISDP(
		    #ISD_P_AID1,
		    SCP03T_SCRIPT(
		        #SCP03_KVN,
		        {PROFILE_PART1}))
		Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}
				EUICC_REQ17,
				EUICC_REQ49,
				EUICC_REQ51,
				EUICC_REQ52,
				EUICC_REQ57,
				EUICC_REQ58

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 61. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 62. #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data formatted in an expanded remote command structure with indefinite length coding 5- The response to the INITIALIZE UPDATE TLV command (i.e. TAG '84') shall be equal to [R_SCP03T_INITUP_OK] 6- The response to the EXTERNAL AUTHENTICATE TLV command (i.e. TAG '85') shall be equal to [R_SCP03T_EXTAUTH_OK] 7- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned	PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61
5	Close HTTPS session as described in section 4.2.1.7 (the other Profile Elements shall not be sent)			
6	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
7	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
8	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
9	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_E3_ISDP1_0F]`	PF_REQ7, PM_REQ3, EUICC_REQ13, EUICC_REQ22
10	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.19 ES8 (SM-DP – eUICC): UpdateConnectivityParameters

4.2.19.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

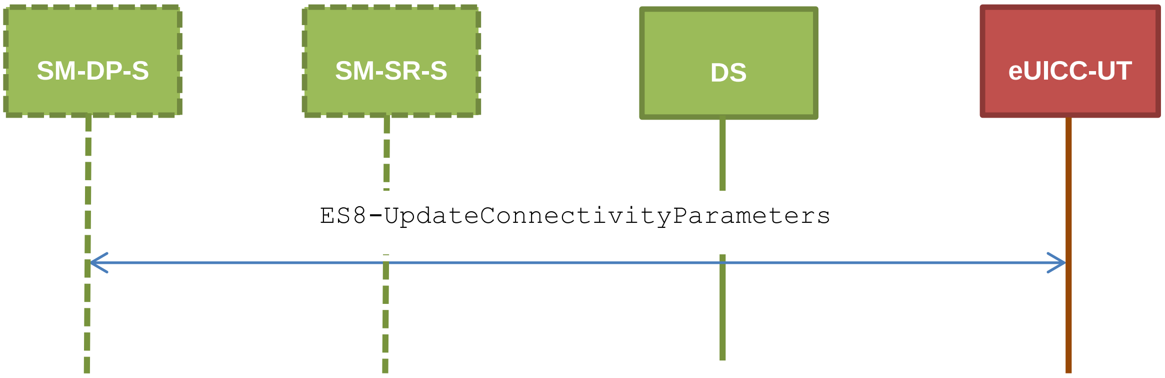
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ17, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ54

4.2.19.2 Test Cases

General Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)

Test Environment



4.2.19.2.1 TC.ES8.UCP.1: UpdateConnectivityParameters_SMS

Test Purpose

To ensure ISD-P can update the Connectivity Parameters on an Enabled Profile using SMS.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ17, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31, EUICC_REQ54

Initial Conditions

- None

4.2.19.2.1.1 Test Sequence N°1 – Nominal Case: Update SMS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#DEFAULT_ISD_P_TAR,`
		`SCP03_SCRIPT(`
		`#DEFAULT_ISD_P_SCP03_KVN,`
		`[STORE_SMS_PARAM_MNO]))`
				EUICC_REQ17,
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- For each R-APDU received: a. SW='9000' or '6108'	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.19.2.1.2 Test Sequence N°2 – Nominal Case: Update CAT_TP Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#DEFAULT_ISD_P_TAR,`
		`SCP03_SCRIPT(`
		`#DEFAULT_ISD_P_SCP03_KVN,`
		`[STORE_CATTP_PARAM_MNO]))`
				EUICC_REQ17,
				EUICC_REQ22,
				EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- For each R-APDU received: a. SW='9000' or '6108'	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.19.2.1.3 Test Sequence N°3 – Nominal Case: Update HTTPS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#DEFAULT_ISD_P_TAR,`
		`SCP03_SCRIPT(`
		`#DEFAULT_ISD_P_SCP03_KVN,`
		`[STORE_HTTPS_PARAM_MNO]))`
				EUICC_REQ17,
				EUICC_REQ22,
				EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- For each R-APDU received: a. SW='9000' or '6108'	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.19.2.1.4 Test Sequence N°4 – Nominal Case: Update SMS and CAT_TP Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#DEFAULT_ISD_P_TAR,`
		`SCP03_SCRIPT(`
		`#DEFAULT_ISD_P_SCP03_KVN,`
		`[STORE_SMSCATTP_PARAM]))`
				EUICC_REQ17, EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- For each R-APDU received: a. SW='9000' or '6108'	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.19.2.1.5 Test Sequence N°5 – Nominal Case: Update SMS and HTTPS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#DEFAULT_ISD_P_TAR,`
		`SCP03_SCRIPT(`
		`#DEFAULT_ISD_P_SCP03_KVN,`
		`[STORE_SMSHTTPS_PARAM]))`
				EUICC_REQ17, EUICC_REQ22, EUICC_REQ54

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- For each R-APDU received: a. SW='9000' or '6108'	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.19.2.2 TC.ES8.UCP.2: UpdateConnectivityParameters_CAT_TP

Test Purpose

To ensure ISD-P can update the Connectivity Parameters on a Disabled Profile using CAT_TP.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ17, EUICC_REQ18, EUICC_REQ23, EUICC_REQ31, EUICC_REQ54

Initial Conditions

- `#ISD_P_AID1` present on the eUICC and personalized with SCP03 keys
  - The process ES8-EstablishISDPKeySet has been used
  - `{SCP_KENC}, {SCP_KMAC}, {SCP_KDEK}` have been set
- `#ISD_P_AID1` in Disabled state

4.2.19.2.2.1 Test Sequence N°1 – Nominal Case: Update CAT_TP Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

3	DS → eUICC-UT	ACK_DATA containing the result of `SCP80_PACKET(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03_SCRIPT(`
		`#SCP03_KVN,`
		`[STORE_CATTP_PARAM_MNO]))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
				EUICC_REQ17, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	ACK_DATA with POR	1- The ACK_DATA contains a response packet 2- Decrypt the response packet with the #SCP80_ENC_KEY 3- For each R-APDU received: a. SW='9000' or '6108'	EUICC_REQ13, EUICC_REQ18, EUICC_REQ23, EUICC_REQ31
5	Close CAT_TP session as described in section 4.2.1.4			

4.2.19.2.3 TC.ES8.UCP.3: UpdateConnectivityParameters_HTTPS

Test Purpose

To ensure ISD-P can update the Connectivity Parameters on a Disabled Profile using HTTPS.

Referenced Requirements

- EUICC_REQ14, EUICC_REQ16, EUICC_REQ17, EUICC_REQ23, EUICC_REQ31, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ54

Initial Conditions

- #ISD_P_AID1 present on the eUICC and personalized with SCP03 keys
  - The process ES8-EstablishISDPKeySet has been used
  - {SCP_KENC}, {SCP_KMAC}, {SCP_KDEK} have been set
- #ISD_P_AID1 in Disabled state

4.2.19.2.3.1 Test Sequence N°1 – Nominal Case: Update HTTPS Parameters

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: #PSK_ID
    - PSK value: #SCP81_PSK

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			

Step	Direction	Sequence / Description	Expected result	REQ
3	DS → eUICC-UT	HTTPS_CONTENT_ISDP(#ISD_P_AID1, SCP03_SCRIPT(#SCP03_KVN, [STORE_HTTPS_PARAM_MNO])) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ54
4	eUICC-UT → DS	TLS_APPLICATION with POR	1- Decrypt the TLS record with the #SCP81_PSK using the cipher-suite negotiated during the TLS handshake 2- The POST URI is equal to #POST_URI 3- The different headers are equal to 63. #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING 64. #X_ADMIN_STATUS_OK 4- For each R-APDU received: a. SW=’9000’ or ‘6108’	EUICC_REQ14, EUICC_REQ16, EUICC_REQ23, EUICC_REQ31, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52
5	Close HTTPS session as described in section 4.2.1.7			

4.3 Off-card Interfaces

4.3.1 ES1 (EUM – SM-SR): RegisterEIS

4.3.1.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PROC_REQ14
- EUICC_REQ32
- PM_REQ14

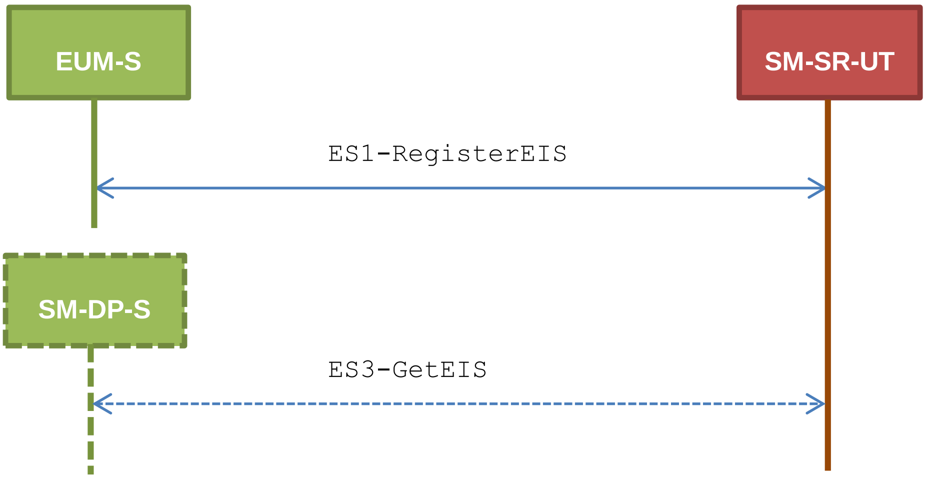
4.3.1.2 Test Cases

General Initial Conditions

- #EUM_S_ID and #EUM_S_ACCESSPOINT well known to the SM-SR-UT
- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT
- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

- #EUM_S_PK_ECDSA well known to the SM-SR-UT

Test Environment



4.3.1.2.1 TC.ES1.REIS.1: RegisterEIS

Test Purpose

To ensure EIS registration is well implemented on SM-SR. The aim is to ask the SM-SR to add a new EIS in its database and check that the new eUICC information set can be returned at any moment by the SM-SR. Some error cases are also described:

- *the EIS is already registered within the EIS database of the SM-SR*
- *the EIS signature is invalid*
- *the EIS data is invalid because the free memory is bigger than full memory*

Referenced Requirements

- PROC_REQ14
- EUICC_REQ32
- PM_REQ14

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS

4.3.1.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	EUM-S → SM-SR-UT	SEND_REQ(ES1-RegisterEIS, #EIS_ES1_RPS)		
2	SM-SR-UT → EUM-S	Send the ES1-RegisterEIS response	The Status is equal to #SUCCESS	PROC_REQ14, EUICC_REQ32

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3- GetEIS, #VIRTUAL_EID_RPS)		
4	SM-SR-UT → SM-DP-S	Send the ES3- GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES3_RPS	EUICC_REQ32, PM_REQ14

4.3.1.2.1.2 Test Sequence N°2 – Error Case: Already Registered

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is already provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	EUM-S → SM-SR-UT	SEND_REQ(ES1-RegisterEIS, #EIS_ES1_RPS)		
2	SM-SR-UT → EUM-S	Send the ES1-RegisterEIS response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EIS 3- The Reason code is equal to #RC_ALREADY_REGISTER	PROC_REQ14, EUICC_REQ32

4.3.1.2.1.3 Test Sequence N°3 – Error Case: Invalid Signature

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	EUM-S → SM-SR-UT	SEND_REQ(ES1-RegisterEIS, #EIS_BADEUMSIGN_RPS)		
2	SM-SR-UT → EUM-S	Send the ES1-RegisterEIS response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EIS 3- The Reason code is equal to #RC_INVALID_SIGN	PROC_REQ14, EUICC_REQ32

4.3.1.2.1.4 Test Sequence N°4 – Error Case: Invalid Data

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	EUM-S → SM-SR-UT	SEND_REQ(ES1-RegisterEIS, #INVALID_EIS_RPS)		
2	SM-SR-UT → EUM-S	Send the ES1-RegisterEIS response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EIS 3- The Reason code is equal to #RC_INVALID_DATA	PROC_REQ14, EUICC_REQ32

4.3.2 ES2 (MNO – SM-DP): GetEIS

4.3.2.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

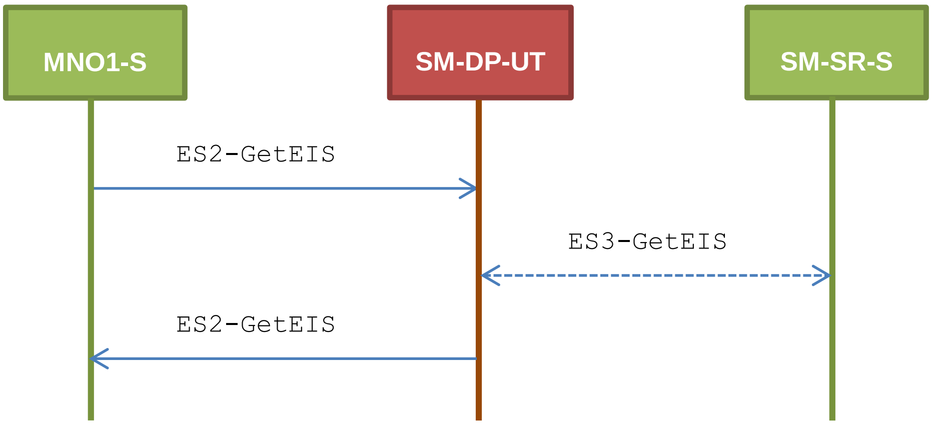
- PM_REQ10, PM_REQ14

4.3.2.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #SM_SR_S_ID and #SM_SR_S_ACCESSPOINT well known to the SM-DP-UT

Test Environment



4.3.2.2.1 TC.ES2.GEIS.1: GetEIS

Test Purpose

To ensure EIS can be retrieved by the SM-DP through the SM-SR when a MNO requests it. Some error cases are also defined:

- *the SM-SR is unknown*
- *the EID is unknown to the SM-SR*

Referenced Requirements

- PM_REQ10, PM_RE14

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_S_ID_RPS

4.3.2.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-GetEIS, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS})		
2	SM-DP-UT → SM-SR-S	Send the ES3-GetEIS request	The EID parameter is equal to #VIRTUAL_EID_RPS	PM_REQ10, PM_REQ14
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-GetEIS, #EIS_ES3_RPS)		
4	SM-DP-UT → MNO1-S	Send the ES2-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES2_RPS	PM_REQ10

4.3.2.2.1.2 Test Sequence N°2 – Error Case: Unknown SM-SR

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-GetEIS, #VIRTUAL_EID_RPS, {UNKNOWN_SM_SR_ID})		

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-DP-UT → MNO1-S	Send the `ES2- GetEIS` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_SM_SR` 3- The `Reason code` is equal to `#RC_UNKNOWN`	PM_REQ10

4.3.2.2.1.3 Test Sequence N°3 – Error Case: Unknown eUICC

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-GetEIS,`
		`#VIRTUAL_EID_RPS,`
		`{SM_SR_ID_RPS})`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-GetEIS` request	The `EID` parameter is equal to `#VIRTUAL_EID_RPS`	PM_REQ10, PM_REQ14

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-GetEIS,`
		`#FAILED,`
		`#SC_EID,`
		`#RC_ID_UNKNOWN)`		

4	SM-DP-UT → MNO1-S	Send the `ES2-GetEIS` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_EID` 3- The `Reason code` is equal to `#RC_UNKNOWN`	PM_REQ10

4.3.3 ES2 (MNO – SM-DP): DownloadProfile

4.3.3.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

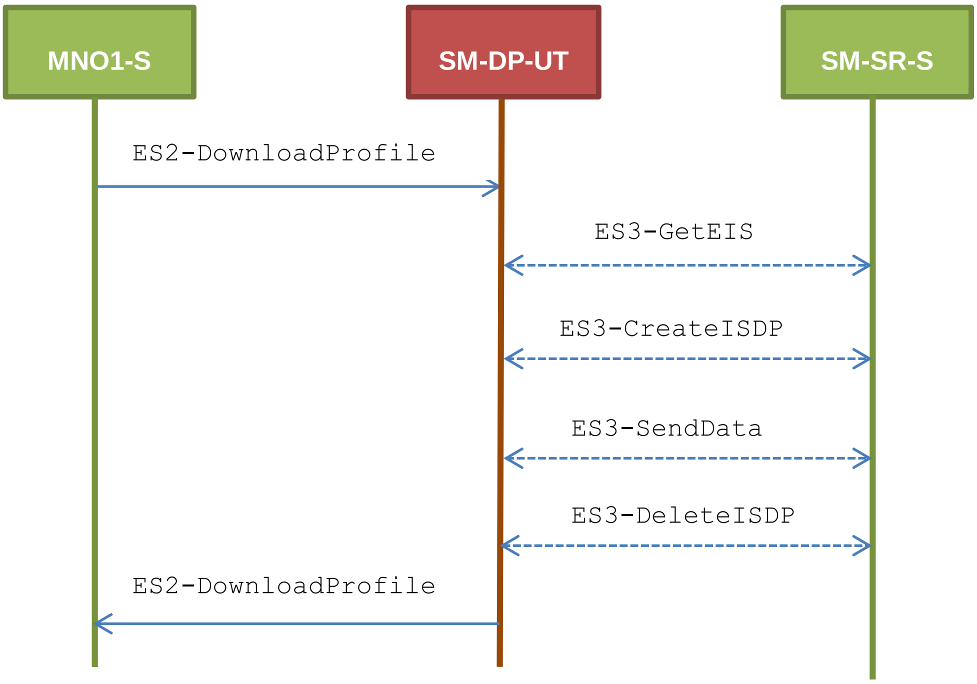
- PROC_REQ1, PROC_REQ2, PROC_REQ4
- PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17
- PF_REQ20

4.3.3.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID` and `#MNO1_S_ACCESSPOINT` well known to the SM-DP-UT
- `#SM_SR_S_ID` and `#SM_SR_S_ACCESSPOINT` well known to the SM-DP-UT
- `#EUM_S_PK_ECDSA` well known to the SM-DP-UT

Test Environment



4.3.3.2.1 TC.ES2.DP.1: DownloadProfile

Test Purpose

To ensure Profile download process is well implemented on SM-DP. The aim of the test cases defined below is to make sure that all ES3 methods are correctly sent. Only error cases are defined:

- *the keys establishment fails*
- *the ISD-P creation fails*
- *a conditional parameter is missing (neither ProfileType nor ICCID are present in the request)*

Referenced Requirements

- PROC_REQ1, PROC_REQ2, PROC_REQ4
- PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17
- PF_REQ20

Initial Conditions

- The variable `{SM_SR_ID_RPS}` shall be set to `#SM_SR_S_ID_RPS`

4.3.3.2.1.1 Test Sequence N°1 – Error Case: Keys Establishment Fails

Initial Conditions

- The Profile #PROFILE_TYPE1 linked to #ICCID1 is well known to the SM-DP-UT
- An associated Profile, as the #PROFILE_PACKAGE, is set on the SM-DP-UT
- The Profile to download shall be compatible with the #EIS_ES3_RPS (i.e. enough memory, the Profile to download is compatible with the eUICC…)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DownloadProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #PROF_TYPE1_RPS, #EP_FALSE_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-GetEIS request	The EID parameter is equal to #VIRTUAL_EID_RPS	PROC_REQ1, PM_REQ11, PM_REQ14
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-GetEIS, #EIS_ES3_RPS)		
4	SM-DP-UT → SM-SR-S	Send the ES3-CreateISDP request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID parameter is equal to #ICCID1 3- The MNO-ID parameter is equal to #MNO1_S_ID 4- The REQUIRED-MEMORY parameter is present and lower than 750000 5- The MORE-TO-DO parameter may be present. If present, it shall be equal to #MORE_TODO_RPS or #NO_MORE_TODO_RPS	PROC_REQ1, PM_REQ11, PM_REQ16
5	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-CreateISDP, #ISD_P_AID1)		

Step	Direction	Sequence / Description	Expected result	REQ
6	SM-DP-UT → SM-SR-S	Send the ES3-SendData request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The SD- AID parameter is equal to #ISD_R_AID 3- The DATA parameter is present. It shall contain APDUs related to the ES8.EstablishISDPKeyset function (i.e. STORE DATA) 4- The MORE-TO-DO parameter may be present. If present, it shall be equal to #MORE_TODO_RPS or #NO_MORE_TODO_RPS	PROC_REQ2, PM_REQ11, PM_REQ17
7	SM-SR-S → SM-DP-UT	SEND_ERROR_RESP(ES3-SendData, #FAILED, #SC_ISDP, #RC_EXECUTION_ERROR, #EUICC_RESP1_RPS)		
8	SM-DP-UT → SM-SR-S	Send the ES3-DeleteISDP request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID parameter is equal to #ICCID1	PROC_REQ4, PM_REQ11, PF_REQ20
9	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-DeleteISDP)		
10	SM-DP-UT → MNO1-S	Send the ES2-DownloadProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_ISDP 3- The Reason code is equal to #RC_EXECUTION_ERROR	PROC_REQ4, PM_REQ11

4.3.3.2.1.2 Test Sequence N°2 – Error Case: ISDP Creation Fails

Initial Conditions

- The Profile #ICCID1 is well known to the SM-DP-UT
- An associated Profile, as the #PROFILE_PACKAGE is set on the SM-DP-UT
- The Profile to download shall be compatible with the #EIS_ES3_RPS (i.e. enough memory, the Profile to download is compatible with the eUICC…)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DownloadProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS, #EP_FALSE_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-GetEIS request	The EID parameter is equal to #VIRTUAL_EID_RPS	PROC_REQ1, PM_REQ11, PM_REQ14
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-GetEIS, #EIS_ES3_RPS)		
4	SM-DP-UT → SM-SR-S	Send the ES3-CreateISDP request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID parameter is equal to #ICCID1 3- The MNO-ID parameter is equal to #MNO1_S_ID 4- The REQUIRED-MEMORY parameter is present and lower than 750000 5- The MORE-TO-DO parameter may be present. If present, it shall be equal to #MORE_TODO_RPS or #NO_MORE_TODO_RPS	PROC_REQ1, PM_REQ11, PM_REQ16
5	SM-SR-S → SM-DP-UT	SEND_ERROR_RESP(ES3-CreateISDP, #FAILED, #SC_EUICC, #RC_MEMORY)		
6	SM-DP-UT → MNO1-S	Send the ES2-DownloadProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EUICC 3- The Reason code is equal to #RC_MEMORY	PM_REQ11

4.3.3.2.1.3 Test Sequence N°3 – Error Case: Conditional Parameters Missing

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DownloadProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #EP_FALSE_RPS)		
2	SM-DP-UT → MNO1-S	Send the ES2-DownloadProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_FUNCTION 3- The Reason code is equal to #RC_COND_PARAM	PM_REQ11

4.3.4 ES2 (MNO – SM-DP): UpdatePolicyRules

4.3.4.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

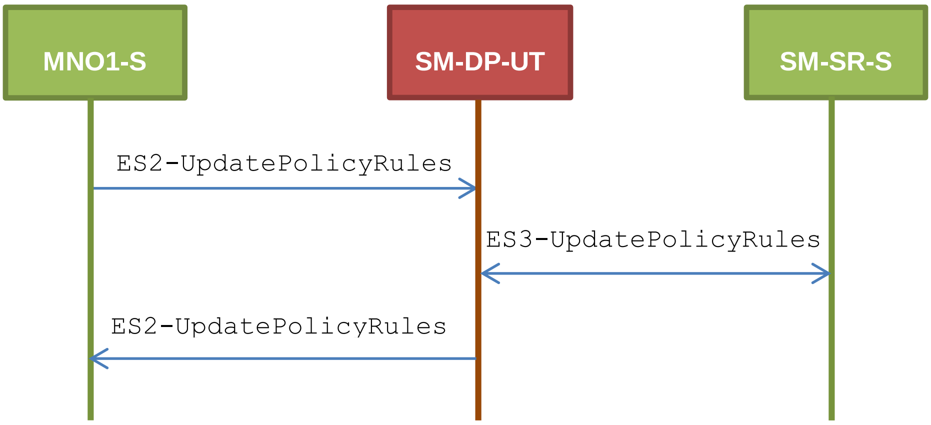
- PROC_REQ16
- PM_REQ12, PM_REQ19

4.3.4.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #SM_SR_S_ID and #SM_SR_S_ACCESSPOINT well known to the SM-DP-UT

Test Environment



4.3.4.2.1 TC.ES2.UPR.1: UpdatePolicyRules

Test Purpose

To ensure POL2 can be updated by the SM-DP through the SM-SR when a MNO requests it. An error case is also defined:

- *the Profile identified by the ICCID is unknown*

Referenced Requirements

- PROC_REQ16
- PM_REQ12, PM_REQ19

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_S_ID_RPS

4.3.4.2.1.1 Test Sequence N°1 – Nominal Case: No Rule

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-UpdatePolicyRules, #VIRTUAL_EID_RPS, #ICCID1_RPS, {SM_SR_ID_RPS}, #POL2_EMPTY_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-UpdatePolicyRules request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- Check that POL2 parameter is equal to #POL2_EMPTY_RPS	PM_REQ12, PM_REQ19, PROC_REQ16
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-UpdatePolicyRules)		
4	SM-DP-UT → MNO1-S	Send the ES2-UpdatePolicyRules response	The Status is equal to #SUCCESS	PM_REQ12, PROC_REQ16

4.3.4.2.1.2 Test Sequence N°2 – Nominal Case: Rule "Disabling not allowed"

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-UpdatePolicyRules,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS,`
		`{SM_SR_ID_RPS},`
		`#POL2_DIS_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-UpdatePolicyRules` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS` 3- The POL2 is equal to `#POL2_DIS_RPS`	PM_REQ12, PM_REQ19, PROC_REQ16

3	SM-SR-S → SM-DP-UT	`SEND_SUCCESS_RESP(`
		`ES3-UpdatePolicyRules)`		

4	SM-DP-UT → MNO1-S	Send the `ES2-UpdatePolicyRules` response	The `Status` is equal to `#SUCCESS`	PM_REQ12, PROC_REQ16

4.3.4.2.1.3 Test Sequence N°3 – Error Case: Unknown Profile ICCID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-UpdatePolicyRules,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS,`
		`{SM_SR_ID_RPS},`
		`#POL2_DEL_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-UpdatePolicyRules` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS` 3- The POL2 is equal to `#POL2_DEL_RPS`	PM_REQ12, PM_REQ19, PROC_REQ16

Step	Direction	Sequence / Description	Expected result	REQ

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-UpdatePolicyRules,`
		`#FAILED,`
		`#SC_PROFILE_ICCID,`
		`#RC_UNKNOWN)`		

4	SM-DP-UT → MNO1-S	Send the `ES2-UpdatePolicyRules` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_PROFILE_ICCID` 3- The `Reason code` is equal to `#RC_UNKNOWN`	PM_REQ12, PROC_REQ16

4.3.5 ES2 (MNO – SM-DP): UpdateSubscriptionAddress

4.3.5.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

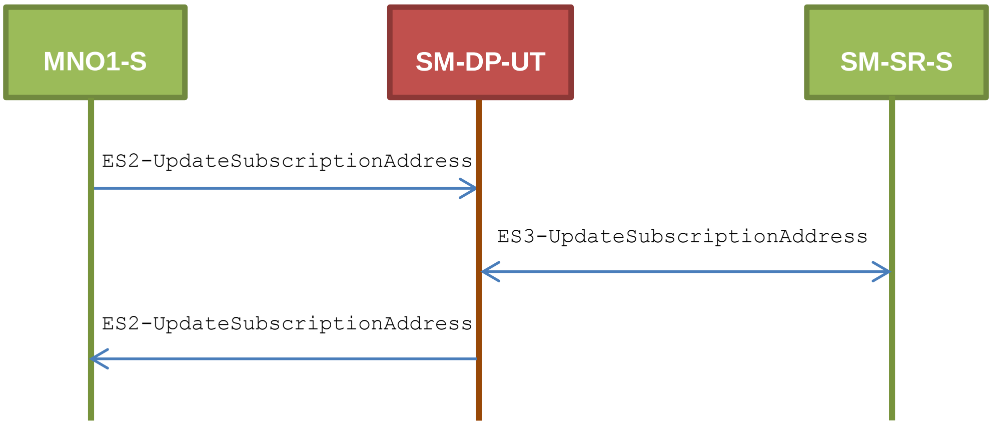
- PM_REQ13, PM_REQ20

4.3.5.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #SM_SR_S_ID and #SM_SR_S_ACCESSPOINT well known to the SM-DP-UT

Test Environment


MNO1-S
SM-DP-UT
SM-SR-S
ES2-UpdateSubscriptionAddress
ES3-UpdateSubscriptionAddress
ES2-UpdateSubscriptionAddress

4.3.5.2.1 TC.ES2.USA.1: UpdateSubscriptionAddress

Test Purpose

To ensure Subscription Address can be updated by the SM-DP through the SM-SR when a MNO requests it.

Referenced Requirements

- PM_REQ13, PM_REQ20

Initial Conditions

- The variable `{SM_SR_ID_RPS}` shall be set to `#SM_SR_S_ID_RPS`

4.3.5.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-UpdateSubscriptionAddress, #VIRTUAL_EID_RPS, #ICCID1_RPS, #NEW_ADDR_RPS, {SM_SR_ID_RPS})		
2	SM-DP-UT → SM-SR-S	Send the ES3-UpdateSubscriptionAddress request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- The Subscription Address is equal to #NEW_ADDR_RPS	PM_REQ13, PM_REQ20
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-UpdateSubscriptionAddress)		
4	SM-DP-UT → MNO1-S	Send the ES2-UpdateSubscriptionAddress response	The Status is equal to #SUCCESS	PM_REQ13

4.3.6 ES2 (MNO – SM-DP): EnableProfile

4.3.6.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PROC_REQ7
- PF_REQ12, PF_REQ15, PF_REQ17, PF_REQ18, PF_REQ21, PF_REQ23

4.3.6.2 Test Cases

General Initial Conditions

- #MNO1_S_ID, #MNO1_S_ACCESSPOINT, #MNO2_S_ID and #MNO2_S_ACCESSPOINT well known to the SM-DP-UT
- #SM_SR_S_ID and #SM_SR_S_ACCESSPOINT well known to the SM-DP-UT

4.3.6.2.1 TC.ES2.EP.1: EnableProfile

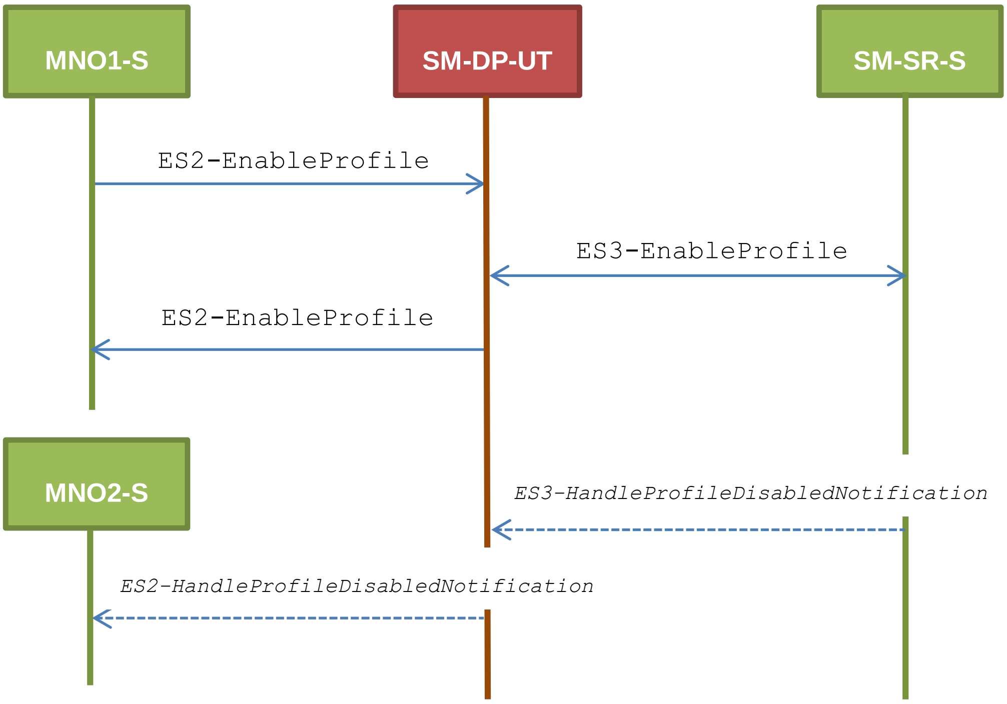
Test Purpose

To ensure a Profile can be Enabled by the SM-DP through the SM-SR when a MNO requests it. After enabling the Profile, the SM-SR sends the notification HandleProfileDisabledNotification to the SM-DP: this notification shall be forwarded to the corresponding MNO.

Some error cases are also defined:

- *the Profile identified by the ICCID is known to the SM-SR but installed on another eUICC than the one identified by the SM-DP*
- *the SM-DP is not allowed to perform this function on the target Profile*

Test Environment



Referenced Requirements

- PROC_REQ7
- PF_REQ12, PF_REQ15, PF_REQ18, PF_REQ21

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_S_ID_RPS

4.3.6.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-EnableProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-EnableProfile request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS	PROC_REQ7, PF_REQ12, PF_REQ18
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-EnableProfile)		
4	SM-DP-UT → MNO1-S	Send the ES2-EnableProfile response	The Status is equal to #SUCCESS	PROC_REQ7, PF_REQ12
5	SM-SR-S → SM-DP-UT	SEND_NOTIF(ES3-HandleProfile DisabledNotification, #VIRTUAL_EID_RPS, #ICCID2_RPS #MNO2_ID_RPS, #TIMESTAMP_RPS)		
6	SM-DP-UT → MNO2-S	Send the ES2-HandleProfile DisabledNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is equal to #TIMESTAMP_RPS	PROC_REQ7, PF_REQ15, PF_REQ21

4.3.6.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-EnableProfile,`
		`#VIRTUAL_EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#ICCID1_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-EnableProfile` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS`	PROC_REQ7, PF_REQ12, PF_REQ18

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-EnableProfile,`
		`#FAILED,`
		`#SC_PROFILE_ICCID,`
		`#RC_INVALID_DEST)`		

4	SM-DP-UT → MNO1-S	Send the `ES2-EnableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_PROFILE_ICCID` 3- The `Reason code` is equal to `#RC_INVALID_DEST`	PROC_REQ7, PF_REQ12

4.3.6.2.1.3 Test Sequence N°3 – Error Case: Not Allowed

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-EnableProfile,`
		`#VIRTUAL_EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#ICCID1_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-EnableProfile` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS`	PROC_REQ7, PF_REQ12, PF_REQ18

Step	Direction	Sequence / Description	Expected result	REQ

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-EnableProfile,`
		`#FAILED,`
		`#SC_PROFILE_ICCID,`
		`#RC_NOT_ALLOWED)`		

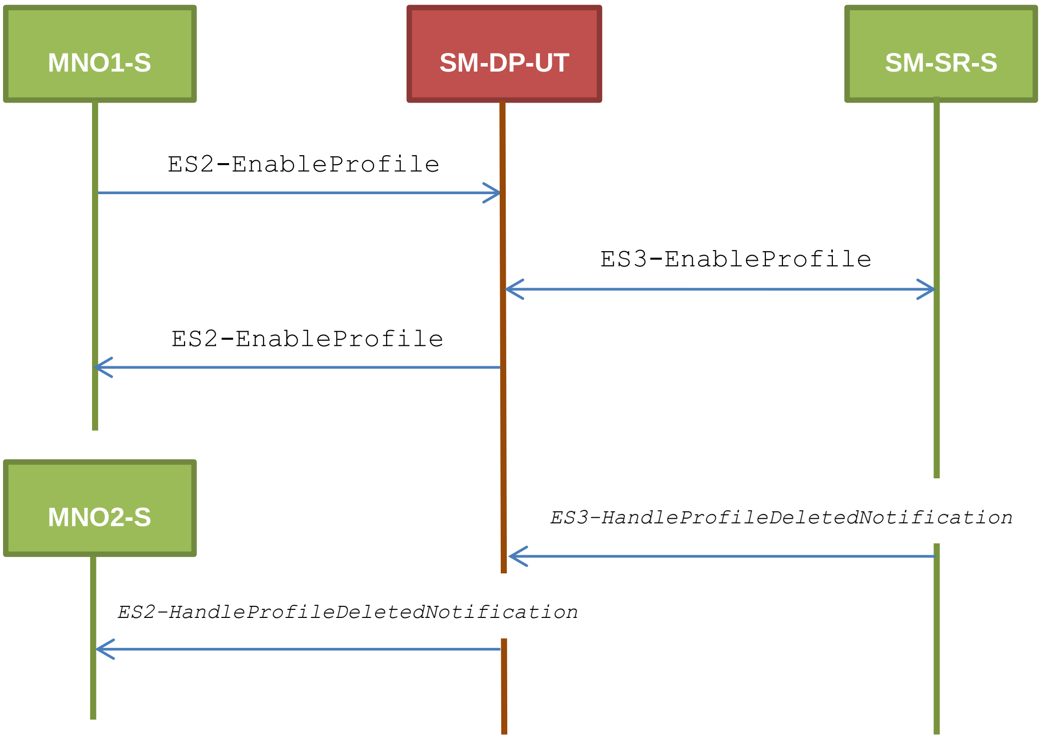
4	SM-DP-UT → MNO1-S	Send the `ES2-EnableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_PROFILE_ICCID` 3- The `Reason code` is equal to `#RC_NOT_ALLOWED`	PROC_REQ7, PF_REQ12

4.3.6.2.2 TC.ES2.EP.2: EnableProfileWithDeletion

Test Purpose

To ensure MNO can ask the SM-DP to enable a Profile. The notification HandleProfileDeletedNotification is tested considering that the deletion has been triggered by the evaluation of POL1 on SM-SR side.

Test Environment



Referenced Requirements

- PROC_REQ7
- PF_REQ12, PF_REQ17, PF_REQ18, PF_REQ23

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_S_ID_RPS

4.3.6.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-EnableProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-EnableProfile request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS	PROC_REQ7, PF_REQ12, PF_REQ18
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-EnableProfile)		
4	SM-DP-UT → MNO1-S	Send the ES2-EnableProfile response	The Status is equal to #SUCCESS	PROC_REQ7, PF_REQ12
5	SM-SR-S → SM-DP-UT	SEND_NOTIF(ES3-HandleProfileDeletedNotification, #VIRTUAL_EID_RPS, #ICCID2_RPS #MNO2_ID_RPS, #TIMESTAMP_RPS)		
6	SM-DP-UT → MNO2-S	Send the ES2-HandleProfileDeletedNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is equal to #TIMESTAMP_RPS	PROC_REQ7, PF_REQ17, PF_REQ23

4.3.7 ES2 (MNO – SM-DP): DisableProfile

4.3.7.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

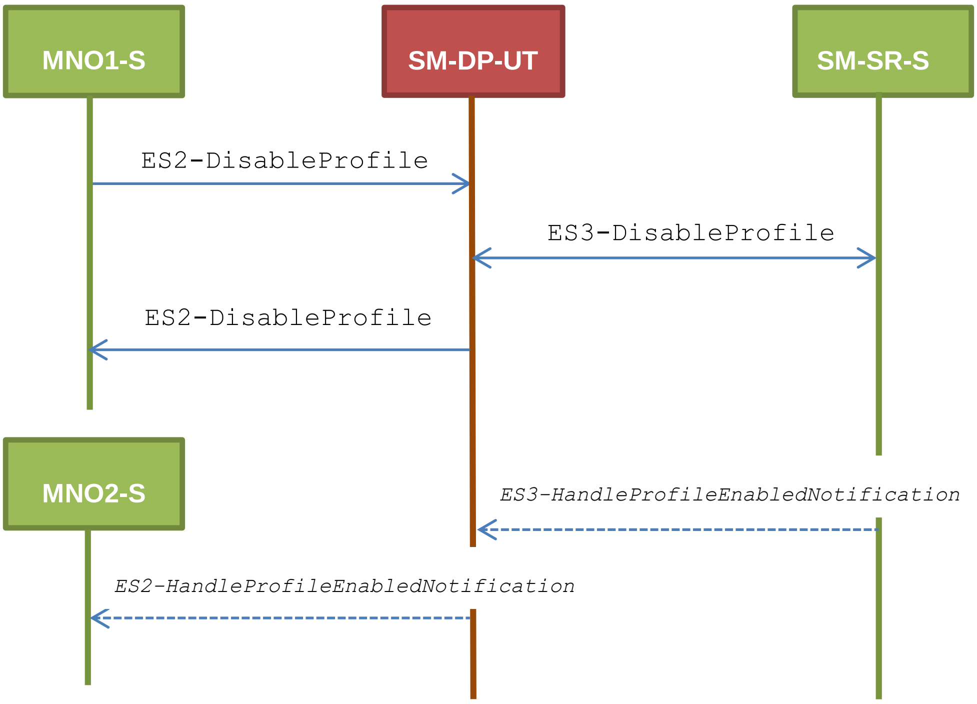
- PROC_REQ10
- PF_REQ13, PF_REQ16, PF_REQ19, PF_REQ22

4.3.7.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID`, `#MNO1_S_ACCESSPOINT`, `#MNO2_S_ID` and `#MNO2_S_ACCESSPOINT` well known to the SM-DP-UT
- `#SM_SR_S_ID` and `#SM_SR_S_ACCESSPOINT` well known to the SM-DP-UT

Test Environment



4.3.7.2.1 TC.ES2.DISP.1: DisableProfile

Test Purpose

To ensure Profile can be Disabled by the SM-DP through the SM-SR when a MNO requests it. After disabling the Profile, the SM-SR sends the notification HandleProfileEnabledNotification which shall be forwarded to the corresponding MNO. Some error cases are also defined:

- *error during execution of the enabling command on the eUICC*

- *the POL1 of the impacted Profiles does not allow this operation*

Referenced Requirements

- PROC_REQ10
- PF_REQ13, PF_REQ16, PF_REQ19, PF_REQ22

Initial Conditions

- The variable `{SM_SR_ID_RPS}` shall be set to `#SM_SR_S_ID_RPS`

4.3.7.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DisableProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-DisableProfile request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS	PROC_REQ10, PF_REQ13, PF_REQ19
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-DisableProfile)		
4	SM-DP-UT → MNO1-S	Send the ES2-DisableProfile response	The Status is equal to #SUCCESS	PROC_REQ10, PF_REQ13
5	SM-SR-S → SM-DP-UT	SEND_NOTIF(ES3-HandleProfile EnabledNotification, #VIRTUAL_EID_RPS, #ICCID2_RPS #MNO2_ID_RPS, #TIMESTAMP_RPS)		
6	SM-DP-UT → MNO2-S	Send the ES2-HandleProfile EnabledNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is equal to #TIMESTAMP_RPS	PROC_REQ10, PF_REQ16, PF_REQ22

4.3.7.2.1.2 Test Sequence N°2 – Error Case: Execution Error

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#ICCID1_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-DisableProfile` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS`	PROC_REQ10, PF_REQ13, PF_REQ19

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-DisableProfile,`
		`#FAILED,`
		`#SC_ISDR,`
		`#RC_EXECUTION_ERROR,`
		`#EUICC_RESP1_RPS)`		

4	SM-DP-UT → MNO1-S	Send the `ES2-DisableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_ISDR` 3- The `Reason code` is equal to `#RC_EXECUTION_ERROR`	PROC_REQ10, PF_REQ13

4.3.7.2.1.3 Test Sequence N°3 – Error Case: Incompatible POL1

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#ICCID1_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-DisableProfile` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS`	PROC_REQ10, PF_REQ13, PF_REQ19

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-S → SM-DP-UT	SEND_ERROR_RESP(ES3-DisableProfile, #FAILED, #SC_POL1, #RC_REFUSED)		
4	SM-DP-UT → MNO1-S	Send the ES2-DisableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_POL1 3- The Reason code is equal to #RC_REFUSED	PROC_REQ10, PF_REQ13

4.3.8 ES2 (MNO – SM-DP): DeleteProfile

4.3.8.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

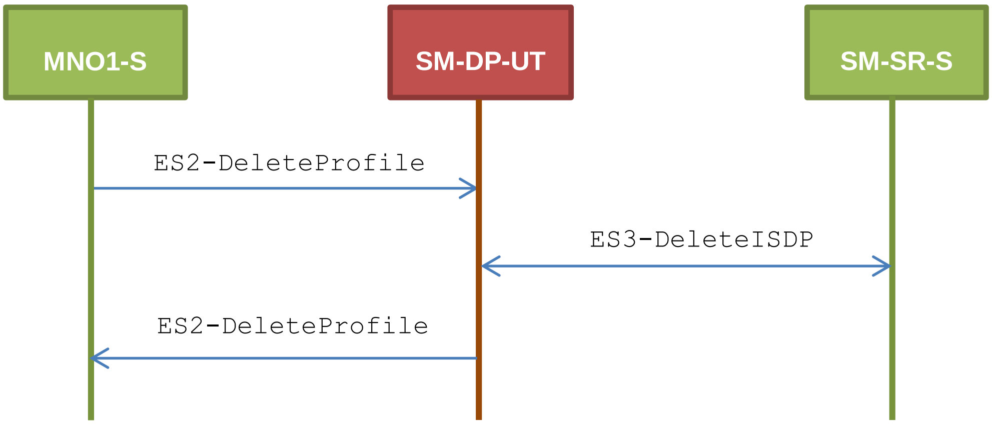
- PROC_REQ12
- PF_REQ14, PF_REQ20

4.3.8.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #SM_SR_S_ID and #SM_SR_S_ACCESSPOINT well known to the SM-DP-UT

Test Environment



4.3.8.2.1 TC.ES2.DP.1: DeleteProfile

Test Purpose

To ensure Profile can be deleted by the SM-DP through the SM-SR when a MNO requests it. Some error cases are also defined:

- *the POL2 of the impacted Profiles does not allow this operation*
- *the target Profile cannot be Disabled (in case of the disabling of the Profile shall be performed before the deletion)*

Referenced Requirements

- PROC_REQ12
- PF_REQ14, PF_REQ20

Initial Conditions

- The variable `{SM_SR_ID_RPS}` shall be set to `#SM_SR_S_ID_RPS`

4.3.8.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DeleteProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-DeleteISDP request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS	PROC_REQ12, PF_REQ14, PF_REQ20
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-DeleteISDP)		
4	SM-DP-UT → MNO1-S	Send the ES2-DeleteProfile response	The Status is equal to #SUCCESS	PROC_REQ12, PF_REQ14

4.3.8.2.1.2 Test Sequence N°2 – Error Case: Incompatible POL2

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DeleteProfile,`
		`#VIRTUAL_EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#ICCID1_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-DeleteISDP` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS`	PROC_REQ12, PF_REQ14, PF_REQ20

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-DeleteISDP,`
		`#FAILED,`
		`#SC_POL2,`
		`#RC_REFUSED)`		

4	SM-DP-UT → MNO1-S	Send the `ES2-DeleteProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_POL2` 3- The `Reason code` is equal to `#RC_REFUSED`	PROC_REQ12, PF_REQ14

4.3.8.2.1.3 Test Sequence N°3 – Error Case: Automatic Disabling Not Allowed

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DeleteProfile,`
		`#VIRTUAL_EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#ICCID1_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-DeleteISDP` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The ICCID is equal to `#ICCID1_RPS`	PROC_REQ12, PF_REQ14, PF_REQ20

Step	Direction	Sequence / Description	Expected result	REQ

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-DeleteISDP,`
		`#FAILED,`
		`#SC_EUICC,`
		`#RC_REFUSED)`		

4	SM-DP-UT → MNO1-S	Send the `ES2-DeleteProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_EUICC` 3- The `Reason code` is equal to `#RC_REFUSED`	PROC_REQ12, PF_REQ14

4.3.9 ES3 (SM-DP – SM-SR): GetEIS

4.3.9.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

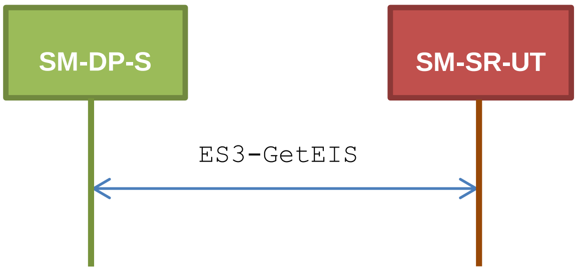
- PM_REQ14

4.3.9.2 Test Cases

General Initial Conditions

- `#SM_DP_S_ID` and `#SM_DP_S_ACCESSPOINT` well known to the SM-SR-UT
- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT

Test Environment



4.3.9.2.1 TC.ES3.GEIS.1: GetEIS

Test Purpose

To ensure EIS can be retrieved by the SM-SR when a SM-DP requests it. An error case is also defined:

- *the EID is unknown to the SM-SR*

Referenced Requirements

- PM_REQ14

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS

4.3.9.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS
  - o {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #VIRTUAL_EID_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3- GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES3_RPS	PM_REQ14

4.3.9.2.1.2 Test Sequence N°2 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #VIRTUAL_EID_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3- GetEIS response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EID 3- The Reason code is equal to #RC_UNKNOWN	PM_REQ14

4.3.10 ES3 (SM-DP – SM-SR): AuditEIS

4.3.10.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

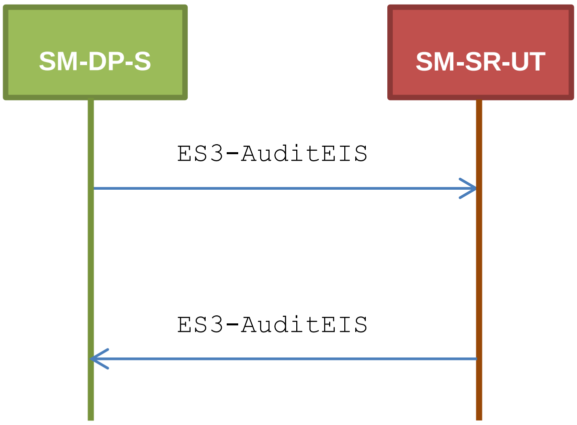
- PM_REQ15

4.3.10.2 Test Cases

General Initial Conditions

- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT

Test Environment



4.3.10.2.1 TC.ES3.AEIS.1: AuditEIS

Test Purpose

To ensure the EIS audit can be performed by the SM-SR if the EID is known to the SM-SR.

Referenced Requirements

- PM_REQ15

Initial Conditions

- None

4.3.10.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-AuditEIS, #VIRTUAL_EID_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3- AuditEIS response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EID 3- The Reason code is equal to #RC_UNKNOWN	PM_REQ15

4.3.11 ES3 (SM-DP – SM-SR): CreateISDP

4.3.11.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

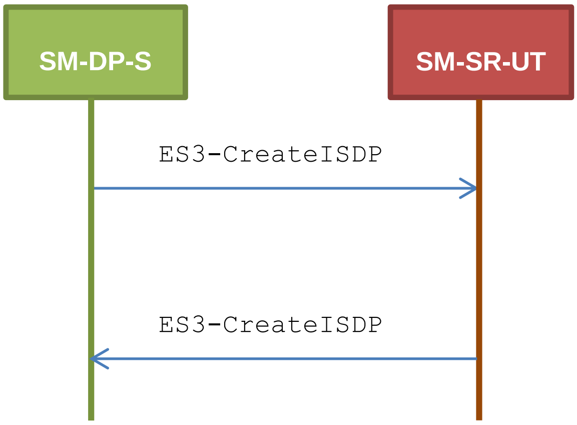
- PM_REQ16

4.3.11.2 Test Cases

General Initial Conditions

- `#SM_DP_S_ID` and `#SM_DP_S_ACCESSPOINT` well known to the SM-SR-UT
- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT

Test Environment



4.3.11.2.1 TC.ES3.CISDP.1: CreateISDP

Test Purpose

To ensure the ISDP creation is well implemented on SM-SR. Only error cases are defined:

- *the eUICC has not enough free memory to execute the creation of the new ISD-P with the required amount of memory*
- *the ICCID is already allocated to another Profile*

Referenced Requirements

- PM_REQ16

Initial Conditions

- None

4.3.11.2.1.1 Test Sequence N°1 – Error Case: Not Enough Memory

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS2_ES1_RPS (i.e. the Profile identified by #ICCID1 is not present)
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-CreateISDP, #VIRTUAL_EID_RPS, #ICCID1_RPS, #MNO1_ID_RPS, #BIG_MEM_RPS, #MORE_TODO_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-CreateISDP response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EUICC 3- The Reason code is equal to #RC_MEMORY	PM_REQ16

4.3.11.2.1.2 Test Sequence N°2 – Error Case: Already In Use

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is in Enabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-CreateISDP, #VIRTUAL_EID_RPS, #ICCID1_RPS, #MNO1_ID_RPS, #SMALL_MEM_RPS, #NO_MORE_TODO_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-CreateISDP response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_ALREADY_USED	PM_REQ16

4.3.12 ES3 (SM-DP – SM-SR): SendData

4.3.12.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

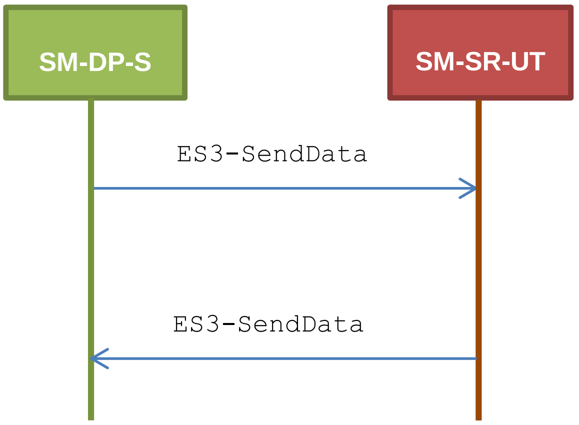
- PM_REQ17

4.3.12.2 Test Cases

General Initial Conditions

- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT
- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

Test Environment



4.3.12.2.1 TC.ES3.SDATA.1: SendData

Test Purpose

To ensure the SendData method can be used by the SM-DP except if:

- *the ISD-P is unknown to the SM-SR or*
- *the ISD-P is known to the SM-SR but installed on another eUICC than the one identified by the SM-DP*

Referenced Requirements

- PM_REQ17

Initial Conditions

- None

4.3.12.2.1.1 Test Sequence N°1 – Error Case: Unknown ISD-P

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS2_ES1_RPS (i.e. the ISD-P identified by #ISDP2_RPS is not present)

  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SendData, #VIRTUAL_EID_RPS, #SD_ISDP2_RPS, #DATA_RPS, #MORE_TODO_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-SendData response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_SD_AID 3- The Reason code is equal to #RC_UNKNOWN	PM_REQ17

4.3.12.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS2_ES1_RPS (i.e. the ISD-P identified by #ISDP3_RPS is only present)
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS
- The eUICC identified by the #VIRTUAL_EID2 is provisioned on the SM-SR-UT with the #EIS3_ES1_RPS (i.e. the ISD-P identified by #ISDP2_RPS is only present)
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SendData, #VIRTUAL_EID_RPS, #SD_ISDP2_RPS, #DATA_RPS, #MORE_TODO_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-SendData response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_SD_AID 3- The Reason code is equal to #RC_INVALID_DEST	PM_REQ17

4.3.13 ES3 (SM-DP – SM-SR): UpdatePolicyRules

4.3.13.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

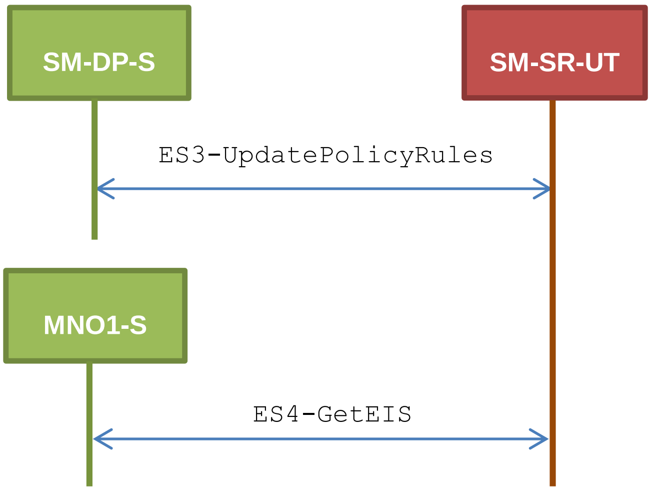
- PROC_REQ16
- PM_REQ19, PM_REQ22

4.3.13.2 Test Cases

General Initial Conditions

- `#SM_DP_S_ID` and `#SM_DP_S_ACCESSPOINT` well known to the SM-SR-UT
- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT
- `#MNO1_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.13.2.1 TC.ES3.UPR.1: UpdatePolicyRules

Test Purpose

To ensure the SM-SR can update the Policy Rules (POL2) according the parameters sent by the SM-DP. To make sure that the POL2 have been set on SM-SR side, the EIS is retrieved just after updating the rules.

Referenced Requirements

- PROC_REQ16
- PM_REQ19, PM_REQ22

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_S_ID_RPS

4.3.13.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS (i.e. the Profile identified by #ICCID1 is present)
  - o {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-UpdatePolicyRules, #VIRTUAL_EID_RPS, #ICCID1_RPS, #POL2_DIS_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-UpdatePolicyRules response	The Status is equal to #SUCCESS	PM_REQ19, PROC_REQ16
3	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #VIRTUAL_EID_RPS)		
4	SM-SR-UT → MNO1-S	Send the ES4- GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES4_RPS except that POL2 of #ICCID1 is equal to #POL2_DIS_RPS	PM_REQ19, PM_REQ22, PROC_REQ16

4.3.14 ES3 (SM-DP – SM-SR): UpdateSubscriptionAddress

4.3.14.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PM_REQ20, PM_REQ22

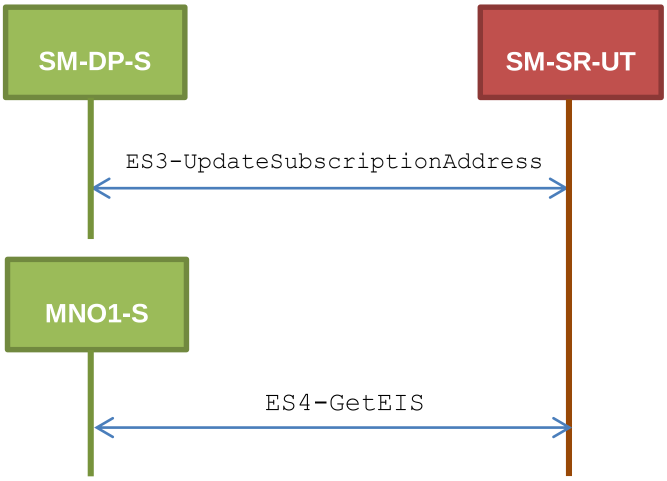
4.3.14.2 Test Cases

General Initial Conditions

- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT
- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO1_S_ACCESSPOINT well known to the SM-SR-UT

  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.14.2.1 TC.ES3.USA.1: UpdateSubscriptionAddress

Test Purpose

To ensure Subscription Address can be updated by the SM-SR when a SM-DP requests it. To make sure that the Subscription Address has been set on SM-SR side, the EIS is retrieved just after updating the address.

Referenced Requirements

- PM_REQ20, PM_REQ22

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS

4.3.14.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS (i.e. the Profile identified by #ICCID1 is present)
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-UpdateSubscriptionAddress, #VIRTUAL_EID_RPS, #ICCID1_RPS, #NEW_ADDR_RPS)		

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-SR-UT → SM-DP-S	Send the ES3-UpdateSubscriptionAddress response	The Status is equal to #SUCCESS	PM_REQ20
3	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #VIRTUAL_EID_RPS)		
4	SM-SR-UT → MNO1-S	Send the ES4- GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES4_RPS except that the Subscription Address of #ICCID1 is equal to #SUB_ADDR3_RPS	PM_REQ20, PM_REQ22

4.3.15 ES3 (SM-DP – SM-SR): UpdateConnectivityParameters

4.3.15.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

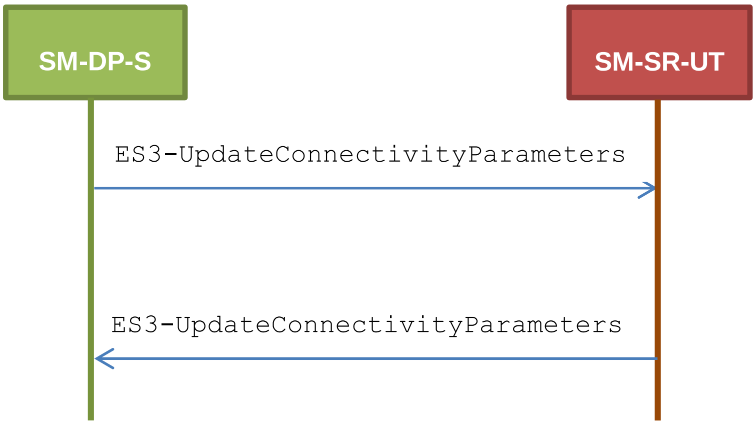
- PM_REQ21

4.3.15.2 Test Cases

General Initial Conditions

- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT
- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

Test Environment


SM-DP-S
SM-SR-UT
ES3-UpdateConnectivityParameters
ES3-UpdateConnectivityParameters

4.3.15.2.1 TC.ES3.UCP.1: UpdateConnectivityParameters

Test Purpose

To ensure the UpdateConnectivityParameters method can be performed by the SM-SR except if:

- *the EID is unknown to the SM-SR or*
- *the Profile identified by the ICCID is unknown*

Referenced Requirements

- PM_REQ21

Initial Conditions

- None

4.3.15.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-UpdateConnectivityParameters, #VIRTUAL_EID_RPS, #ICCID1_RPS, #CON_PARAM_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-UpdateConnectivityParameters response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EID 3- The Reason code is equal to #RC_UNKNOWN	PM_REQ21

4.3.15.2.1.2 Test Sequence N°2 – Error Case: Unknown Profile ICCID

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS2_ES1_RPS (i.e. the Profile identified by #ICCID1 is not present)
    - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-UpdateConnectivityParameters, #VIRTUAL_EID_RPS, #ICCID1_RPS, #CON_PARAM_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-UpdateConnectivityParameters response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_UNKNOWN	PM_REQ21

4.3.16 ES3 (SM-DP – SM-SR): EnableProfile

4.3.16.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

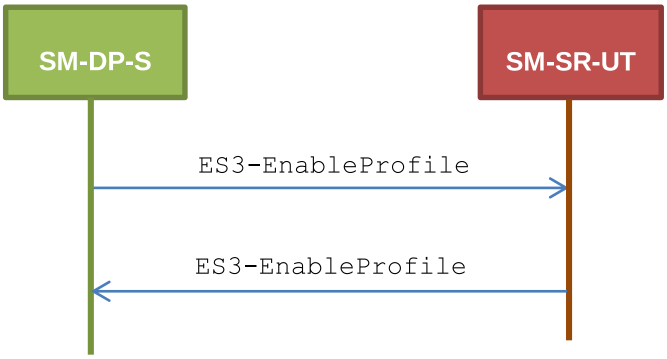
- PF_REQ18

4.3.16.2 Test Cases

General Initial Conditions

- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT
- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

Test Environment



4.3.16.2.1 TC.ES3.EP.1: EnableProfile

Test Purpose

To ensure a Profile can be Enabled by the SM-SR, when an SM-DP requests it, only if:

- *the SM-SR is responsible for the management of the targeted eUICC*
- *the Profile identified by its ICCID is loaded on the targeted eUICC*
- *the Profile identified by its ICCID is in Disabled state*
- *the POL2 of the target Profile and the POL2 of the currently Enabled Profile allows the enabling*

Referenced Requirements

- PF_REQ18

Initial Conditions

- None

4.3.16.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EID 3- The Reason code is equal to #RC_UNKNOWN	PF_REQ18

4.3.16.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is not installed on the eUICC identified by #VIRTUAL_EID

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_INVALID_DEST	PF_REQ18

4.3.16.2.1.3 Test Sequence N°3 – Error Case: Already Enabled Profile

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is in Enabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_NOT_ALLOWED	PF_REQ18

4.3.16.2.1.4 Test Sequence N°4 – Error Case: Incompatible Enabled Profile POL2

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID2 is installed on the eUICC identified by #VIRTUAL_EID and is in Enabled state

- The POL2 of the Profile identified by the #ICCID2 is "Disabling of this Profile not allowed"
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_POL2 3- The Reason code is equal to #RC_REFUSED	PF_REQ18

4.3.17 ES3 (SM-DP – SM-SR): DisableProfile

4.3.17.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

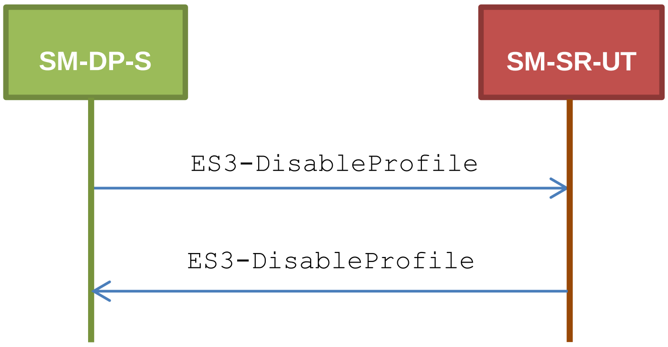
- PF_REQ19

4.3.17.2 Test Cases

General Initial Conditions

- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT
- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

Test Environment



4.3.17.2.1 TC.ES3.DISP.1: DisableProfile

Test Purpose

To ensure a Profile can be Disabled by the SM-SR, when an SM-DP requests it, only if:

- *the SM-SR is responsible for the management of the targeted eUICC*
- *the Profile identified by its ICCID is loaded on the targeted eUICC*
- *the Profile identified by its ICCID is in Enabled state*
- *the POL2 of the target Profile allows the disabling*

Referenced Requirements

- PF_REQ19

Initial Conditions

- None

4.3.17.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DisableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DisableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EID 3- The Reason code is equal to #RC_UNKNOWN	PF_REQ19

4.3.17.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is not installed on the eUICC identified by #VIRTUAL_EID

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DisableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DisableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_INVALID_DEST	PF_REQ19

4.3.17.2.1.3 Test Sequence N°3 – Error Case: Already Disabled Profile

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DisableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DisableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_NOT_ALLOWED	PF_REQ19

4.3.17.2.1.4 Test Sequence N°4 – Error Case: Incompatible POL2

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The POL2 of the Profile identified by the #ICCID1 is “Disabling of this Profile not allowed”
- The Profile identified by the #ICCID1 is in Enabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DisableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DisableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_POL2 3- The Reason code is equal to #RC_REFUSED	PF_REQ19

4.3.18 ES3 (SM-DP – SM-SR): DeleteISDP

4.3.18.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

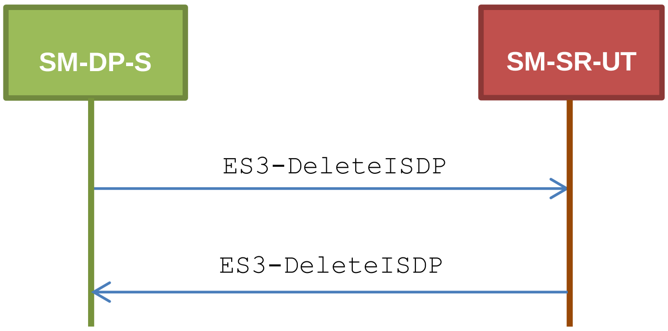
- PF_REQ20

4.3.18.2 Test Cases

General Initial Conditions

- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT
- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

Test Environment



4.3.18.2.1 TC.ES3.DISDP.1: DeleteISDP

Test Purpose

To ensure a Profile can be deleted by the SM-SR, when an SM-DP requests it, only if:

- *the SM-SR is responsible for the management of the targeted eUICC*

- *the Profile identified by its ICCID is loaded on the targeted eUICC*
- *the SM-DP is authorized to delete the target Profile by the MNO owning the target Profile*
- *the POL2 of the target Profile allows the deletion*
- *the target Profile is not the Profile having the Fall-back Attribute*

Referenced Requirements

- PF_REQ20

Initial Conditions

- None

4.3.18.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DeleteISDP, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DeleteISDP response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EID 3- The Reason code is equal to #RC_UNKNOWN	PF_REQ20

4.3.18.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is not installed on the eUICC identified by #VIRTUAL_EID

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DeleteISDP, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DeleteISDP response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_INVALID_DEST	PF_REQ20

4.3.18.2.1.3 Test Sequence N°3 – Error Case: Incompatible POL2

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The POL2 of the Profile identified by the #ICCID1 is "Deletion of this Profile not allowed"
- The Profile identified by the #ICCID1 is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DeleteISDP, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DeleteISDP response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_POL2 3- The Reason code is equal to #RC_REFUSED	PF_REQ20

4.3.18.2.1.4 Test Sequence N°5 – Error Case: Fall-back Profile

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID
- The Profile identified by the #ICCID1 has the Fall-back Attribute
- The Profile identified by the #ICCID1 is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DeleteISDP, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-DeleteISDP response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_REFUSED	PF_REQ20

4.3.19 ES4 (MNO – SM-SR): GetEIS

4.3.19.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

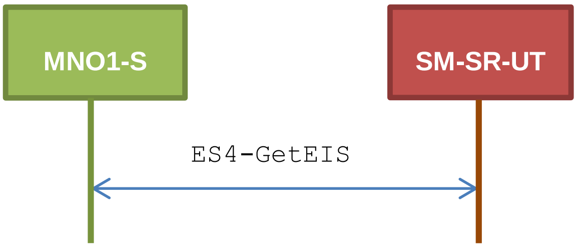
- PM_REQ22

4.3.19.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO1_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.19.2.1 TC.ES4.GEIS.1: GetEIS

Test Purpose

To ensure EIS can be retrieved by the SM-SR when a MNO requests it.

Referenced Requirements

- PM_REQ22

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS

4.3.19.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #VIRTUAL_EID_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4- GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES4_RPS	PM_REQ22

4.3.19.2.1.2 Test Sequence N°2 – Error Case: Not Allowed to Manage the EIS

This test case is defined as FFS pending further clarification in the GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2].

4.3.20 ES4 (MNO – SM-SR): UpdatePolicyRules

4.3.20.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

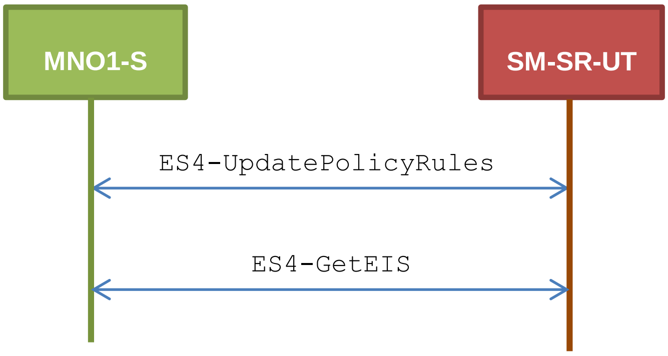
- PM_REQ22, PM_REQ23

4.3.20.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO1_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.20.2.1 TC.ES4.UPR.1: UpdatePolicyRules

Test Purpose

To ensure the SM-SR can update the Policy Rules (POL2) according the parameters sent by the MNO. To make sure that the POL2 have been set on SM-SR side, the EIS is retrieved just after updating the rules.

Referenced Requirements

- PM_REQ22, PM_REQ23

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS

4.3.20.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS (i.e. the Profile identified by #ICCID1 is present)
  - o {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-UpdatePolicyRules, #VIRTUAL_EID_RPS, #ICCID1_RPS, #POL2_DIS_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-UpdatePolicyRules response	The Status is equal to #SUCCESS	PM_REQ23
3	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #VIRTUAL_EID_RPS)		

Step	Direction	Sequence / Description	Expected result	REQ
4	SM-SR-UT → MNO1-S	Send the `ES4- GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` returned is equal to `#EIS_ES4_RPS` except that POL2 of `#ICCID1` is equal to `#POL2_DIS_RPS`	PM_REQ22, PM_REQ23

4.3.21 ES4 (MNO – SM-SR): UpdateSubscriptionAddress

4.3.21.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

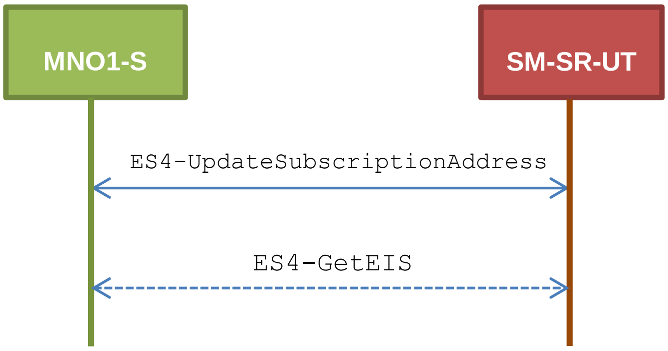
- PM_REQ22, PM_REQ24

4.3.21.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT
- `#MNO1_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.21.2.1 TC.ES4.USA.1: UpdateSubscriptionAddress

Test Purpose

To ensure Subscription Address can be updated by the SM-SR when a MNO requests it. To make sure that the Subscription Address has been set on SM-SR side, the EIS is retrieved just after updating the address. An error case is also defined:

- *the MNO is not allowed to manage the Subscription Address*

Referenced Requirements

- PM_REQ22, PM_REQ24

Initial Conditions

- The variable `{SM_SR_ID_RPS}` shall be set to `#SM_SR_UT_ID_RPS`

4.3.21.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- The eUICC identified by the `#VIRTUAL_EID` is provisioned on the SM-SR-UT with the `#EIS_ES1_RPS` (i.e. the Profile identified by `#ICCID1` is present)
    - o `{SM_SR_ID_RPS}` has been set to `#SM_SR_UT_ID_RPS`

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-UpdateSubscriptionAddress, #VIRTUAL_EID_RPS, #ICCID1_RPS, #NEW_ADDR_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-UpdateSubscriptionAddress response	The Status is equal to #SUCCESS	PM_REQ24
3	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #VIRTUAL_EID_RPS)		
4	SM-SR-UT → MNO1-S	Send the ES4- GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES4_RPS except that the Subscription Address of #ICCID1 is equal to #SUB_ADDR3_RPS	PM_REQ22, PM_REQ24

4.3.21.2.1.2 Test Sequence N°2 – Error Case: Not Allowed

Initial Conditions

- The eUICC identified by the `#VIRTUAL_EID` is provisioned on the SM-SR-UT (e.g. using `#EIS_ES1_RPS`)
- The Profile identified by the `#ICCID1` is installed on the eUICC identified by `#VIRTUAL_EID` and is not owned by MNO1-S (i.e. the `MNO-ID` is not equal to `#MNO1_S_ID`)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-UpdateSubscriptionAddress, #VIRTUAL_EID_RPS, #ICCID1_RPS, #NEW_ADDR_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-UpdateSubscriptionAddress response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_SUB_ADDR 3- The Reason code is equal to #RC_NOT_ALLOWED	PM_REQ24

4.3.22 ES4 (MNO – SM-SR): AuditEIS

4.3.22.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

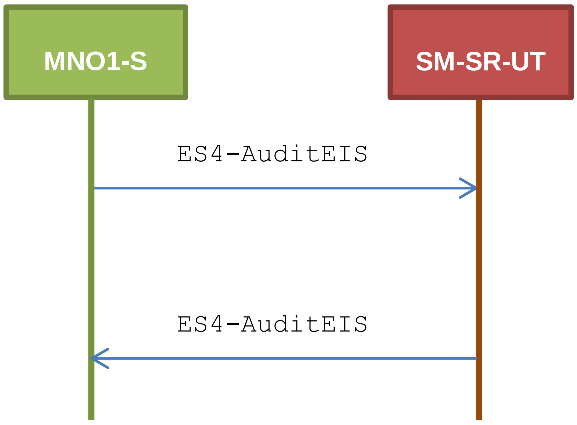
- PM_REQ25

4.3.22.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO1_S_ACCESSPOINT well known to the SM-SR-UT
    - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.22.2.1 TC.ES4.AEIS.1: AuditEIS

Test Purpose

To ensure the EIS audit can be performed by the SM-SR when MNO requests it, except if:

- *the Profile identified by the ICCID in the list does not belong to the MNO*

Referenced Requirements

- PM_REQ25

Initial Conditions

- None

4.3.22.2.1.1 Test Sequence N°1 – Error Case: Profile does not Belong to MNO

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is not owned by MNO1-S (i.e. the MNO-ID is not equal to #MNO1_S_ID)
- The Profile identified by the #ICCID1 is Enabled

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #VIRTUAL_EID_RPS, #ICCID1_RPS)		

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_PROFILE` 3- The `Reason code` is equal to `#RC_NOT_ALLOWED`	PM_REQ25

4.3.23 ES4 (MNO – SM-SR): EnableProfile

4.3.23.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

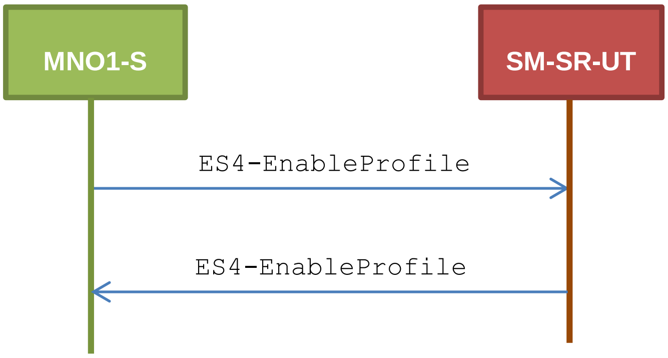
- PF_REQ24

4.3.23.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT
- `#MNO1_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.23.2.1 TC.ES4.EP.1: EnableProfile

Test Purpose

To ensure a Profile can be Enabled by the SM-SR, when an MNO requests it, only if:

- *the SM-SR is responsible for the management of the targeted eUICC*
- *the Profile identified by its ICCID is loaded on the targeted eUICC*
- *the Profile identified by its ICCID is in Disabled state*
- *the POL2 of the target Profile and the POL2 of the currently Enabled Profile allows the enabling*
- *the target Profile is owned by the requesting MNO*

Referenced Requirements

- PF_REQ24

Initial Conditions

- None

4.3.23.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EID 3- The Reason code is equal to #RC_UNKNOWN	PF_REQ24

4.3.23.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is not installed on the eUICC identified by #VIRTUAL_EID

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_INVALID_DEST	PF_REQ24

4.3.23.2.1.3 Test Sequence N°3 – Error Case: Already Enabled Profile

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)

- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is in Enabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_NOT_ALLOWED	PF_REQ24

4.3.23.2.1.4 Test Sequence N°4 – Error Case: Incompatible Enabled Profile POL2

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID2 is installed on the eUICC identified by #VIRTUAL_EID and is in Enabled state
- The POL2 of the Profile identified by the #ICCID2 is “Disabling of this Profile not allowed”
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-enableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_POL2 3- The Reason code is equal to #RC_REFUSED	PF_REQ24

4.3.23.2.1.5 Test Sequence N°5 – Error Case: Bad Profile Owner

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)

- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is not owned by MNO1-S (i.e. the MNO-ID is not equal to #MNO1_S_ID)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_NOT_ALLOWED	PF_REQ24

4.3.24 ES4 (MNO – SM-SR): DisableProfile

4.3.24.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

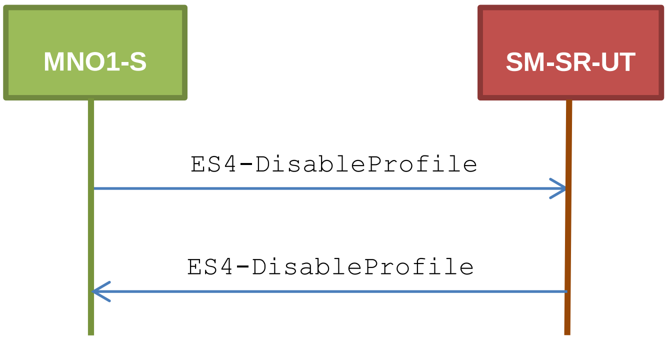
- PF_REQ25

4.3.24.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO1_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.24.2.1 TC.ES4.DISP.1: DisableProfile

Test Purpose

To ensure a Profile can be Disabled by the SM-SR, when an MNO requests it, only if:

- *the SM-SR is responsible for the management of the targeted eUICC*
- *the Profile identified by its ICCID is loaded on the targeted eUICC*
- *the Profile identified by its ICCID is in Enabled state*
- *the POL2 of the target Profile allows the disabling*
- *the target Profile is owned by the requesting MNO*

Referenced Requirements

- PF_REQ25

Initial Conditions

- None

4.3.24.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the `#VIRTUAL_EID` is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS)`		

2	SM-SR-UT → MNO1-S	Send the `ES4-DisableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_EID` 3- The `Reason code` is equal to `#RC_UNKNOWN`	PF_REQ25

4.3.24.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- The eUICC identified by the `#VIRTUAL_EID` is provisioned on the SM-SR-UT (e.g. using `#EIS_ES1_RPS`)
- The Profile identified by the `#ICCID1` is not installed on the eUICC identified by `#VIRTUAL_EID`

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-SR-UT → MNO1-S	Send the `ES4-DisableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_PROFILE_ICCID` 3- The `Reason code` is equal to `#RC_INVALID_DEST`	PF_REQ25

4.3.24.2.1.3 Test Sequence N°3 – Error Case: Already Disabled Profile

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS)`		

2	SM-SR-UT → MNO1-S	Send the `ES4-DisableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_PROFILE_ICCID` 3- The `Reason code` is equal to `#RC_NOT_ALLOWED`	PF_REQ25

4.3.24.2.1.4 Test Sequence N°4 – Error Case: Incompatible POL2

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The POL2 of the Profile identified by the #ICCID1 is “Disabling of this Profile not allowed”
- The Profile identified by the #ICCID1 is in Enabled state

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-SR-UT → MNO1-S	Send the `ES4-DisableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_POL2` 3- The `Reason code` is equal to `#RC_REFUSED`	PF_REQ25

4.3.24.2.1.5 Test Sequence N°6 – Error Case: Bad Profile Owner

Initial Conditions

- The eUICC identified by the `#VIRTUAL_EID` is provisioned on the SM-SR-UT (e.g. using `#EIS_ES1_RPS`)
- The Profile identified by the `#ICCID1` is installed on the eUICC identified by `#VIRTUAL_EID` and is not owned by MNO1-S (i.e. the `MNO-ID` is not equal to `#MNO1_S_ID`)

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS)`		

2	SM-SR-UT → MNO1-S	Send the `ES4-DisableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_PROFILE_ICCID` 3- The `Reason code` is equal to `#RC_NOT_ALLOWED`	PF_REQ25

4.3.25 ES4 (MNO – SM-SR): DeleteProfile

4.3.25.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

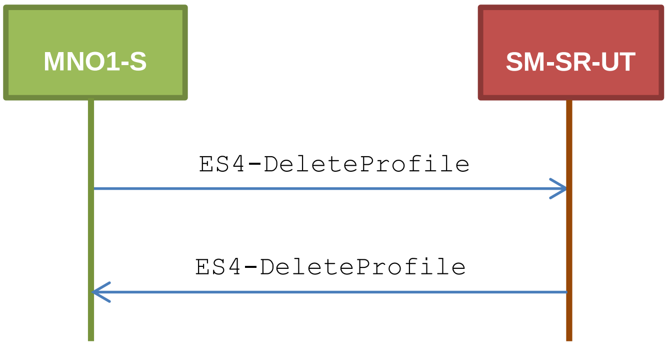
- PF_REQ26

4.3.25.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT
- `#MNO1_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT

Test Environment



4.3.25.2.1 TC.ES4.DP.1: DeleteProfile

Test Purpose

To ensure a Profile can be Disabled by the SM-SR, when an MNO requests it, only if:

- *the SM-SR is responsible for the management of the targeted eUICC*
- *the Profile identified by its ICCID is loaded on the targeted eUICC*
- *the POL2 of the target Profile allows the deletion*
- *the target Profile is not the Profile having the Fall-back Attribute*
- *the target Profile is owned by the requesting MNO*

Referenced Requirements

- PF_REQ26

Initial Conditions

- None

4.3.25.2.1.1 Test Sequence N°1 – Error Case: Unknown eUICC

Initial Conditions

- The eUICC identified by the `#VIRTUAL_EID` is not provisioned on the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DeleteProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS)`		

2	SM-SR-UT → MNO1-S	Send the `ES4-DeleteProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_EID` 3- The `Reason code` is equal to `#RC_UNKNOWN`	PF_REQ26

4.3.25.2.1.2 Test Sequence N°2 – Error Case: Invalid Destination

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is not installed on the eUICC identified by #VIRTUAL_EID

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-DeleteProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_INVALID_DEST	PF_REQ26

4.3.25.2.1.3 Test Sequence N°3 – Error Case: Incompatible POL2

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The POL2 of the Profile identified by the #ICCID1 is “Deletion of this Profile not allowed”
- The Profile identified by the #ICCID1 is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-DeleteProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_POL2 3- The Reason code is equal to #RC_REFUSED	PF_REQ26

4.3.25.2.1.4 Test Sequence N°4 – Error Case: Bad Profile Owner

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)

- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID and is not owned by MNO1-S (i.e. the MNO-ID is not equal to #MNO1_S_ID)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-DeleteProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_NOT_ALLOWED	PF_REQ26

4.3.25.2.1.5 Test Sequence N°5 – Error Case: Fall-back Profile

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT (e.g. using #EIS_ES1_RPS)
- The Profile identified by the #ICCID1 is installed on the eUICC identified by #VIRTUAL_EID
- The Profile identified by the #ICCID1 has the Fall-back Attribute
- The Profile identified by the #ICCID1 is in Disabled state

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-DeleteProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PROFILE_ICCID 3- The Reason code is equal to #RC_REFUSED	PF_REQ26

4.3.26 ES4 (MNO – SM-SR): PrepareSMSRChange

4.3.26.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

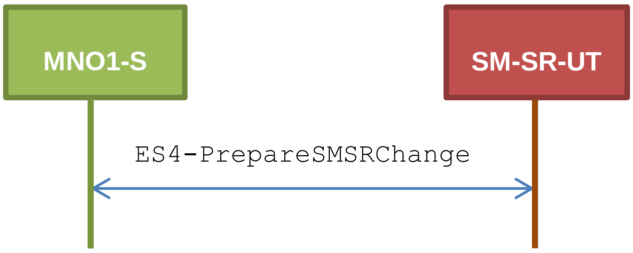
- EUICC_REQ35

4.3.26.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- The eUICC identified by the #VIRTUAL_EID is not provisioned on the SM-SR-UT

Test Environment



4.3.26.2.1 TC.ES4.PSMSRC.1: PrepareSMSRChange

Test Purpose

To ensure the method PrepareSMSRChange is well implemented on the SM-SR.
An error case is also defined:

- *the SM-SR is not capable of managing the eUICC identified by this EID*

Referenced Requirements

- EUICC_REQ35

Initial Conditions

- None

4.3.26.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-PrepareSMSRChange, #VIRTUAL_EID_RPS, #CUR_SR_S_ID_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-PrepareSMSRChange response	The Status is equal to #SUCCESS	EUICC_REQ35

4.3.26.2.1.2 Test Sequence N°2 – Error Case: SM-SR Not Capable of Managing the eUICC

Initial Conditions

- No setting has been initialized on SM-SR-UT to accept the SM-SR change

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-PrepareSMSRChange, #VIRTUAL_EID_RPS, #CUR_SR_S_ID_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-PrepareSMSRChange response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_FUN_PROV 3- The Reason code is equal to #RC_COND_USED	EUICC_REQ35

4.3.27 ES4 (MNO – SM-SR): SMSRchange

4.3.27.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

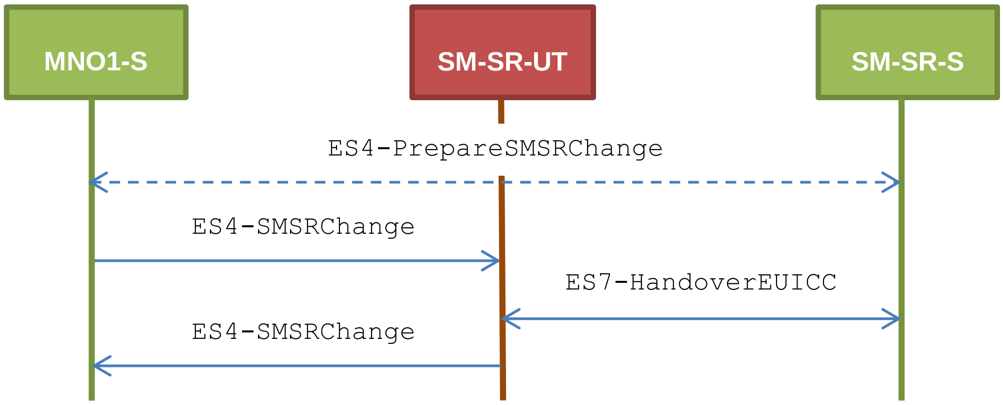
- EUICC_REQ36, EUICC_REQ39

4.3.27.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

Test Environment



Note that the function ES4-PrepareSMSRChange shall not be performed by the simulators (in the schema above, this is only an informative message).
In the following test cases, the Initiator Role (see GSMA Embedded SIM Remote Provisioning Architecture [1] section 2.3.1) is assumed to be played by the MNO1-S.

4.3.27.2.1 TC.ES4.SMSRC.1: SMSRChange

Test Purpose

To ensure the method SMSRChange can be performed by the SM-SR except if:

- *the ECASD certificate is expired or*
- *the new SM-SR is not capable of managing the eUICC identified by this EID or*
- *the preparation step has not been performed for the eUICC*

Referenced Requirements

- EUICC_REQ36, EUICC_REQ39

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS

4.3.27.2.1.1 Test Sequence N°1 – Error Case: Invalid ECASD

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-SMSRChange, #VIRTUAL_EID_RPS, #TGT_SR_S_ID_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC request	The EIS is equal to #EIS_ES1_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #FAILED, #SC_ECASD, #RC_EXPIRED)		
4	SM-SR-UT → MNO1-S	Send the ES4-SMSRChange response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_ECASD 3- The Reason code is equal to #RC_EXPIRED	EUICC_REQ36

4.3.27.2.1.2 Test Sequence N°2 – Error Case: Condition of Use Not Satisfied

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-SMSRChange, #VIRTUAL_EID_RPS, #TGT_SR_S_ID_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC request	The EIS is equal to #EIS_ES1_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #FAILED, #SC_FUN_PROV, #RC_COND_USED)		
4	SM-SR-UT → MNO1-S	Send the ES4-SMSRChange response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_FUN_PROV 3- The Reason code is equal to #RC_COND_USED	EUICC_REQ36

4.3.27.2.1.3 Test Sequence N°3 – Error Case: Preparation Step Not Performed

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-SMSRChange, #VIRTUAL_EID_RPS, #TGT_SR_S_ID_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC request	The EIS is equal to #EIS_ES1_RPS	EUICC_REQ36, EUICC_REQ39

Step	Direction	Sequence / Description	Expected result	REQ

3	SM-SR-S→ SM-SR-UT	`SEND_ERROR_RESP(`
		`ES7-HandoverEUICC,`
		`#FAILED,`
		`#SC_EID,`
		`#RC_ID_UNKNOWN)`		

4	SM-SR-UT → MNO1-S	Send the `ES4-SMSRChange` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_EID` 3- The `Reason code` is equal to `#RC_ID_UNKNOWN`	EUICC_REQ36

4.3.28 ES7 (SM-SR – SM-SR): HandoverEUICC

4.3.28.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

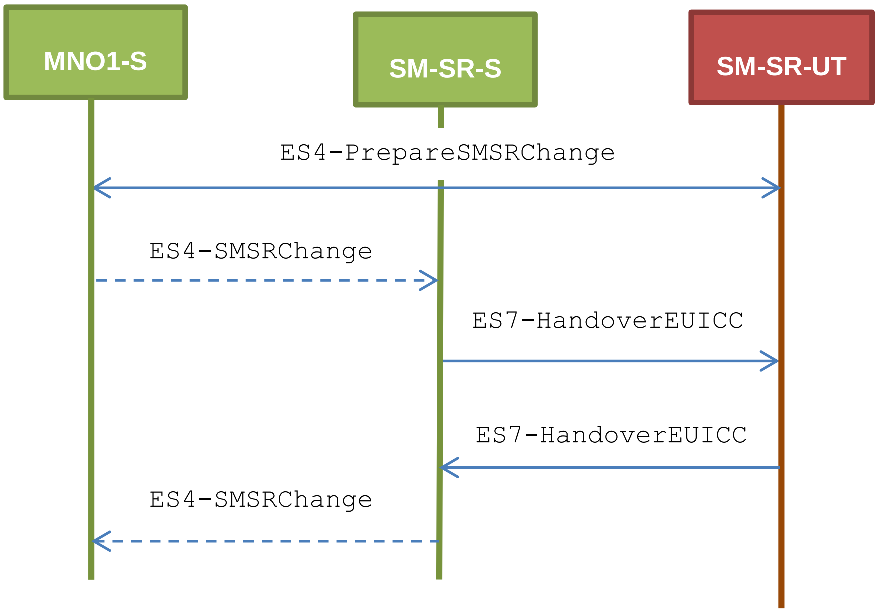
- EUICC_REQ35, EUICC_REQ39

4.3.28.2 Test Cases

General Initial Conditions

- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)
- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT
- `#EUM_S_PK_ECDSA` well known to the SM-SR-UT

Test Environment



Note that the function `ES4-SMSRChange` shall not be performed by the simulators (in the schema above, they are only informative messages).

4.3.28.2.1 TC.ES7.HEUICC.1: HandoverEUICC

Test Purpose

To ensure the method HandoverEUICC is well implemented on the SM-SR. Only error case is defined:

- *the ECASD certificate is expired*

Referenced Requirements

- EUICC_REQ35, EUICC_REQ39

Initial Conditions

- None

4.3.28.2.1.1 Test Sequence N°1 – Error Case: Invalid ECASD

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-PrepareSMSRChange,`
		`#VIRTUAL_EID_RPS,`
		`#CUR_SR_S_ID_RPS)`		

2	SM-SR-UT → MNO1-S	Send the `ES4-PrepareSMSRChange` response	The `Status` is equal to `#SUCCESS`	EUICC_REQ35

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-HandoverEUICC, #EIS_EXPIREDCASD_RPS)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_ECASD 3- The Reason code is equal to #RC_EXPIRED	EUICC_REQ39

4.3.29 ES7 (SM-SR – SM-SR): AuthenticateSMSR

4.3.29.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

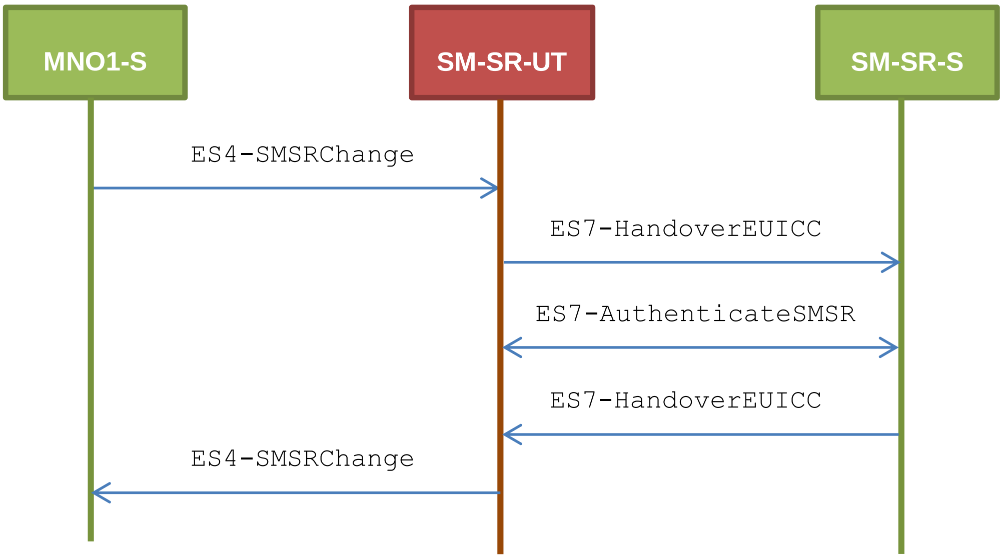
- EUICC_REQ36, EUICC_REQ39, EUICC_REQ40

4.3.29.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT

Test Environment



4.3.29.2.1 TC.ES7.ASMSR.1: AuthenticateSMSR

Test Purpose

To ensure the method AuthenticateSMSR is well implemented on the SM-SR. Only error case is defined:

- *SM-SR certificate expired*

Referenced Requirements

- EUICC_REQ36, EUICC_REQ39, EUICC_REQ40

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS

4.3.29.2.1.1 Test Sequence N°1 – Error Case: Invalid SM-SR Certificate

Initial Conditions

- The eUICC identified by the #VIRTUAL_EID is provisioned on the SM-SR-UT with the #EIS_ES1_RPS
  - o {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-SMSRChange, #VIRTUAL_EID_RPS, #TGT_SR_S_ID_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC request	The EIS is equal to #EIS_ES1_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #VIRTUAL_EID_RPS, #EXPIRED_SM_SR_CERTIFICATE)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_SR_CERTIF 3- The Reason code is equal to #RC_EXPIRED	EUICC_REQ40
5	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #FAILED, #SC_SR_CERTIF, #RC_EXPIRED)		

Step	Direction	Sequence / Description	Expected result	REQ
6	SM-SR-UT → MNO1-S	Send the ES4-SMSRChange response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_SR_CERTIF 3- The Reason code is equal to #RC_EXPIRED	EUICC_REQ39

5 System Behaviour Testing

5.1 General Overview

This section focuses on the implementation of the system according to the GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2]. The aim is to verify the functional behaviour of the system.

5.2 eUICC Behaviour

5.2.1 Device – eUICC

5.2.1.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- EUICC_REQ10, EUICC_REQ11

5.2.1.2 Test Cases

General Initial Conditions

- None

5.2.1.2.1 TC.ECASD.1: EIDRetrieval

Test Purpose

To ensure the Device can retrieve the EID by reading the ECASD information.

Referenced Requirements

- EUICC_REQ10, EUICC_REQ11

Initial Conditions

- None

5.2.1.2.1.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	`[SELECT_ECASD]`		
3	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ10, EUICC_REQ11
4	DS → eUICC-UT	`[GET_DATA_5A]`		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	TAG '5A' returned	1- TAG '5A' content: a. is equal to #EID b. starts with the byte '89' c. is 16 bytes long 2- SW='9000' 3- Using the TAG content as a decimal integer, the remainder of the division by 97 shall be equal to 1	EUICC_REQ10
Note: On this test, the basic channel 00 is used but it is assumed that a logical channel can be used				

5.2.2 LOCKED State Unsupported by ISD-R and ISD-P

5.2.2.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ7
- EUICC_REQ1, EUICC_REQ6, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

5.2.2.2 Test Cases

General Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

5.2.2.2.1 TC.LOCKISDR.1: LockISDR

Test Purpose

To ensure ISD-R cannot be locked. After trying to lock the ISD-R, an audit is performed to make sure that the lifecycle state of the security domain remains unchanged.

Referenced Requirements

- PF_REQ7
- EUICC_REQ1, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

5.2.2.2.1.1 Test Sequence N°1 – Error Case: Unable to Lock the ISD-R

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[LOCK_ISDR])`
				EUICC_REQ22

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_6985]` 65. (see Note 1)	EUICC_REQ1, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_ENABLED])`
				EUICC_REQ22

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_E3_ISDP_3F]` 66. (i.e. the ISD-R is not LOCKED)	EUICC_REQ1, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PF_REQ7
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW may be also '6A80' or '6D00' or '6A86' or '6A81'				

5.2.2.2.2 TC.LOCKISDP.1: LockISDP

Test Purpose

To ensure an ISD-P cannot be locked. After trying to lock the ISD-P, an audit is performed to make sure that the lifecycle state of the security domain remains unchanged.

Referenced Requirements

- PF_REQ7
- EUICC_REQ6, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

5.2.2.2.2.1 Test Sequence N°1 – Error Case: Unable to Lock an ISD-P

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[LOCK_DEFAULT_ISDP])`
				EUICC_REQ22

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_6985]` 67. (see Note 1)	EUICC_REQ6, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_ISDP_ENABLED])`
				EUICC_REQ22

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_E3_ISDP_3F]`	EUICC_REQ6, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PF_REQ7
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW may be also '6A80' or '6D00' or '6A86' or '6A81'				

5.2.3 Components and Visibility

5.2.3.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PM_REQ1, PM_REQ2, PM_REQ5
- EUICC_REQ2, EUICC_REQ3, EUICC_REQ8, EUICC_REQ9, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

5.2.3.2 Test Cases

General Initial Conditions

- None

5.2.3.2.1 TC.CV.1: ComponentVisibility

Test Purpose

To ensure Profile Component cannot have any visibility to components outside its ISD-P and that an ISD-P shall not have any visibility of, or access to, any other ISD-P.

Referenced Requirements

- PM_REQ2
- EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

5.2.3.2.1.1 Test Sequence N°1 – Nominal Case: No Visibility for the MNO-SD to the ISD-R

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [GET_STATUS_ISDR]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		

Step	Direction	Sequence / Description	Expected result	REQ
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_6A88]	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22; PM_REQ2
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.3.2.1.2 Test Sequence N°2 – Nominal Case: No Visibility for an ISD-P to another ISD-P

Initial Conditions

- #DEFAULT_ISD_P_AID and #ISD_P_AID1 are present on the eUICC

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #DEFAULT_ISD_P_TAR, SCP03_SCRIPT(#DEFAULT_ISD_P_SCP03_KVN, [GET_ISDP1]))		EUICC_REQ22
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed (i.e. SW='9000') 3- SW='6A88' for the GET STATUS command 68. (see Note 1)	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PM_REQ2
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW may be also '6A80' or '6D00'				

5.2.3.2.2 TC.CV.2: ISDRVisibility

Test Purpose

To ensure any component outside the ISD-P cannot have any visibility to Profile Components. In this test case, the aim is to verify that the ISD-R cannot have any visibility on the MNO-SD.

Referenced Requirements

- PM_REQ1
- EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

5.2.3.2.2.1 Test Sequence N°1 – Nominal Case: No Visibility for the ISD-R to the MNO-SD

Initial Conditions

- `#DEFAULT_ISD_P_AID` present on the eUICC

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_MNO_SD])`
				EUICC_REQ22

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_6A88]`	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PM_REQ1
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.3.2.3 TC.CV.3: ISDPNotEnabled

Test Purpose

To ensure the applications or the file system within a Disabled Profile cannot be selected. In this test case, a new Profile including an applet and a file is dynamically downloaded: the selection of these two components shall be only possible when the Profile state is updated to Enabled.

Referenced Requirements

- EUICC_REQ8, EUICC_REQ9

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)
- #ISD_P_AID1 present on the eUICC and personalized with SCP03 keys
  - The process *ES8-EstablishISDPKeySet* has been used
  - {SCP_KENC}, {SCP_KMAC}, {SCP_KDEK} have been set
- No POL1 is defined on the #DEFAULT_ISD_P_AID
- TP-Destination-Address has been set on #ISD_R_AID with #DEST_ADDR

5.2.3.2.3.1 Test Sequence N°1 - Nominal Case using CAT_TP: Applet Selectable Only on an Enabled Profile

Initial Conditions

- Applet3 (defined in A.3) is not present on the Profile linked to the #DEFAULT_ISD_P_AID
- #PE_APPLET3 defined in section B.7.3 shall be added to the #PROFILE_PACKAGE

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	Execute the test sequence defined in section 4.2.18.2.1.1 (TC.ES8.DAI.1:DownloadAndInstallation_CAT_TP) from step 3 to step 8 in order to download the #PROFILE_PACKAGE (including #PE_APPLET3) under the #ISD_P_AID1		All steps successfully executed	
4	Close CAT_TP session as described in section 4.2.1.4			
5	Execute the test sequence defined in section 4.2.19.2.1.1 (TC.ES8.UCP.1:UpdateConnectivityParameters_SMS) from step 2 to step 6 in order to set the SMS Connectivity Parameters in the #ISD_P_AID1		All steps successfully executed	
6	DS → eUICC-UT	[SELECT_APPLET3]		
7	eUICC-UT → DS	ATS	SW='6A82'	EUICC_REQ9
8	Initialization sequence as described in section 4.2.1.1			
9	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 9 in order to Enable the #ISD_P_AID1		All steps successfully executed	
10	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 1 to step 16 in order to manage the different notifications exchanged with the eUICC and to make sure that the Profile linked to the #ISD_P_AID1 is now Enabled		All steps successfully executed	
11	DS → eUICC-UT	[SELECT_APPLET3]		
12	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ9

5.2.3.2.3.2 Test Sequence N°2 - Nominal Case using HTTPS: Applet Selectable Only on an Enabled Profile

Initial Conditions

- Applet3 (defined in A.3) is not present on the Profile linked to the #DEFAULT_ISD_P_AID
- #PE_APPLET3 defined in section B.7.3 shall be added to the #PROFILE_PACKAGE

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	Execute the test sequence defined in section 4.2.18.2.2.1 (TC.ES8.DAI.2:DownloadAndInstallation_HTTPS) from step 3 to step 8 in order to download the #PROFILE_PACKAGE (including #PE_APPLET3) under the #ISD_P_AID1		All steps successfully executed	
4	Close HTTPS session as described in section 4.2.1.7			
5	Execute the test sequence defined in section 4.2.19.2.1.1 (TC.ES8.UCP.1:UpdateConnectivityParameters_SMS) from step 2 to step 6 in order to set the SMS Connectivity Parameters in the #ISD_P_AID1		All steps successfully executed	
6	DS → eUICC-UT	[SELECT_APPLET3]		
7	eUICC-UT → DS	ATS	SW='6A82'	EUICC_REQ9
8	Initialization sequence as described in section 4.2.1.1			
9	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 9 in order to Enable the #ISD_P_AID1		All steps successfully executed	
10	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 1 to step 16 in order to manage the different notifications exchanged with the eUICC and to make sure that the Profile linked to the #ISD_P_AID1 is now Enabled		All steps successfully executed	
11	DS → eUICC-UT	[SELECT_APPLET3]		
12	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ9

5.2.3.2.3.3 Test Sequence N°3 - Nominal Case using CAT_TP: File Selectable Only on an Enabled Profile

Initial Conditions

- Elementary File with the identifier '1122' is not present on the Profile linked to the #DEFAULT_ISD_P_AID
- #PE_EF1122 defined in section B.7.3 shall be added to the #PROFILE_PACKAGE

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
3	Execute the test sequence defined in section 4.2.18.2.1.1 (TC.ES8.DAI.1:DownloadAndInstallation_CAT_TP) from step 3 to step 8 in order to download the #PROFILE_PACKAGE (including #PE_EF1122) under the #ISD_P_AID1		All steps successfully executed	
4	Close CAT_TP session as described in section 4.2.1.4			
5	Execute the test sequence defined in section 4.2.19.2.1.1 (TC.ES8.UCP.1:UpdateConnectivityParameters_SMS) from step 2 to step 6 in order to set the SMS Connectivity Parameters in the #ISD_P_AID1		All steps successfully executed	
6	DS → eUICC-UT	[SELECT_FILE_1122]		
7	eUICC-UT → DS	ATS	SW='6A82'	EUICC_REQ8
8	Initialization sequence as described in section 4.2.1.1			
9	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 9 in order to Enable the #ISD_P_AID1		All steps successfully executed	
10	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 1 to step 16 in order to manage the different notifications exchanged with the eUICC and to make sure that the Profile linked to the #ISD_P_AID1 is now Enabled		All steps successfully executed	
11	DS → eUICC-UT	[SELECT_FILE_1122]		
12	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ8

5.2.3.2.3.4 Test Sequence N°4 - Nominal Case using HTTPS: File Selectable Only on an Enabled Profile

Initial Conditions

- Elementary File with the identifier '1122' is not present on the Profile linked to the #DEFAULT_ISD_P_AID
- #PE_EF1122 defined in section B.7.3 shall be added to the #PROFILE_PACKAGE

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	Execute the test sequence defined in section 4.2.18.2.2.1 (TC.ES8.DAI.2:DownloadAndInstallation_HTTPS) from step 3 to step 8 in order to download the #PROFILE_PACKAGE (including #PE_EF1122) under the #ISD_P_AID1		All steps successfully executed	

Step	Direction	Sequence / Description	Expected result	REQ
4	Close HTTPS session as described in section 4.2.1.7			
5	Execute the test sequence defined in section 4.2.19.2.1.1 (TC.ES8.UCP.1:UpdateConnectivityParameters_SMS) from step 2 to step 6 in order to set the SMS Connectivity Parameters in the `#ISD_P_AID1`		All steps successfully executed	
6	DS → eUICC-UT	`[SELECT_FILE_1122]`		
7	eUICC-UT → DS	ATS	SW='6A82'	EUICC_REQ8
8	Initialization sequence as described in section 4.2.1.1			
9	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 9 in order to Enable the `#ISD_P_AID1`		All steps successfully executed	
10	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 1 to step 16 in order to manage the different notifications exchanged with the eUICC and to make sure that the Profile linked to the `#ISD_P_AID1` is now Enabled		All steps successfully executed	
11	DS → eUICC-UT	`[SELECT_FILE_1122]`		
12	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ8

5.2.3.2.4 TC.CV.4: TarAllocation

Test Purpose

To ensure it is possible to allocate the same TAR within distinct Profiles. In this test case, an applet is installed through the MNO-SD on the Enabled Profile. Then, another applet with the same TAR is installed during the downloading of a new Profile. An error case is also defined to make sure that a Profile Component cannot use the reserved ISD-R TAR.

Referenced Requirements

- EUICC_REQ3

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)
- Applet1 and Applet2 (defined in Annex A) are not present on the default Profile identified by `#DEFAULT_ISD_P_AID`

5.2.3.2.4.1 Test Sequence N°1 - Nominal Case using CAT_TP: Same TAR within Two Profiles

Initial Conditions

- `#ISD_P_AID1` present on the eUICC and personalized with SCP03 keys
  - o The process *ES8-EstablishISDPKeySet* has been used
  - o `{SCP_KENC}, {SCP_KMAC}, {SCP_KDEK}` have been set
- Applet1 and Applet2 (defined in Annex A) are not present on the Profile identified by `#ISD_P_AID1`
- `#PE_APPLET1` defined in section B.7.3 shall be added to the `#PROFILE_PACKAGE`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`{LOAD_APPLET2};`
		`[INSTALL_APPLET2])`
		Use `#MNO_SCP80_ENC_KEY,`
		`#MNO_SCP80_AUTH_KEY,`
		`#MNO_SCP80_DATA_ENC_KEY`		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	PROACTIVE COMMAND: SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- SW='9000' for all commands	
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
8	Execute the test sequence defined in section 4.2.18.2.1.1 (TC.ES8.DAI.1:DownloadAndInstallation_CAT_TP) from step 3 to step 8 in order to download the `#PROFILE_PACKAGE` (including `#PE_APPLET1`) under the `#ISD_P_AID1`		All steps successfully executed	EUICC_REQ3
9	Close CAT_TP session as described in section 4.2.1.4			

5.2.3.2.4.2 Test Sequence N°2 - Nominal Case using HTTPS: Same TAR within Two Profiles

Initial Conditions

- `#ISD_P_AID1` present on the eUICC and personalized with SCP03 keys
  - The process *ES8-EstablishISDPKeySet* has been used
  - `{SCP_KENC}, {SCP_KMAC}, {SCP_KDEK}` have been set
- Applet1 and Applet2 (defined in Annex A) are not present on the Profile identified by `#ISD_P_AID1`
- `#PE_APPLET1` defined in section B.7.3 shall be added to the `#PROFILE_PACKAGE`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`{LOAD_APPLET2};`
		`[INSTALL_APPLET2])`
		Use `#MNO_SCP80_ENC_KEY,`
		`#MNO_SCP80_AUTH_KEY,`
		`#MNO_SCP80_DATA_ENC_KEY`		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	PROACTIVE COMMAND: SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- SW='9000' for all commands	
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	Open HTTPS session on ISD-R as described in section 4.2.1.5			
8	Execute the test sequence defined in section 4.2.18.2.2.1 (TC.ES8.DAI.2:DownloadAndInstallation_HTTPS) from step 3 to step 8 in order to download the `#PROFILE_PACKAGE` (including `#PE_APPLET1`) under the `#ISD_P_AID1`		All steps successfully executed	EUICC_REQ3
9	Close HTTPS session as described in section 4.2.1.7			

5.2.3.2.4.3 Test Sequence N°3 - Error Case: Unauthorized ISD-R TAR

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`{LOAD_APPLET1})`
		Use `#MNO_SCP80_ENC_KEY,`
		`#MNO_SCP80_AUTH_KEY,`
		`#MNO_SCP80_DATA_ENC_KEY`		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- SW='9000' for all commands	
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

7	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#MNO_TAR,`
		`[INSTALL_TAR_ISDR])`
		Use `#MNO_SCP80_ENC_KEY,`
		`#MNO_SCP80_AUTH_KEY,`
		`#MNO_SCP80_DATA_ENC_KEY`		

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- SW='6985' for the INSTALL command (see Note 1)	EUICC_REQ3
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW may be also '6A80'				

5.2.3.2.5 TC.CV.5: AIDAllocation

Test Purpose

To ensure it is possible to allocate the same AID within distinct Profiles. In this test case, an applet is installed through the MNO-SD on the Enabled Profile. Then, another applet with the same AID is installed during the downloading of a new Profile. An error case is also defined to make sure that a Profile Component cannot use the reserved ECASD AID.

Referenced Requirements

- EUICC_REQ2

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)
- Applet3 (defined in A.3) is not present on the default Profile identified by #DEFAULT_ISD_P_AID

5.2.3.2.5.1 Test Sequence N°1 - Nominal Case using CAT_TP: Same AID within Two Profiles

Initial Conditions

- #ISD_P_AID1 present on the eUICC and personalized with SCP03 keys
  - The process *ES8-EstablishISDPKeySet* has been used
  - {SCP_KENC}, {SCP_KMAC}, {SCP_KDEK} have been set
- Applet3 (defined in A.3) is not present on the Profile identified by #ISD_P_AID1
- #PE_APPLET3 defined in section B.7.3 shall be added to the #PROFILE_PACKAGE

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, {LOAD_APPLET3}; [INSTALL_APPLET3]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- SW='9000' for all commands	

Step	Direction	Sequence / Description	Expected result	REQ
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
8		Execute the test sequence defined in section 4.2.18.2.1.1 (TC.ES8.DAI.1:DownloadAndInstallation_CAT_TP) from step 3 to step 8 in order to download the `#PROFILE_PACKAGE` (including `#PE_APPLET3`) under the `#ISD_P_AID1`	All steps successfully executed	EUICC_REQ2
9	Close CAT_TP session as described in section 4.2.1.4			

5.2.3.2.5.2 Test Sequence N°2 - Nominal Case using HTTPS: Same AID within Two Profiles

Initial Conditions

- `#ISD_P_AID1` present on the eUICC and personalized with SCP03 keys
  - The process *ES8-EstablishISDPKeySet* has been used
  - `{SCP_KENC}, {SCP_KMAC}, {SCP_KDEK}` have been set
- Applet3 (defined in A.3) is not present on the Profile identified by `#ISD_P_AID1`
- `#PE_APPLET3` defined in section B.7.3 shall be added to the `#PROFILE_PACKAGE`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, {LOAD_APPLET3}; [INSTALL_APPLET3]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- SW='9000' for all commands	
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Step	Direction	Sequence / Description	Expected result	REQ
7	Open HTTPS session on ISD-R as described in section 4.2.1.5			
8	Execute the test sequence defined in section 4.2.18.2.2.1 (TC.ES8.DAI.2:DownloadAndInstallation_HTTPS) from step 3 to step 8 in order to download the `#PROFILE_PACKAGE` (including `#PE_APPLET3`) under the `#ISD_P_AID1`		All steps successfully executed	EUICC_REQ2
9	Close HTTPS session as described in section 4.2.1.7			

5.2.3.2.5.3 Test Sequence N°3 - Error Case: Unauthorized ECASD AID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, {LOAD_APPLET3}) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- SW='9000' for all commands	
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_AID_ECASD]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		

Step	Direction	Sequence / Description	Expected result	REQ
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- SW='6985' for the INSTALL command (see Note 1)	EUICC_REQ2
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW may be also '6A80'				

5.2.3.2.6 TC.CV.6: MNOSDDefinition

Test Purpose

To ensure the MNO-SD AID and TAR can be freely allocated during the Profile definition. In this test case, a GET STATUS is sent to the MNO-SD to retrieve its information.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
- PM_REQ5

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

5.2.3.2.6.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [GET_MNO_ISD]) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		EUICC_REQ22
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#MNO_SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `#MNO_SCP80_AUTH_KEY` 3- The response data is equal to `[R_AB_MNO_SD]`	PM_REQ5, EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.4 Security and Responsibility

5.2.4.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ1
- SEC_REQ6
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ19, EUICC_REQ20, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ54, EUICC_REQ55, EUICC_REQ56, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61

5.2.4.2 Test Cases

General Initial Conditions

- None

5.2.4.2.1 TC.SAR.1: SecurityError_SMS

Test Purpose

To ensure a SMS shall be rejected by the eUICC (i.e. no POR returned) when:

- *the security level does not meet the one expected by the ISD-R*
- *the SM-SR is not authenticated*

Referenced Requirements

- EUICC_REQ20

Initial Conditions

- None

5.2.4.2.1.1 Test Sequence N°1 – Error Case: Low Security Level

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#BAD_SPI,`
		`#ISD_R_TAR,`
		`[GET_DEFAULT_ISDP])`		

3	eUICC-UT → DS	*NO PROACTIVE COMMAND PENDING*	No SMS POR sent SW='9000'	EUICC_REQ20

5.2.4.2.1.2 Test Sequence N°2 – Error Case: eUICC cannot Authenticate the SM-SR

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DEFAULT_ISDP])`
		Do not use the `#SCP80_ENC_KEY, #SCP80_AUTH_KEY, #SCP80_DATA_ENC_KEY`
		see Note		

3	eUICC-UT → DS	*NO PROACTIVE COMMAND PENDING*	No SMS POR sent SW='9000'	EUICC_REQ20
Note: The correct ISD-R SCP80 keys shall not be used. Other values with same length can be freely chosen.				

5.2.4.2.2 TC.SAR.2: ISDRResponsibility

Test Purpose

To ensure only ISD-R can create an ISD-P.

Referenced Requirements

- PF_REQ1
- EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

5.2.4.2.2.1 Test Sequence N°1 - Error Case: ISD-P Cannot Create another ISD-P

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#DEFAULT_ISD_P_TAR,`
		`SCP03_SCRIPT(`
		`#DEFAULT_ISD_P_SCP03_KVN,`
		`[INSTALL_ISDP]))`
				EUICC_REQ22

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed (i.e. SW='9000') 3- The SW is '6985' for the INSTALL command 69. (see Note 1)	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PF_REQ1
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW may be also '6A80' or '6A88'				

5.2.4.2.3 TC.SAR.3: ReplayAttack

Test Purpose

To ensure the communication between the SM-SR and the eUICC is protected against replay attacks. In this test case, the same secured packet is sent twice to make sure that only the first one is accepted by the eUICC.

Referenced Requirements

- SEC_REQ6
- EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

Initial Conditions

- None

5.2.4.2.3.1 Test Sequence N°1 - Error Case: Same Secured Packet Not Accepted

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[GET_DEFAULT_ISDP])`
				EUICC_REQ22

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `170.#SCP80_AUTH_KEY` 3- The response data is in expanded format with definite length	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	Send exactly the same SMS as the previous one		EUICC_REQ22
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE	see Note	
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- Verify the cryptographic checksum using `171.#SCP80_AUTH_KEY` 3- No response data is returned 4- The status code is equal to '02' - Counter low	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, SEC_REQ6
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: Depending on the implementation, the eUICC may decide to not send back a POR (i.e. SW '9000' on the ENVELOPE command). Therefore, the steps 8, 9, 10 and 11 shall be considered as optional.				

5.2.4.2.4 TC.SAR.4: HTTPSRestrictions

Test Purpose

To ensure the following HTTPS restrictions are well configured on the ISD-R:

- *TLS 1.2 shall only be supported meaning that the 'i' parameter is set to '04'*
- *session resumption shall not be supported*
- *several parallel sessions shall not be supported*

Referenced Requirements

- EUICC_REQ13, EUICC_REQ14, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ54, EUICC_REQ55, EUICC_REQ56

Initial Conditions

- None

5.2.4.2.4.1 Test Sequence N°1 - Nominal Case: TLS 1.2 only Supported by ISD-R

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.1 [15] shall be supported
  - Only the cipher-suite TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
    - Note: the cipher-suite TLS_PSK_WITH_AES_128_GCM_SHA256 cannot be used here as it shall be only negotiated using TLS version 1.2
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [OPEN_SCP81_SESSION])		EUICC_REQ22, EUICC_REQ42, EUICC_REQ54
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- The SCP80 status code is equal to '00' – POR OK	EUICC_REQ21
6	DS → eUICC-UT	TERMINAL RESPONSE		
7	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
8	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
9	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	1- The bearer description is equal to `#BEARER_DESCRIPTION` 2- The buffer size is equal to `#BUFFER_SIZE` 3- The NAN is equal to `#NAN_VALUE` 4- The port is equal to `#TCP_PORT` 5- The IP is equal to `#IP_VALUE`	EUICC_REQ13, EUICC_REQ14, EUICC_REQ42
10	DS → eUICC-UT	TERMINAL RESPONSE		
11	*For readability reason, the proactive commands are not fully specified in the next steps.* *The BIP communication between the DS and the eUICC-UT shall be compliant with the Annex F.* *The TLS records used here after shall be compliant with the 0.*			
12	eUICC-UT → DS	TLS_CLIENT_HELLO		EUICC_REQ14, EUICC_REQ43
13	DS → eUICC-UT	TLS_1_1_SERVER_HELLO and TLS_1_1_SERVER_HELLO_DONE		
14	eUICC-UT → DS	TLS_ALERT_PROTOCOL_VERSION		EUICC_REQ55
15	eUICC-UT → DS	*PROACTIVE COMMAND:* CLOSE CHANNEL	The HTTP session is closed.	EUICC_REQ55
16	DS → eUICC-UT	TERMINAL RESPONSE		

5.2.4.2.4.2 Test Sequence N°2 - Nominal Case: No HTTPS Session Resumption

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
4	eUICC-UT → DS	*NO PROACTIVE COMMAND PENDING*	No OPEN CHANNEL sent	EUICC_REQ56
5	Execute the test sequence defined in section 4.2.2.2.3.1 (TC.TP.HTTPS.3:Transport_HTTPs) from step 2 to step 5 in order to make sure that a new HTTPs session can be open		All steps successfully executed	EUICC_REQ56

5.2.4.2.4.3 Test Sequence N°3 - Nominal Case: No HTTPS Sessions in Parallel

Initial Conditions

- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: `#PSK_ID`
    - PSK value: `#SCP81_PSK`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [OPEN_SCP81_SESSION])		EUICC_REQ22, EUICC_REQ42, EUICC_REQ54
4	eUICC-UT → DS	*NO OPEN CHANNEL COMMAND PENDING*	A new HTTPS session shall not be open (see Note)	EUICC_REQ56
Note: Depending on the implementation, a SMS POR may be returned by the eUICC with an incorrect SW (e.g. '9300').				

5.2.4.2.5 TC.SAR.5: SCP03t_ErrorManagement

Test Purpose

To ensure SCP03t is well implemented on the eUICC. This test case proposes to check that a dedicated error (e.g. reference data not found, error in length, security error) is returned when incorrect SCP03t command is sent.

Note that all the following error cases propose to send small SCP03t scripts over SMS. Depending on the eUICC implementation, it may be necessary to run these tests only over HTTPS or CAT_TP.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61

Initial Conditions

- `#ISD_P_AID1` present on the eUICC and personalized with SCP03 keys

- The process ES8-EstablishISDPKeySet has been used
- {SCP_KENC}, {SCP_KMAC}, {SCP_KDEK} have been set

5.2.4.2.5.1 Test Sequence N°1 – Error Case: Incorrect Length in INITIALIZE UPDATE

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_P_TAR1, SCP03T_SCRIPT(#SCP03_KVN, #PE_HEADER)) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK} Change the length value of the INITIALIZE UPDATE TLV command before sending the script (e.g. with '11' instead of '0A')		
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- The response data is equal to [R_AB_SCP03T_IU_01]	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ59
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.4.2.5.2 Test Sequence N°2 – Error Case: Incorrect Parameter in INITIALIZE UPDATE

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03T_SCRIPT(`
		`#BAD_SCP03_KVN,`
		`#PE_HEADER))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_SCP03T_IU_03]`	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ59
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.4.2.5.3 Test Sequence N°3 – Error Case: Incorrect Length in EXTERNAL AUTHENTICATE

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03T_SCRIPT(`
		`#SCP03_KVN,`
		`#PE_HEADER))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
		Change the length value of the EXTERNAL AUTHENTICATE TLV command (TAG '85') before sending the script (e.g. with '19' instead of '11')		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_SCP03T_EA_01]`	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ60
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.4.2.5.4 Test Sequence N°4 – Error Case: Incorrect Security in EXTERNAL AUTHENTICATE

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03T_SCRIPT(`
		`#SCP03_KVN,`
		`#PE_HEADER))`
		Do not use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
		see Note		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_SCP03T_EA_02]`	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ60
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: The correct ISD-P SCP03 keys shall not be used. Other values with same length can be freely chosen.				

5.2.4.2.5.5 Test Sequence N°5 – Error Case: Incorrect Length in Profile TLV Command

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`    #SPI_VALUE,`
		`    #ISD_P_TAR1,`
		`    SCP03T_SCRIPT(`
		`        #SCP03_KVN,`
		`        #PE_HEADER))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
		Change the length value of the Profile data TLV command (TAG '86') before sending the script		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the `#SCP80_ENC_KEY` 2- The response data is equal to `[R_AB_SCP03T_01]`	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ61
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.4.2.5.6 Test Sequence N°6 – Error Case: Incorrect Security in Profile TLV Command

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`    #SPI_VALUE,`
		`    #ISD_P_TAR1,`
		`    SCP03T_SCRIPT(`
		`        #SCP03_KVN,`
		`        #PE_HEADER))`
		Use the SCP03 keys `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}`
		Corrupt a block of ciphered data in the Profile data TLV command (TAG '86') before sending the script		

3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #SCP80_ENC_KEY 2- The response data is equal to [R_AB_SCP03T_02]	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ61
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.5 Confidential Setup of MNO Secure Channel Keys

5.2.5.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]

Requirements

- SEC_REQ20

5.2.5.2 Test Cases

General Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (shall be the initial state of the eUICC)

5.2.5.2.1 TC.CSMNOSCK.1: Scenario#2.B

Test Purpose

To ensure MNO can update the OTA Keys on its Profile using the scenario #2.B as defined in GlobalPlatform Card Specification v.2.2.1 - UICC Configuration [13].

Referenced Requirements

- SEC_REQ20

Initial Conditions

- None

5.2.5.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by the eUICC	
2	DS → eUICC-UT	[SELECT_CASD]		
3	eUICC-UT → DS	ATS	SW='9000'	SEC_REQ20
4	DS → eUICC-UT	[GET_DATA_CASD_CERT]		

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	DGI '7F21' returned	1- The response is equal to [R_CASD] 2- The {PK_CASD_CT} shall be recovered from the signature using the #EUM_PK_CA_AUT	SEC_REQ20
6	Initialization sequence as described in section 4.2.1.1			
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, STORE_MNO_KEYS_2B({PK_CASD_CT})) Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_9000]	SEC_REQ20
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: After the execution of this test, all the MNO-SD keysets should be deleted except the one identified by #MNO_SCP80_KVN				

5.2.5.2.2 TC.CSMNOSCK.2: Scenario#3

Test Purpose

To ensure MNO can update the OTA Keys on its Profile using the scenario #3 as defined in GlobalPlatform Card Specification v.2.2 Amendment E: Security Upgrade for Card Content Management [13].

Referenced Requirements

- SEC_REQ20

Initial Conditions

- None

5.2.5.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	DS → eUICC-UT	RESET	ATR returned by the eUICC	
2	DS → eUICC-UT	[SELECT_CASD]		
3	eUICC-UT → DS	ATS	SW='9000'	SEC_REQ20
4	DS → eUICC-UT	[GET_DATA_CASD_CERT]		
5	eUICC-UT → DS	DGI '7F21' returned	1- The response is equal to [R_CASD] 2- The {PK_CASD_CT} shall be retrieved from the TAG '7F49'	SEC_REQ20
6	Initialization sequence as described in section 4.2.1.1			

7	DS → eUICC-UT	ENVELOPE_SMS_PP(
		#SPI_VALUE,
		#MNO_TAR,
		STORE_MNO_KEYS_3())
		Use #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY, #MNO_SCP80_DATA_ENC_KEY		

8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* SEND SHORT MESSAGE	1- Decrypt the response packet with the #MNO_SCP80_ENC_KEY 2- Verify the cryptographic checksum using 72. #MNO_SCP80_AUTH_KEY 3- The response data is equal to [R_AB_RECEIPT] 4- Calculate ShS from #SM_ESK_ECKA and {PK_CASD_CT} 5- Derive keyset from ShS and retrieve the {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK} 6- Verify the {RECEIPT} (i.e. it shall be generated by calculating a MAC across the tag 'A6')	SEC_REQ20
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.6 Full Profile Installation Process

5.2.6.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PROC_REQ1, PROC_REQ2, PROC_REQ3, PROC_REQ7, PROC_REQ19, PROC_REQ21

5.2.6.2 Test Cases

General Initial Conditions

- ISD-P `#ISD_P_AID1` not present on the eUICC
- `#DEFAULT_ISD_P_AID` in Enabled state (shall be the initial state of the eUICC)
- No POL1 is defined on the `#DEFAULT_ISD_P_AID`

5.2.6.2.1 TC.FPIP.1: ProfileDownloadAndEnabling

Test Purpose

To ensure a Profile can be fully downloaded using only one OTA session and Enabled. Here are the different steps that are executed:

- *ISD-P creation*
- *ISD-P keys establishment with scenario #3*
- *Download and installation of a Profile*
- *Profile enabling*

The test sequences below propose to execute these steps using either CAT_TP or HTTPS. Between each step related to the Profile Downloading process, no operation is performed on the eUICC during a delay of 30 seconds in order to simulate exchanges related to the off-card interfaces.

Referenced Requirements

- PROC_REQ1, PROC_REQ2, PROC_REQ3, PROC_REQ7, PROC_REQ19, PROC_REQ21

Initial Conditions

- None

5.2.6.2.1.1 Test Sequence N°1 – Nominal Case: Using CAT_TP

Initial Conditions

- CAT_TP Connectivity Parameters have been set on `#ISD_R_AID` with `#UDP_PORT, #CAT_TP_PORT` and `#IP_VALUE`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open CAT_TP session on ISD-R as described in section 4.2.1.2			

Step	Direction	Sequence / Description	Expected result	REQ
3	Execute the test sequence defined in section 4.2.3.2.2.1 (TC.ES5.CISDP.2:CreateISDP_CAT_TP) from step 3 to step 4 in order to create the `#ISD_P_AID1`		All steps successfully executed	PROC_REQ1
Maintain open the CAT_TP session for 30 seconds by sending an ACK NUL every 10 seconds *(as defined in steps 4 and 5)*				
4	DS → eUICC-UT	ACK_NUL		
5	eUICC-UT → DS	ACK_NO_DATA		
Third ACK NUL sent (Timer of 30 seconds reached)				
6	Execute the test sequence defined in section 4.2.17.2.2.1 (TC.ES8.EISDPK.2:EstablishISDPkeyset_CAT_TP) from step 3 to step 4 in order to start the personalization of the `#ISD_P_AID1`		All steps successfully executed	PROC_REQ2
7	Maintain open the CAT_TP session for 30 seconds by executing steps 4 and 5 of this sequence			
8	Execute the test sequence defined in section 4.2.17.2.2.1 (TC.ES8.EISDPK.2:EstablishISDPkeyset_CAT_TP) from step 5 to step 6 in order to finish the personalization of the `#ISD_P_AID1`		All steps successfully executed	PROC_REQ2
9	Maintain open the CAT_TP session for 30 seconds by executing steps 4 and 5 of this sequence			
10	Execute the test sequence defined in section 4.2.18.2.1.1 (TC.ES8.DAI.1:DownloadAndInstallation_CAT_TP) from step 3 to step 8 in order to download the `#PROFILE_PACKAGE` under the `#ISD_P_AID1`		All steps successfully executed	PROC_REQ3
11	Close CAT_TP session as described in section 4.2.1.4			
12	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
13	Execute the test sequence defined in section 4.2.19.2.2.1 (TC.ES8.UCP.2:UpdateConnectivityParameters_CAT_TP) from step 3 to step 4 in order to set the CAT_TP Connectivity Parameters in the `#ISD_P_AID1`		All steps successfully executed	PROC_REQ19
14	Close CAT_TP session as described in section 4.2.1.4			
15	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
16	Execute the test sequence defined in section 4.2.4.2.2.1 (TC.ES5.EP.2:EnableProfile_CAT_TP) from step 3 to step 8 in order to Enable the `#ISD_P_AID1`		All steps successfully executed	PROC_REQ7
17	Execute the test sequence defined in section 4.2.13.2.2 (TC.ES5.NOTIFPE.2:Notification_CAT_TP) from step 1 to step 18 in order to manage the different notifications exchanged with the eUICC and to make sure that the Profile linked to the `#ISD_P_AID1` is now Enabled		All steps successfully executed	

5.2.6.2.1.2 Test Sequence N°2 – Nominal Case: Using HTTPS

Initial Conditions

- HTTPS Connectivity Parameters have been set on #ISD_R_AID with #TCP_PORT, #IP_VALUE, #ADMIN_HOST, #AGENT_ID, #PSK_ID, #SCP81_KVN, #SCP81_KEY_ID and #ADMIN_URI
- The HTTPS server shall be configured as follow:
  - Only the version TLS Protocol 1.2 [8] shall be supported
  - Only the cipher-suites TLS_PSK_WITH_AES_128_GCM_SHA256 and TLS_PSK_WITH_AES_128_CBC_SHA256 as defined in RFC 5487 [9] shall be accepted
  - The following Pre-Shared Key shall be defined:
    - PSK identifier: #PSK_ID
    - PSK value: #SCP81_PSK

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Open HTTPS session on ISD-R as described in section 4.2.1.5			
3	Execute the test sequence defined in section 4.2.3.2.3.1 (TC.ES5.CISDP.3:CreateISDP_HTTPS) from step 3 to step 4 in order to create the #ISD_P_AID1		All steps successfully executed	PROC_REQ1
4	Do not send any TLS records during 30 seconds (the HTTPS session shall remain open)			
5	Execute the test sequence defined in section 4.2.17.2.3.1 (TC.ES8.EISDPK.3:EstablishISDPkeyset_HTTPS) from step 3 to step 4 in order to start the personalization of the #ISD_P_AID1		All steps successfully executed	PROC_REQ2
6	Do not send any TLS records during 30 seconds (the HTTPS session shall remain open)			
7	Execute the test sequence defined in section 4.2.17.2.3.1 (TC.ES8.EISDPK.3:EstablishISDPkeyset_HTTPS) from step 5 to step 6 in order to finish the personalization of the #ISD_P_AID1		All steps successfully executed	PROC_REQ2
8	Do not send any TLS records during 30 seconds (the HTTPS session shall remain open)			
9	Execute the test sequence defined in section 4.2.18.2.2.1 (TC.ES8.DAI.2:DownloadAndInstallation_HTTPS) from step 3 to step 8 in order to download the #PROFILE_PACKAGE under the #ISD_P_AID1		All steps successfully executed	PROC_REQ3
10	Close HTTPS session as described in section 4.2.1.7			
11	Open HTTPS session on ISD-R as described in section 4.2.1.5			
12	Execute the test sequence defined in section 4.2.19.2.3.1 (TC.ES8.UCP.3:UpdateConnectivityParameters_HTTPS) from step 3 to step 4 in order to set the HTTPS Connectivity Parameters in the #ISD_P_AID1		All steps successfully executed	PROC_REQ19
13	Close HTTPS session as described in section 4.2.1.7			
14	Open HTTPS session on ISD-R as described in section 4.2.1.5			

Step	Direction	Sequence / Description	Expected result	REQ
15	Execute the test sequence defined in section 4.2.4.2.3.1 (TC.ES5.EP.3:EnableProfile_HTTPS) from step 3 to step 8 in order to Enable the #ISD_P_AID1		All steps successfully executed	PROC_REQ7
16	Execute the test sequence defined in section 4.2.13.2.3.1 (TC.ES5.NOTIFPE.3:Notification_HTTPS) from step 1 to step 19 in order to manage the different notifications exchanged with the eUICC and to make sure that the Profile linked to the #ISD_P_AID1 is now Enabled		All steps successfully executed	PROC_REQ21

5.3 Platform Behaviour

5.3.1 eUICC Identity Check

5.3.1.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- SEC_REQ15
- PROC_REQ1
- PM_REQ11, PM_REQ14
- EUICC_REQ35, EUICC_REQ39

5.3.1.2 Test Cases

General Initial Conditions

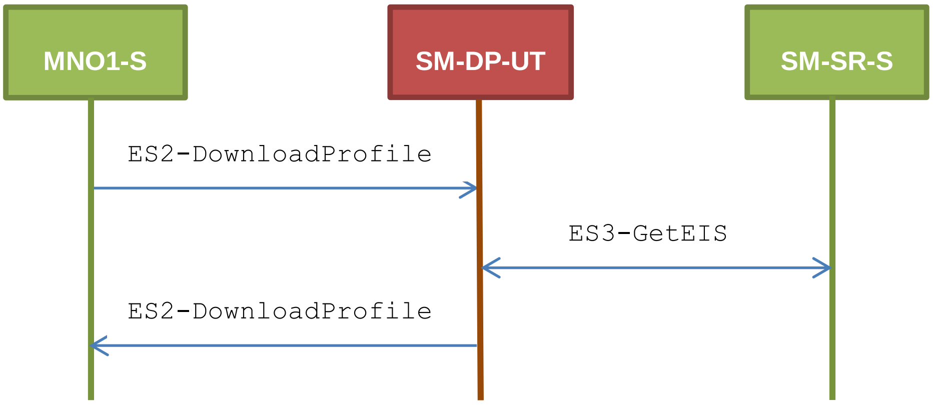
- None

5.3.1.2.1 TC.EUICCIC.1: eUICCEligibilitySMDP

Test Purpose

To ensure SM-DP is able to check the validity of an eUICC. In case of a bad ECASD in the eUICC, the SM-DP shall be able to refuse the download of the Profile.

Test Environment



Referenced Requirements

- SEC_REQ15
- PROC_REQ1
- PM_REQ11, PM_REQ14

Initial Conditions

- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_S_ID_RPS
- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #SM_SR_S_ID and #SM_SR_S_ACCESSPOINT well known to the SM-DP-UT
- #EUM_S_PK_ECDSA well known to the SM-DP-UT
- The Profile #ICCID1 is well known to the SM-DP-UT

5.3.1.2.1.1 Test Sequence N°1 – Error Case: Invalid Signature in ECASD Certificate

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DownloadProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS, #EP_FALSE_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-GetEIS request	The EID parameter is equal to #VIRTUAL_EID_RPS	PROC_REQ1, PM_REQ11, PM_REQ14

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-GetEIS, #EIS_BADCASDSIGN_RPS)		
4	SM-DP-UT → MNO1-S	Send the ES2-DownloadProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_ECASD	PM_REQ11, SEC_REQ15

5.3.1.2.1.2 Test Sequence N°2 – Error Case: Invalid CI Public Key in ECASD

Initial Conditions

- None

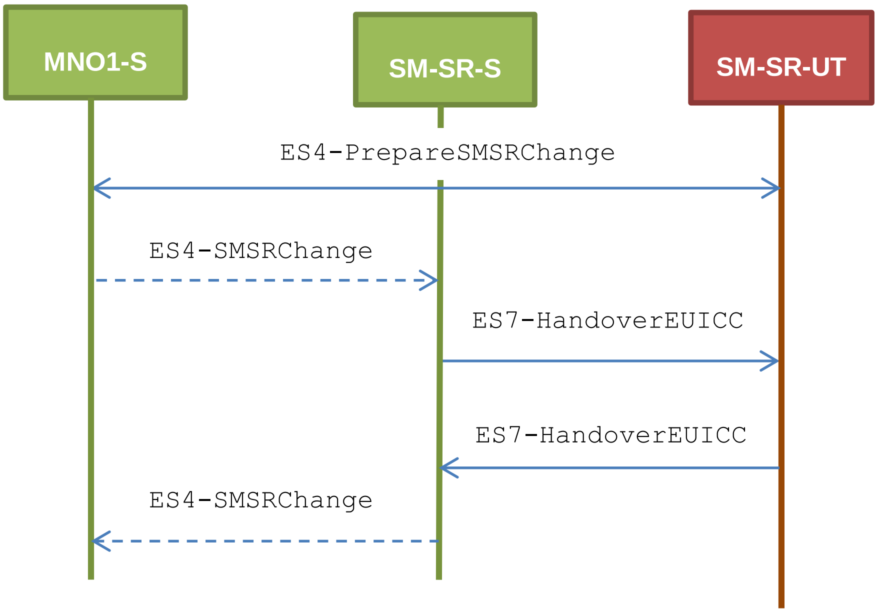
Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DownloadProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS, #EP_FALSE_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-GetEIS request	The EID parameter is equal to #VIRTUAL_EID_RPS	PROC_REQ1, PM_REQ11, PM_REQ14
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-GetEIS, #EIS_BADCASDKEY_RPS)		
4	SM-DP-UT → MNO1-S	Send the ES2-DownloadProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_ECASD	PM_REQ11, SEC_REQ15

5.3.1.2.2 TC.EUICCIC.2: eUICCEligibilitySMSR

Test Purpose

To ensure SM-SR is able to check the validity of an eUICC. In case of a bad ECASD in the eUICC, the SM-SR shall be able to refuse the change of a SM-SR.

Test Environment



Note that the function `ES4-SMSRChange` shall not be performed by the simulators (in the schema above, they are only informative messages).

Referenced Requirements

- SEC_REQ15
- EUICC_REQ35, EUICC_REQ39

Initial Conditions

- The variable `{SM_SR_ID_RPS}` shall be set to `#SM_SR_S_ID_RPS`
- `#MNO1_S_ID` and `#MNO2_S_ID` well known to the SM-SR-UT (because Profiles related to these operators are present in the EIS)
- The eUICC identified by the `#VIRTUAL_EID` is not provisioned on the SM-SR-UT
- `#EUM_S_PK_ECDSA` well known to the SM-SR-UT
- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)

5.3.1.2.2.1 Test Sequence N°1 – Error Case: Invalid Signature in ECASD Certificate

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-PrepareSMSRChange, #VIRTUAL_EID_RPS, #CUR_SR_S_ID_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-PrepareSMSRChange response	The Status is equal to #SUCCESS	EUICC_REQ35
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-HandoverEUICC, #EIS2_BADCASDSIGN_RPS)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_ECASD	EUICC_REQ39, SEC_REQ15

5.3.1.2.2.2 Test Sequence N°2 – Error Case: Invalid CI Public Key in ECASD

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-PrepareSMSRChange, #VIRTUAL_EID_RPS, #CUR_SR_S_ID_RPS)		
2	SM-SR-UT → MNO1-S	Send the ES4-PrepareSMSRChange response	The Status is equal to #SUCCESS	EUICC_REQ35
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-HandoverEUICC, #EIS2_BADCASDKEY_RPS)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_ECASD	EUICC_REQ39, SEC_REQ15

5.3.2 Profile Download and Installation Process

5.3.2.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PROC_REQ1, PROC_REQ2, PROC_REQ3, PROC_REQ7, PROC_REQ20
- PM_REQ3, PM_REQ4, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17, PM_REQ18, PM_REQ22, PM_REQ25
- PF_REQ2, PF_REQ3, PF_REQ4, PF_REQ7, PF_REQ18, PF_REQ27
- EUICC_REQ27, EUICC_REQ29, EUICC_REQ42, EUICC_REQ53

5.3.2.2 Test Cases

General Initial Conditions

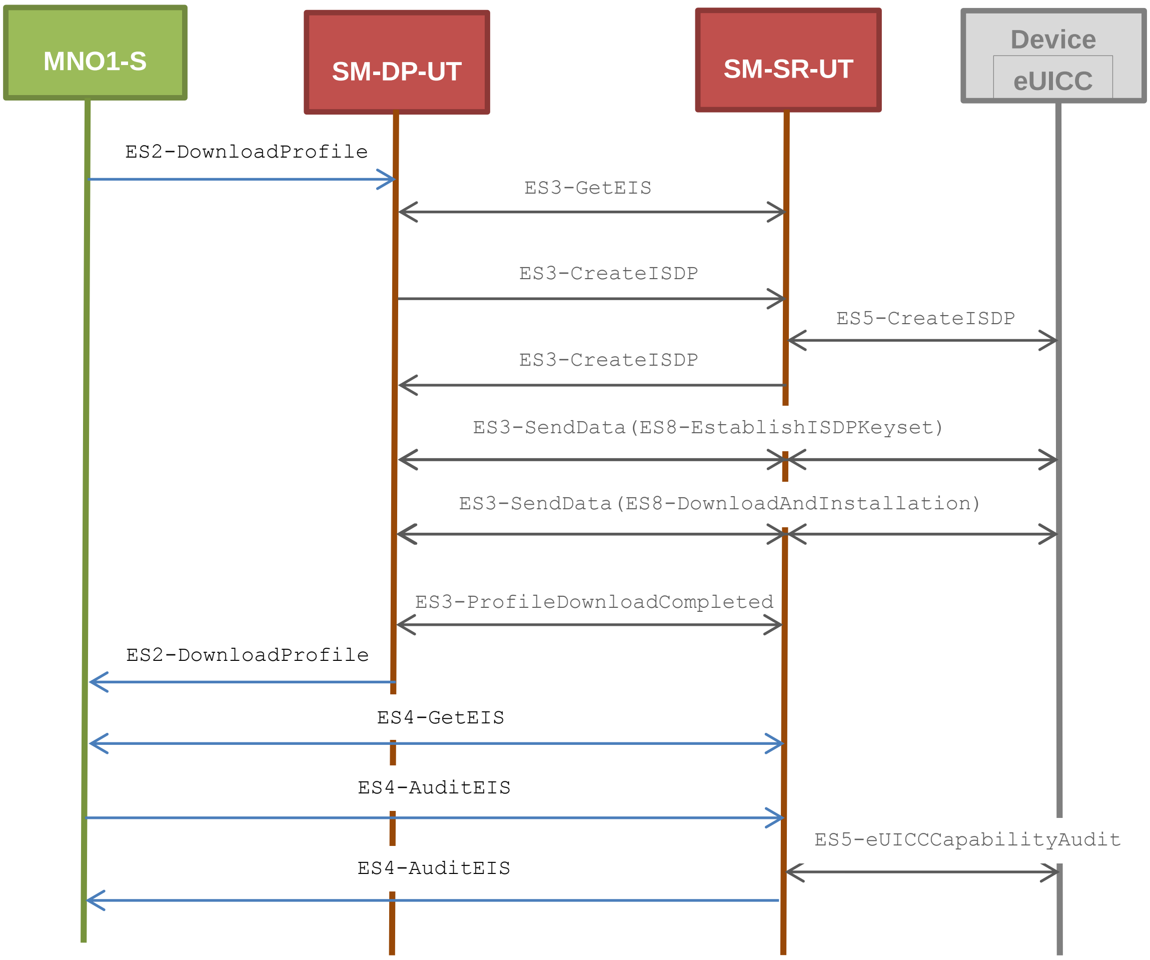
- `#MNO1_S_ID` and `#MNO1_S_ACCESSPOINT` well known to the SM-DP-UT
- `#MNO1_S_ID` well known to the SM-SR-UT
- `#MNO1_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT
- The variable `{SM_SR_ID_RPS}` shall be set to `#SM_SR_UT_ID_RPS`
- `#SM_SR_ID` and `#SM_SR_ACCESSPOINT` well known to the SM-DP-UT
- `#SM_DP_ID` and `#SM_DP_ACCESSPOINT` well known to the SM-SR-UT
- The Profile identified by `#ICCID` is owned by MNO2-S and is in Enabled state
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by `#ICCID`)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the current Enabled Profile (i.e. `#MNO2_CON_NAN, #MNO2_CON_LOGIN, #MNO2_CON_PWD`)
- SM-DP-UT is responsible for downloading and installation of the Profile identified by `#NEW_ICCID`
  - A Profile similar to `#PROFILE_PACKAGE` shall be stored on the SM-DP-UT and compatible with the eUICC
  - The Profile shall be associated with the Subscription Address `#NEW_MSISDN`

5.3.2.2.1 TC.PROC.DIP.1: DownloadAndInstallProfile

Test Purpose

To ensure that the Profile download and installation procedure is properly implemented on the SM-DP and the SM-SR. After the Profile download execution, an audit request is sent to the SM-SR to make sure that the Profile has been downloaded. The OTA capabilities set during the eUICC registration allow the use of CAT_TP or HTTPS during the download process.

Test Environment



Referenced Requirements

- EUICC_REQ42, EUICC_REQ53
- PROC_REQ1, PROC_REQ2, PROC_REQ3
- PM_REQ3, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17, PM_REQ18, PM_REQ22, PM_REQ25
- PF_REQ2, PF_REQ3, PF_REQ7

Initial Conditions

- None

5.3.2.2.1.1 Test Sequence N°1 - Nominal Case: Using CAT_TP

Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_RPS
    - the #EIS_RPS shall be adapted to indicate that the eUICC does not support HTTPS
    - the capabilities #CATTP_CAP_RPS shall be used in the #EIS_RPS

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DownloadProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS,`
		`#EP_FALSE_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the `ES2-DownloadProfile` response	1- The `Status` is equal to `#SUCCESS` 2- The `ICCID` returned is equal to `#NEW_ICCID_RPS`	PROC_REQ1,PROC_REQ2,PROC_REQ3, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17, PM_REQ18, PF_REQ2, PF_REQ3, EUICC_REQ53

4	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS})`		

5	SM-DP-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` returned contains the new Profile information (i.e. identified by `#NEW_ICCID`) 3- The new Profile information has a state equal to Disabled 4- The new Profile information has the SM-DP identifier set to `#SM-DP-ID` 5- The new Profile information has an ISD-P RID equal to `#ISD_P_RID` 6- The new Profile information has an ISD-P PIX that starts with `#ISD_P_PIX_PREFIX` 7- The new Profile information has a MNO-ID equal to `#MNO1_S_ID` 8- The new Profile information has the Subscription Address equal to `#NEW_MSISDN`	PM_REQ3, PM_REQ22

Step	Direction	Sequence / Description	Expected result	REQ
6	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS, #NEW_ICCID_RPS)		
7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS parameter is equal to that received in step 5 except that: a. the free memory of the new Profile is updated (i.e. lower than that received in step 5) b. the remaining memory and the available memory for Profiles are updated (i.e. lower than that received in step 5)	PM_REQ25, PF_REQ2, PF_REQ7

5.3.2.2.1.2 Test Sequence N°2 - Nominal Case: Using HTTPS

Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_RPS
  - the #EIS_RPS shall be adapted to indicate that the eUICC does not support CAT_TP
  - the capabilities #HTTPS_CAP_RPS shall be used in the #EIS_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DownloadProfile, #EID_RPS, {SM_SR_ID_RPS}, #NEW_ICCID_RPS, #EP_FALSE_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-DP-UT → MNO1-S	Send the `ES2-DownloadProfile` response	1- The `Status` is equal to `#SUCCESS` 2- The `ICCID` returned is equal to `#NEW_ICCID_RPS`	PROC_REQ1,PROC_REQ2,PROC_REQ3, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17, PM_REQ18, PF_REQ2, PF_REQ3, EUICC_REQ42

4	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS})`		

5	SM-DP-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` returned contains the new Profile information (i.e. identified by `#NEW_ICCID`) 3- The new Profile information has a state equal to Disabled 4- The new Profile information has the SM-DP identifier set to `#SM-DP-ID` 5- The new Profile information has an ISD-P RID equal to `#ISD_P_RID` 6- The new Profile information has an ISD-P PIX that starts with `#ISD_P_PIX_PREFIX` 7- The new Profile information has a MNO-ID equal to `#MNO1_S_ID` 8- The new Profile information has the Subscription Address equal to `#NEW_MSISDN`	PM_REQ3, PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` parameter is equal to that received in step 5 except that: a. the `free memory` of the new Profile is updated (i.e. lower than that received in step 5) b. the `remaining memory` and the `available memory for Profiles` are updated (i.e. lower than that received in step 5)	PM_REQ25, PF_REQ2, PF_REQ7

5.3.2.2.2 TC.PROC.DIP.2: DownloadAndInstallAndEnableProfile

Test Purpose

To ensure that the Profile download process followed by the Enable procedure is properly implemented on the SM-DP and the SM-SR. After the Profile download execution, an audit request is sent to the SM-SR to make sure that the Profile has been Enabled. An error case is also described to illustrate the platforms behaviour in case of enabling error.

Referenced Requirements

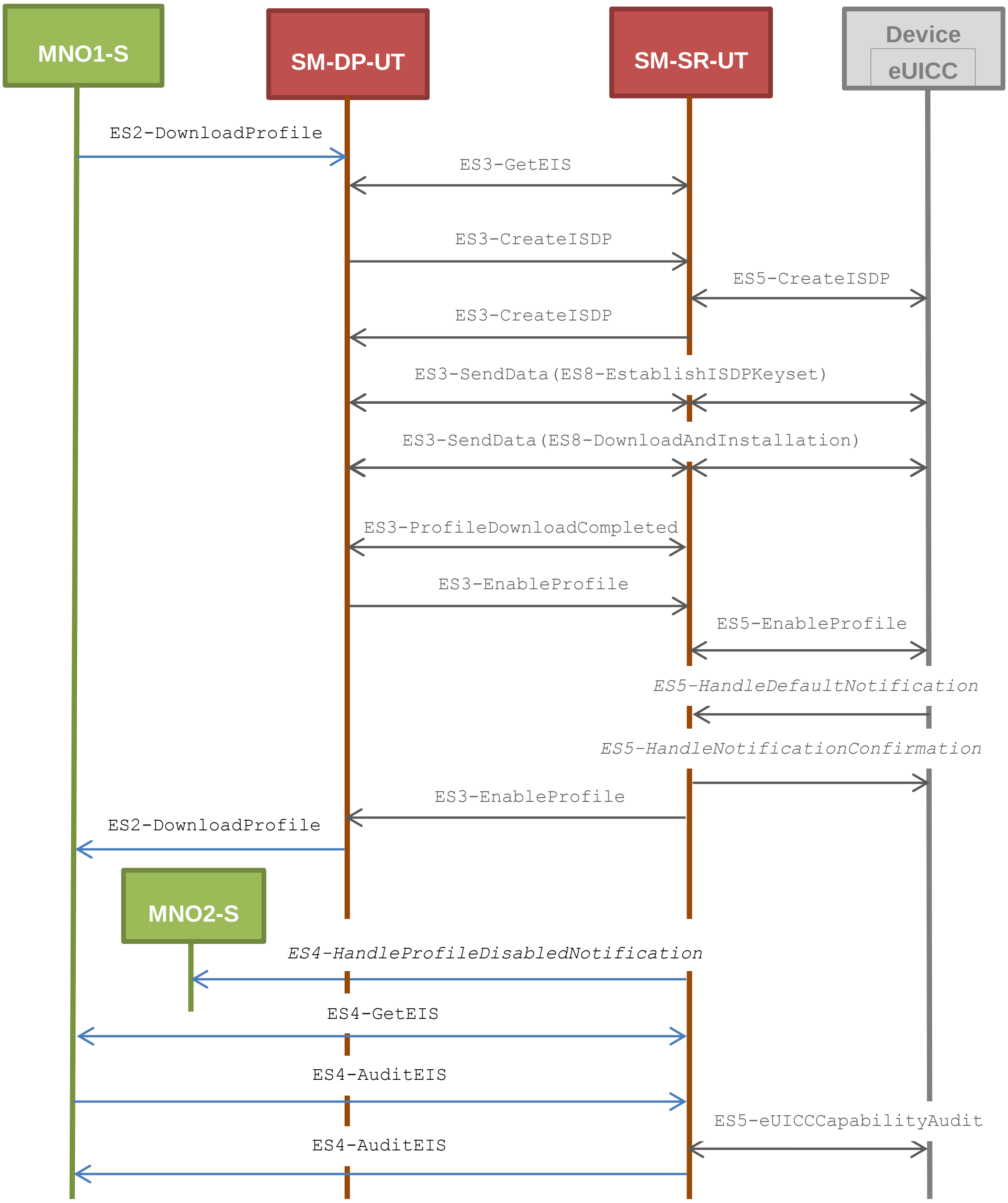
- PROC_REQ1, PROC_REQ2, PROC_REQ3, PROC_REQ7, PROC_REQ20
- PM_REQ4, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17, PM_REQ18, PM_REQ22, PM_REQ25
- PF_REQ2, PF_REQ3, PF_REQ4, PF_REQ7, PF_REQ18, PF_REQ27
- EUICC_REQ27, EUICC_REQ29

Initial Conditions

- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by `#NEW_ICCID`)
  - o It means that the SM-SR-UT knows the Connectivity Parameters of the MNO’s network related to the new Profile (i.e. `#MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD`)
- The Profile identified by `#NEW_ICCID` shall be adapted to contain correct Connectivity Parameters (i.e. `#MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD`)
- `#MNO2_S_ID` well known to the SM-SR-UT
- `#MNO2_S_ACCESSPOINT` well known to the SM-SR-UT
  - o A direct connection exists between the MNO2-S and the SM-SR-UT
- The SMS mode is the default way (priority order 1) to send the notification

5.3.2.2.2.1 Test Sequence N°1 - Nominal Case

Test Environment



Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_RPS
- POL1 and POL2 of the Profile identified by #ICCID do not contain any rules
  - Disabling of the Profile is allowed
  - “Profile deletion is mandatory when it is disabled” is not required
  - POL2 may be adapted on the #EIS_RPS
  - POL1 may be adapted in the eUICC

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DownloadProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS,`
		`#EP_TRUE_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the `ES2-DownloadProfile` response	1- The `Status` is equal to `#SUCCESS` 2- The `ICCID` returned is equal to `#NEW_ICCID_RPS`	PROC_REQ1, PROC_REQ2, PROC_REQ3, PROC_REQ7, PROC_REQ20, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17, PM_REQ18, PF_REQ2, PF_REQ3, PF_REQ4, PF_REQ18, EUICC_REQ27, EUICC_REQ29
4	SM-SR-UT → MNO2-S	Send the `ES4-HandleProfileDisabledNotification` notification	1- The `EID` parameter is equal to `#EID_RPS` 2- The ICCID is equal to `#ICCID_RPS` 3- The completion timestamp is present	PF_REQ27, PROC_REQ7

5	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS})`		

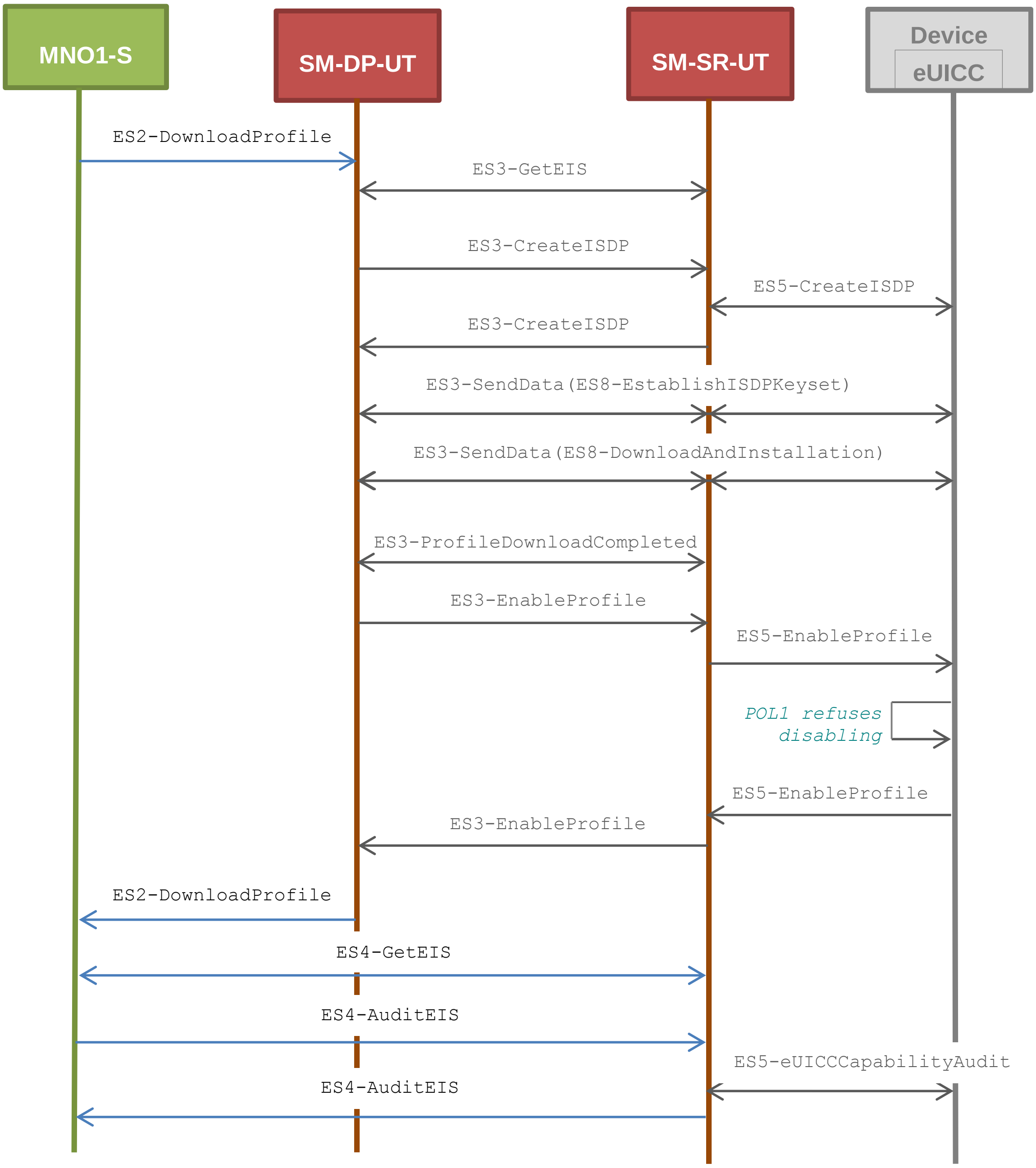
Step	Direction	Sequence / Description	Expected result	REQ
6	SM-DP-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` returned contains the new Profile information (i.e. identified by `#NEW_ICCID`) 3- The new Profile information has a state equal to Enabled 4- The new Profile information has the SM-DP identifier set to `#SM-DP-ID` 5- The new Profile information has an ISD-P RID equal to `#ISD_P_RID` 6- The new Profile information has an ISD-P PIX that starts with `#ISD_P_PIX_PREFIX` 7- The new Profile information has a MNO-ID equal to `#MNO1_S_ID` 8- The new Profile information has the Subscription Address equal to `#NEW_MSISDN`	PM_REQ4, PM_REQ22

7	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 6 except that: a. the `free memory` of the new Profile is updated (i.e. lower than that received in step 6) b. the `remaining memory` and the `available memory for Profiles` are updated (i.e. lower than that received in step 6)	PM_REQ25, PF_REQ2, PF_REQ7

5.3.2.2.2.2 Test Sequence N°2 – Error Case: POL1 Refuses Profile Disabling

Test Environment



Initial Conditions

- The eUICC identified by `#EID` has been provisioned on the SM-SR-UT using the `#EIS_RPS`
- POL1 of the Profile identified by `#ICCID` contains the rule “Disabling not Allowed”
- POL2 of the Profile identified by `#ICCID` does not contain any rules

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DownloadProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS,`
		`#EP_TRUE_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the `ES2-DownloadProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_ISDR` 3- The `Reason code` is equal to `#RC_EXECUTION_ERROR` 4- The `euiccResponseData` is present and contains the POR generated by the eUICC (i.e. SW='69E1')	PROC_REQ1, PROC_REQ2, PROC_REQ3, PROC_REQ8, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ12, PM_REQ17, PM_REQ18, PF_REQ2, PF_REQ3, PF_REQ4, PF_REQ18, EUICC_REQ27, EUICC_REQ29

4	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS})`		

5	SM-DP-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by `#NEW_ICCID` c. the Profile identified by `#ICCID` is not present d. the Profile identified by `#NEW_ICCID` is Disabled	PM_REQ4, PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5 except that: a. the `free memory` of the new Profile is updated (i.e. lower than that received in step 5) b. the `remaining memory` and the `available memory for Profiles` are updated (i.e. lower than that received in step 5)	PM_REQ25, PF_REQ2, PF_REQ7

5.3.3 Profile Enabling Process

5.3.3.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ2, PF_REQ4, PF_REQ6, PF_REQ7, PF_REQ12, PF_REQ15, PF_REQ17, PF_REQ18, PF_REQ21, PF_REQ23, PF_REQ24, PF_REQ27, PF_REQ29
- PROC_REQ5, PROC_REQ6, PROC_REQ7, PROC_REQ8, PROC_REQ20
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

5.3.3.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID` well known to the SM-SR-UT
- `#MNO1_S_ACCESSPOINT` well known to the SM-SR-UT
    - A direct connection exists between the MNO1-S and the SM-SR-UT
- `#MNO2_S_ID` well known to the SM-SR-UT
- The Profile identified by `#ICCID` is owned by MNO2-S and is in Enabled state
- The Profile identified by `#NEW_ICCID` is owned by MNO1-S and is in Disabled state
    - To download the new Profile (e.g. `#PROFILE_PACKAGE`), the test sequence defined in section 5.3.2.2.1.1 may be used
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by `#ICCID`)

  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the default Enabled Profile (i.e. `#MNO2_CON_NAN, #MNO2_CON_LOGIN, #MNO2_CON_PWD`)
- The eUICC identified by `#EID` has been provisioned on the SM-SR-UT using the `#EIS_RPS`
- The SMS mode is the default way (priority order 1) to send the notification

Note: To facilitate the execution of the test cases, the default Enabled Profile and the Profile to be Enabled may use the same Connectivity Parameters (i.e. the two Profiles are linked to the same MNO's network).

5.3.3.2.1 TC.PROC.PE.1: ProfileEnablingByMNO

Test Purpose

To ensure a Profile can be Enabled by the SM-SR when the MNO requests it, different Policy Rules are used and an error case, using bad Connectivity Parameters, is described to make sure that the roll-back process is well implemented. In case of a successful enabling process, an audit request is sent to the SM-SR to make sure that the Profile has been Enabled.

Referenced Requirements

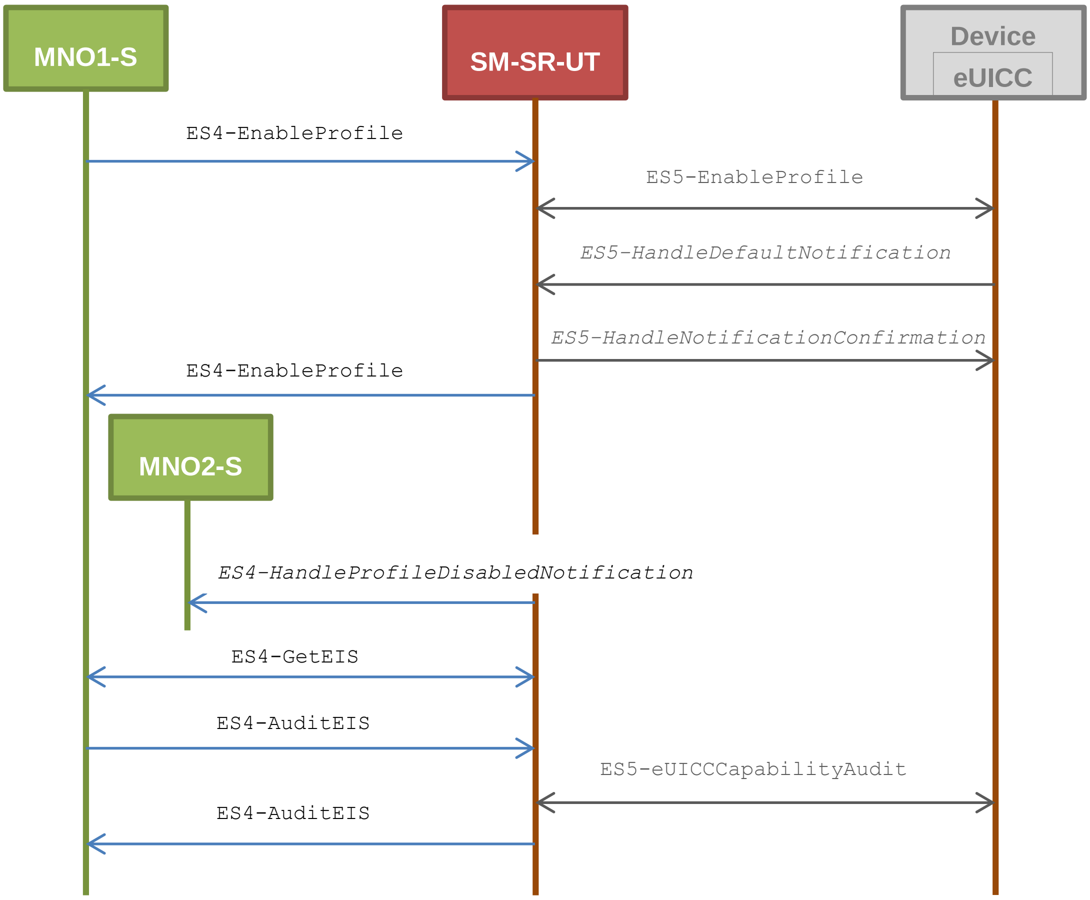
- PF_REQ2, PF_REQ4, PF_REQ6, PF_REQ7, PF_REQ24, PF_REQ27, PF_REQ29
- PROC_REQ5, PROC_REQ6, PROC_REQ20
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

Initial Conditions

- `#MNO2_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO2-S and the SM-SR-UT

5.3.3.2.1.1 Test Sequence N°1 – Nominal Case: Empty POL1 and POL2

Test Environment



Initial Conditions

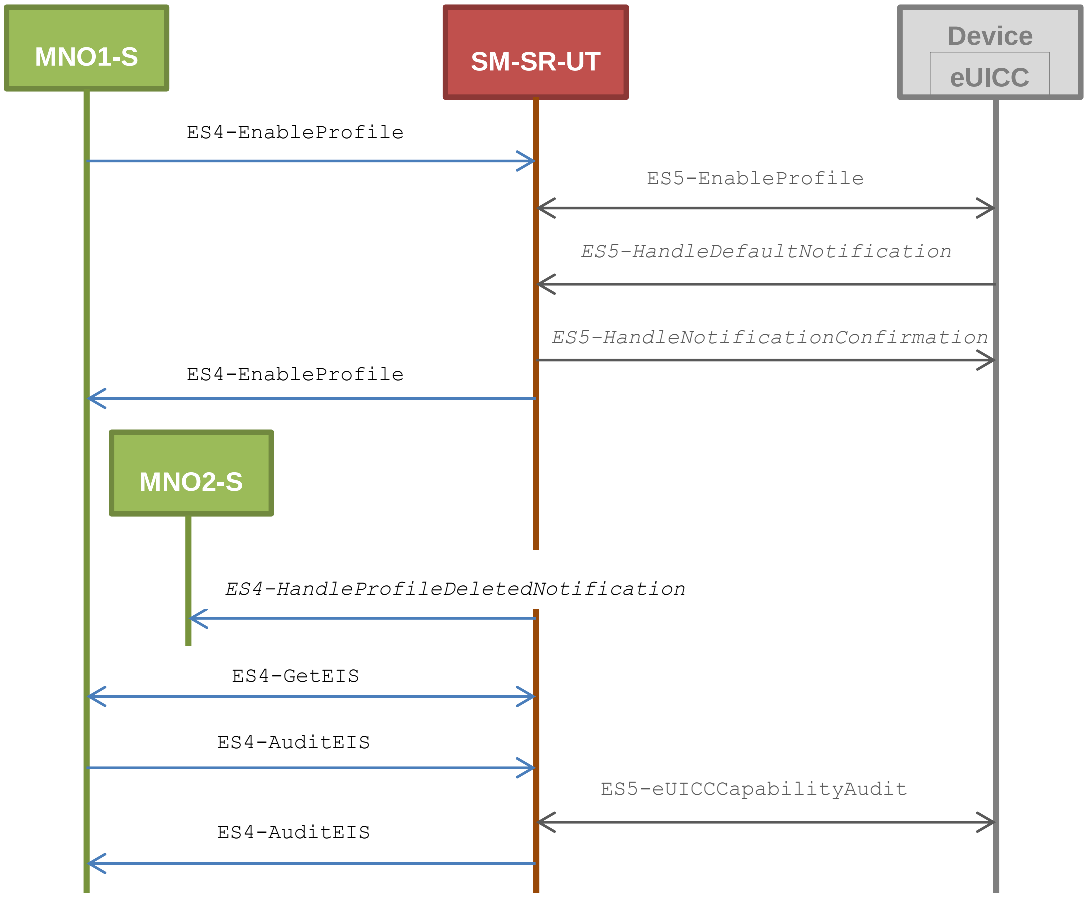
- The Profile downloaded, identified by #NEW_ICCID, shall be adapted to contain correct Connectivity Parameters (i.e. #MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by #NEW_ICCID)
    - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the new Profile (i.e. #MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD)
- POL1 and POL2 of the Profile identified by #ICCID do not contain any rules and may need to be adapted on the #EIS_RPS and in the eUICC as follow:
    - Disabling of the Profile is allowed
    - "Profile deletion is mandatory when it is disabled" is not set

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	The Status is equal to #SUCCESS	PF_REQ2, PF_REQ4, PF_REQ24, PROC_REQ5, PROC_REQ20, EUICC_REQ27, EUICC_REQ29
4	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileDisabledNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ27, PROC_REQ5
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by #NEW_ICCID c. the Profile identified by #ICCID is not present d. the Profile identified by #NEW_ICCID is Enabled	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
9	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 6	PF_REQ2, PF_REQ7, PM_REQ26

5.3.3.2.1.2 Test Sequence N°2 - Nominal Case: POL1 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment



Initial Conditions

- The Profile downloaded, identified by `#NEW_ICCID`, shall be adapted to contain correct Connectivity Parameters (i.e. `#MNO1_CON_NAN`, `#MNO1_CON_LOGIN`, `#MNO1_CON_PWD`)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by `#NEW_ICCID`)
    - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the new Profile (i.e. `#MNO1_CON_NAN`, `#MNO1_CON_LOGIN`, `#MNO1_CON_PWD`)
- POL1 of the Profile identified by `#ICCID` contains only the rule "Delete when Disabling" (POL1 may need to be adapted on the eUICC)

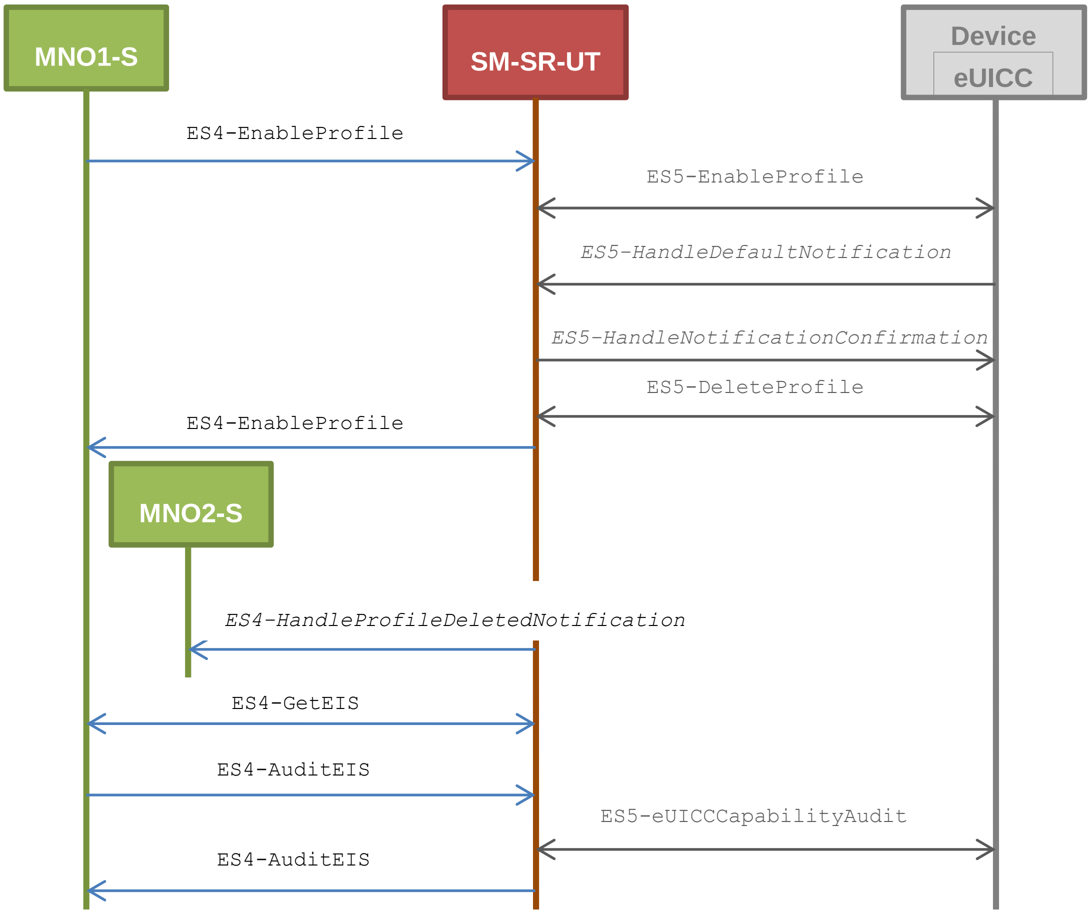
- POL2 of the Profile identified by #ICCID does not contain any rules (POL2 may need to be adapted on the #EIS_RPS)
  - Disabling of the Profile is allowed
  - “Profile deletion is mandatory when it is disabled” is not set

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_POL1	PF_REQ2, PF_REQ4, PF_REQ24, PROC_REQ5, PROC_REQ20, EUICC_REQ27, EUICC_REQ29
4	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileDeletedNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ29, PROC_REQ5
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by #NEW_ICCID c. the Profile identified by #ICCID is not present d. the Profile identified by #NEW_ICCID is Enabled	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		

Step	Direction	Sequence / Description	Expected result	REQ
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 6 except that: a. the `remaining memory` and the `available memory for Profiles` are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.3.2.1.3 Test Sequence N°3 - Nominal Case: POL2 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment



Initial Conditions

- The Profile downloaded, identified by `#NEW_ICCID`, shall be adapted to contain correct Connectivity Parameters (i.e. `#MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD`)

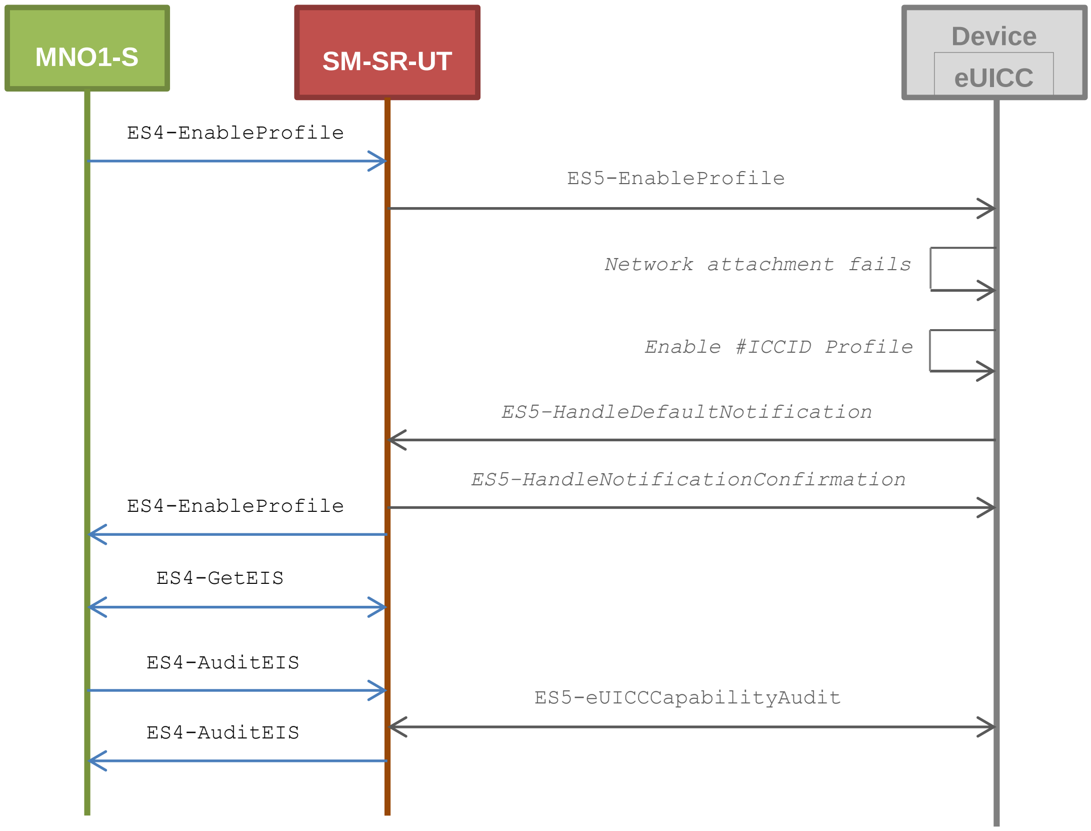
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by #NEW_ICCID)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the new Profile (i.e. #MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD)
- POL1 of the Profile identified by #ICCID does not contain any rules (POL1 may need to be adapted on the eUICC)
  - Disabling of the Profile is allowed
  - "Profile deletion is mandatory when it is disabled" is not set
- POL2 of the Profile identified by #ICCID contains only the rule "Profile deletion is mandatory when it is disabled" (POL2 may need to be adapted on the #EIS_RPS)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_POL2	PF_REQ2, PF_REQ4, PF_REQ6, PF_REQ24, PROC_REQ5, PROC_REQ20, EUICC_REQ27, EUICC_REQ29
4	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileDeletedNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ29, PROC_REQ5
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		

Step	Direction	Sequence / Description	Expected result	REQ
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by #NEW_ICCID c. the Profile identified by #ICCID is not present d. the Profile identified by #NEW_ICCID is Enabled	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 6 except that: a. the remaining memory and the available memory for Profiles are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.3.2.1.4 Test Sequence N°4 – Error Case: Bad Connectivity Parameters

Test Environment



Initial Conditions

- The Profile downloaded, identified by #NEW_ICCID, shall be adapted to contain inconsistent Connectivity Parameters (e.g. #NAN_VALUE, #LOGIN, #PWD)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EUICC 3- The Reason code is equal to #RC_INACCESSIBLE	PF_REQ2, PF_REQ4, PF_REQ24, PROC_REQ6, PROC_REQ20, EUICC_REQ27, EUICC_REQ29

Step	Direction	Sequence / Description	Expected result	REQ

4	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by `#NEW_ICCID` c. the Profile identified by `#ICCID` is not present d. the Profile identified by `#NEW_ICCID` is Disabled	PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5	PF_REQ2, PF_REQ7, PM_REQ26

5.3.3.2.2 TC.PROC.PE.2: ProfileEnablingViaSMDP

Test Purpose

To ensure a Profile can be Enabled by the SM-DP and the SM-SR when the MNO requests it, different Policy Rules are used and an error case, using bad Connectivity Parameters, is described to make sure that the roll-back process is well implemented. In case of successful enabling process, an audit request is sent to the SM-SR to make sure that the Profile has been Enabled.

Referenced Requirements

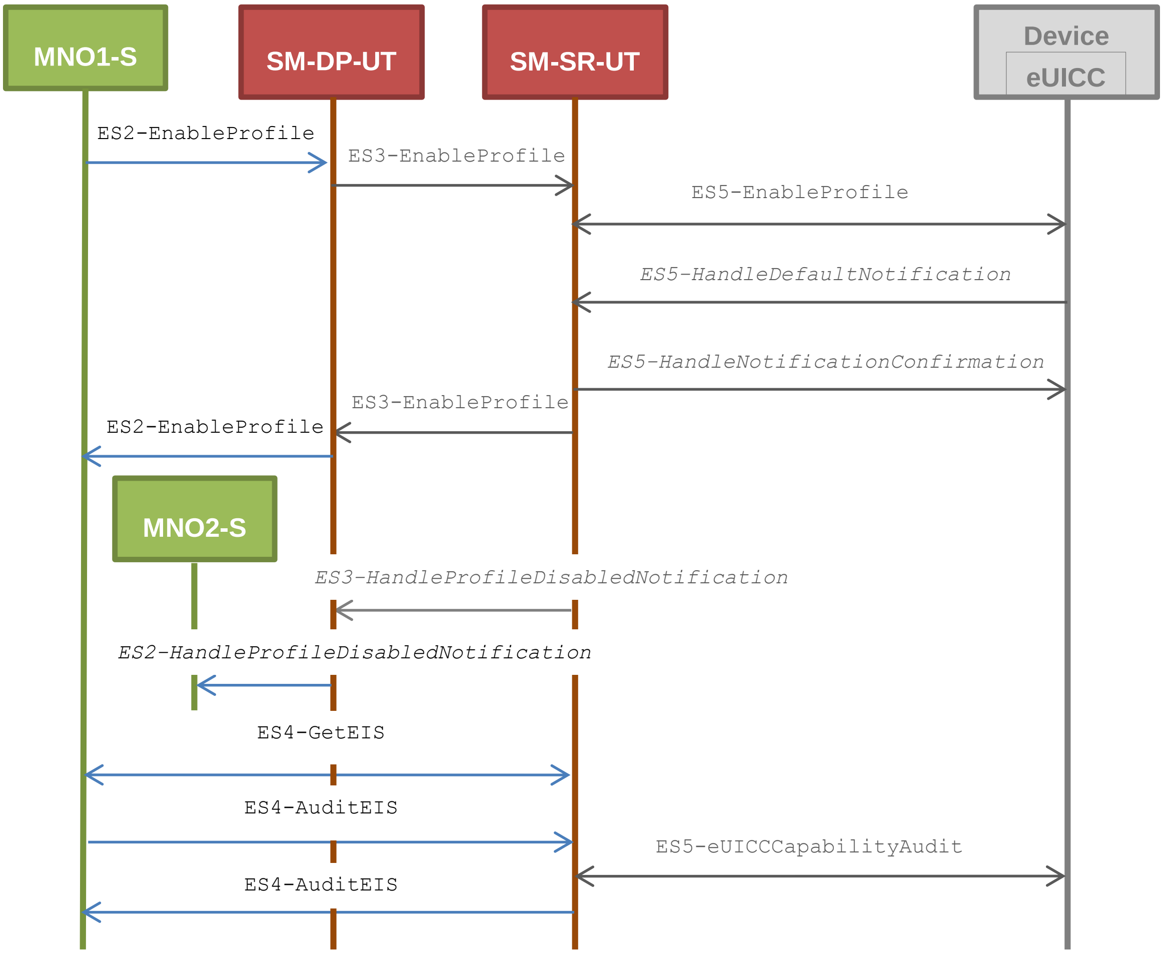
- PF_REQ2, PF_REQ4, PF_REQ6, PF_REQ7, PF_REQ12, PF_REQ15, PF_REQ17, PF_REQ18, PF_REQ21, PF_REQ23
- PROC_REQ7, PROC_REQ8, PROC_REQ20
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

Initial Conditions

- #MNO2_S_ACCESSPOINT is unknown to the SM-SR-UT
- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #MNO2_S_ID and #MNO2_S_ACCESSPOINT well known to the SM-DP-UT
- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS
- #SM_SR_ID and #SM_SR_ACCESSPOINT well known to the SM-DP-UT
- #SM_DP_ID and #SM_DP_ACCESSPOINT well known to the SM-SR-UT
- The Profile identified by #ICCID is linked to the SM-DP identified by #SM_DP_ID (the #EIS_RPS may need to be adapted on the SM-SR-UT)

5.3.3.2.2.1 Test Sequence N°1 – Nominal Case: Empty POL1 and POL2

Test Environment



Initial Conditions

- The Profile downloaded, identified by #NEW_ICCID, shall be adapted to contain correct Connectivity Parameters (i.e. #MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by #NEW_ICCID)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the new Profile (i.e. #MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD)

- POL1 and POL2 of the Profile identified by #ICCID do not contain any rules and may need to be adapted on the #EIS_RPS and in the eUICC as follow:
    - Disabling of the Profile is allowed
    - “Profile deletion is mandatory when it is disabled” is not set

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-EnableProfile, #EID_RPS, {SM_SR_ID_RPS}, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the ES2-EnableProfile response	The Status is equal to #SUCCESS	PF_REQ2, PF_REQ4, PF_REQ12, PF_REQ18, PF_REQ21, PROC_REQ7, PROC_REQ20, EUICC_REQ27, EUICC_REQ29
4	SM-DP-UT → MNO2-S	Send the ES2-HandleProfileDisabledNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ15, PROC_REQ7
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by #NEW_ICCID c. the Profile identified by #ICCID is not present d. the Profile identified by #NEW_ICCID is Enabled	PM_REQ22

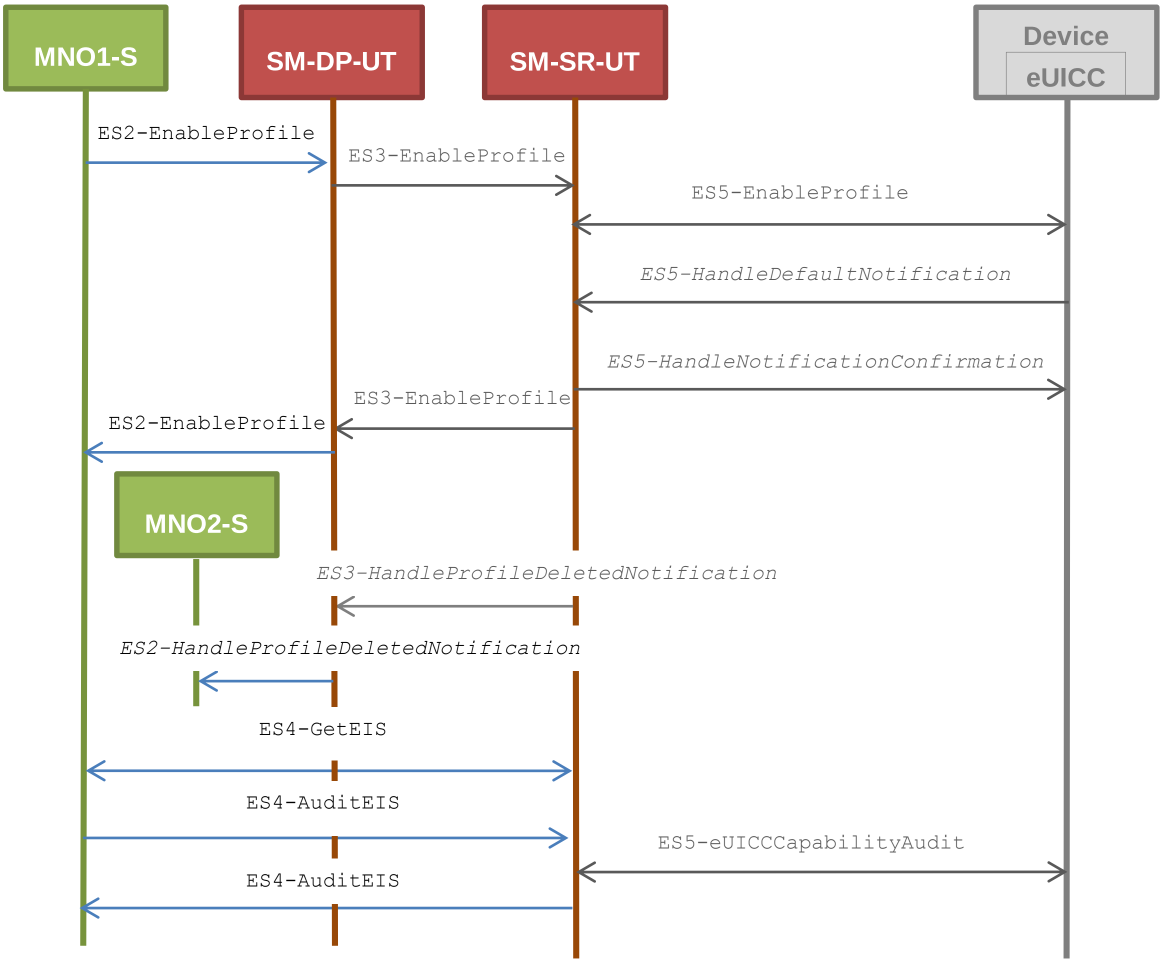
Step	Direction	Sequence / Description	Expected result	REQ

7	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 6	PF_REQ2, PF_REQ7, PM_REQ26

5.3.3.2.2.2 Test Sequence N°2 – Nominal Case: POL1 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment



Initial Conditions

- The Profile downloaded, identified by `#NEW_ICCID`, shall be adapted to contain correct Connectivity Parameters (i.e. `#MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD`)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by `#NEW_ICCID`)

  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the new Profile (i.e. `#MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD`)
- POL1 of the Profile identified by `#ICCID` contains only the rule "Profile deletion is mandatory when it is disabled" (POL1 may need to be adapted on the eUICC)
- POL2 of the Profile identified by `#ICCID` does not contain any rules (POL2 may need to be adapted on the `#EIS_RPS)`
  - Disabling of the Profile is allowed
  - "Profile deletion is mandatory when it is disabled" is not set

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-EnableProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the `ES2-EnableProfile` response	1- The `Status` is equal to `#WARNING` 2- The `Subject code` is equal to `#SC_POL1`	PF_REQ2, PF_REQ4, PF_REQ12, PF_REQ18, PF_REQ23, PROC_REQ7, PROC_REQ20, EUICC_REQ27, EUICC_REQ29
4	SM-DP-UT → MNO2-S	Send the `ES2-HandleProfileDeletedNotification` notification	1- The `EID` parameter is equal to `#EID_RPS` 2- The ICCID is equal to `#ICCID_RPS` 3- The completion timestamp is present	PF_REQ17, PROC_REQ7

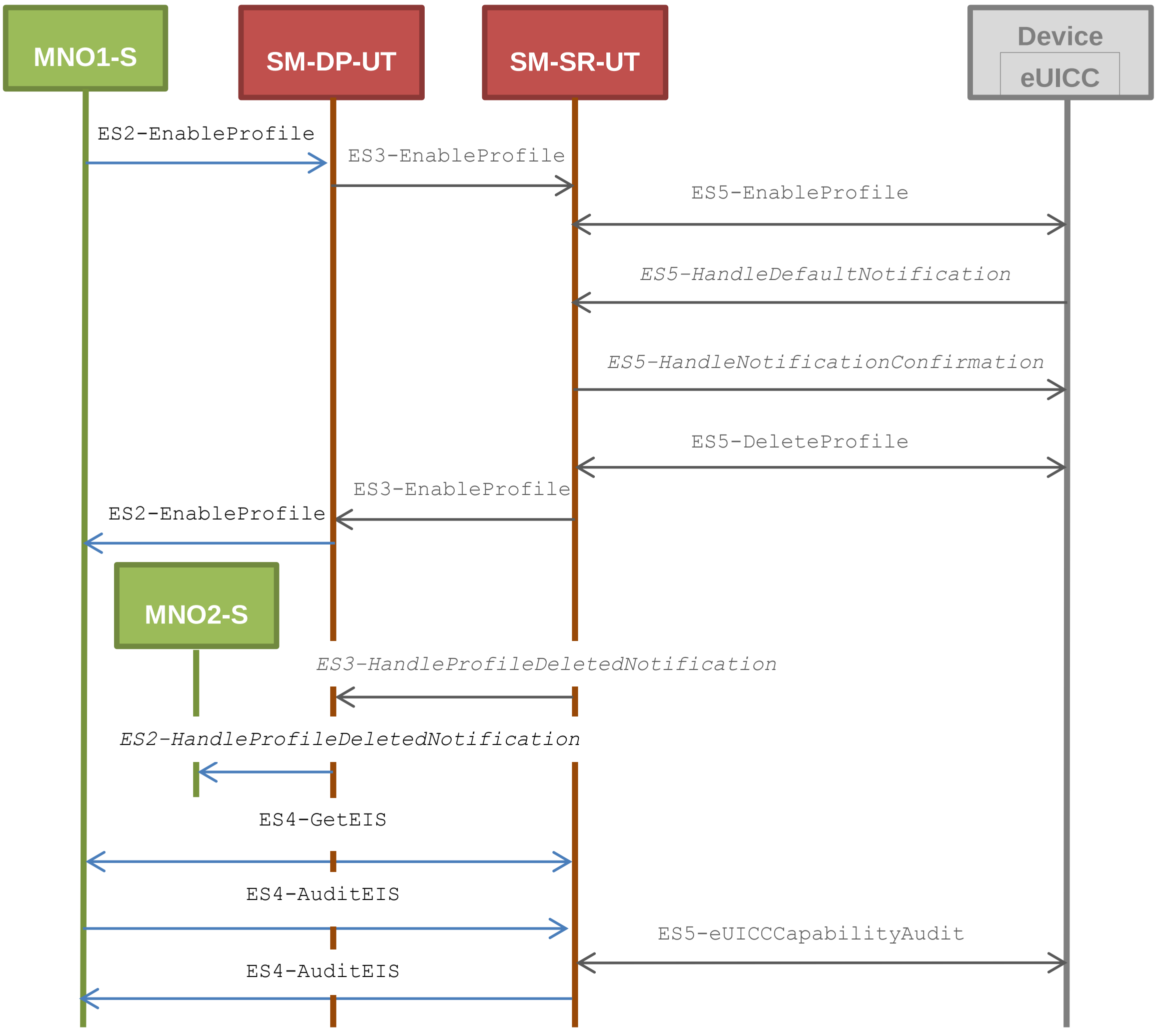
5	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by #NEW_ICCID c. the Profile identified by #ICCID is not present d. the Profile identified by #NEW_ICCID is Enabled	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 6 except that: a. the remaining memory and the available memory for Profiles are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.3.2.2.3 Test Sequence N°3 – Nominal Case: POL2 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment

Initial Conditions



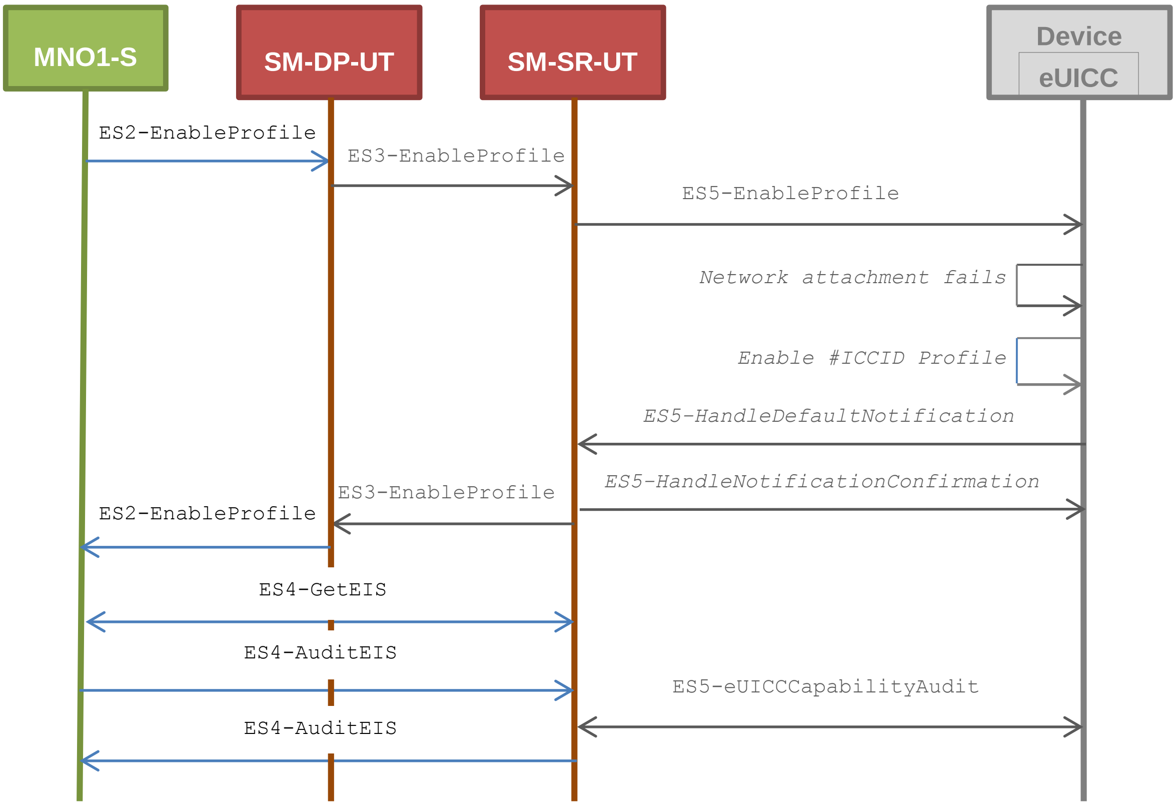
- The Profile downloaded, identified by #NEW_ICCID, shall be adapted to contain correct Connectivity Parameters (i.e. #MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by #NEW_ICCID)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the new Profile (i.e. #MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD)
- POL1 of the Profile identified by #ICCID does not contain any rules (POL1 may need to be adapted on the eUICC)
  - Disabling of the Profile is allowed
  - "Profile deletion is mandatory when it is disabled" is not set
- POL2 of the Profile identified by #ICCID contains only the rule "Profile deletion is mandatory when it is disabled" (POL2 may need to be adapted on the #EIS_RPS)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-EnableProfile, #EID_RPS, {SM_SR_ID_RPS}, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the ES2-EnableProfile response	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_POL1	PF_REQ2, PF_REQ4, PF_REQ6, PF_REQ12, PF_REQ18, PF_REQ23, PROC_REQ7, PROC_REQ20, EUICC_REQ27, EUICC_REQ29
4	SM-DP-UT → MNO2-S	Send the ES2-HandleProfileDeletedNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ17, PROC_REQ7
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by #NEW_ICCID c. the Profile identified by #ICCID is not present d. the Profile identified by #NEW_ICCID is Enabled	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
9	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 6 except that: a. the remaining memory and the available memory for Profiles are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.3.2.2.4 Test Sequence N°4 – Error Case: Bad Connectivity Parameters

Test Environment



Initial Conditions

- The Profile downloaded, identified by #NEW_ICCID, shall be adapted to contain inconsistent Connectivity Parameters (e.g. #NAN_VALUE, #LOGIN, #PWD)

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-EnableProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the `ES2-EnableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_EUICC` 3- The `Reason code` is equal to `#RC_INACCESSIBLE`	PF_REQ2, PF_REQ4, PF_REQ12, PF_REQ18, PROC_REQ8, PROC_REQ20, EUICC_REQ27, EUICC_REQ29

4	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO1-S are present among which that identified by `#NEW_ICCID` c. the Profile identified by `#ICCID` is not present d. the Profile identified by `#NEW_ICCID` is Disabled	PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4 Profile Disabling Process

5.3.4.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ2, PF_REQ5, PF_REQ6, PF_REQ7, PF_REQ13, PF_REQ16, PF_REQ19, PF_REQ22, PF_REQ25, PF_REQ28
- PROC_REQ9, PROC_REQ10, PROC_REQ20, PROC_REQ22
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

5.3.4.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID` well known to the SM-SR-UT
- `#MNO1_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT
- `#MNO2_S_ID` well known to the SM-SR-UT
- The Profile identified by `#ICCID` is owned by MNO2-S, is in Disabled state and has the Fall-back Attribute
  - The Profile may need to be adapted to have the Fall-back Attribute Set
- The Profile identified by `#NEW_ICCID` is owned by MNO1-S and is in Enabled state
  - To Enable the new Profile (e.g. `#PROFILE_PACKAGE`), the test sequence defined in section 5.3.3.2.1.1 may be used
- The SM-SR-UT is able to communicate with the network linked to the Enabled Profile (identified by `#NEW_ICCID`)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO’s network related to the Enabled Profile (i.e. `#MNO1_CON_NAN, #MNO1_CON_LOGIN, #MNO1_CON_PWD`)
- The SM-SR-UT is able to communicate with the network linked to the Profile with the Fall-back Attribute (identified by `#ICCID`)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO’s network related to the Profile with the Fall-back attribute (i.e. `#MNO2_CON_NAN, #MNO2_CON_LOGIN, #MNO2_CON_PWD`)
- The eUICC identified by `#EID` has been provisioned on the SM-SR-UT using the `#EIS_RPS`
- The SMS mode is the default way (priority order 1) to send the notification

Note: To facilitate the execution of the test cases, the Profile with the Fall-back Attribute and the Profile to be Disabled may use the same Connectivity Parameters (i.e. the two Profiles are linked to the same MNO’s network).

5.3.4.2.1 TC.PROC.DIS.1: ProfileDisablingByMNO

Test Purpose

To ensure a Profile can be Disabled by the SM-SR when the MNO requests it, different Policy Rules are used. After the Profile disabling, an audit request is sent to the SM-SR to make sure that the Profile has been Disabled. Some error cases are also described:

- *the Profile with the Fall-back Attribute Set contains bad Connectivity Parameters*
- *the Profile to be Disabled contains the POL1 "Disabling not Allowed"*

Referenced Requirements

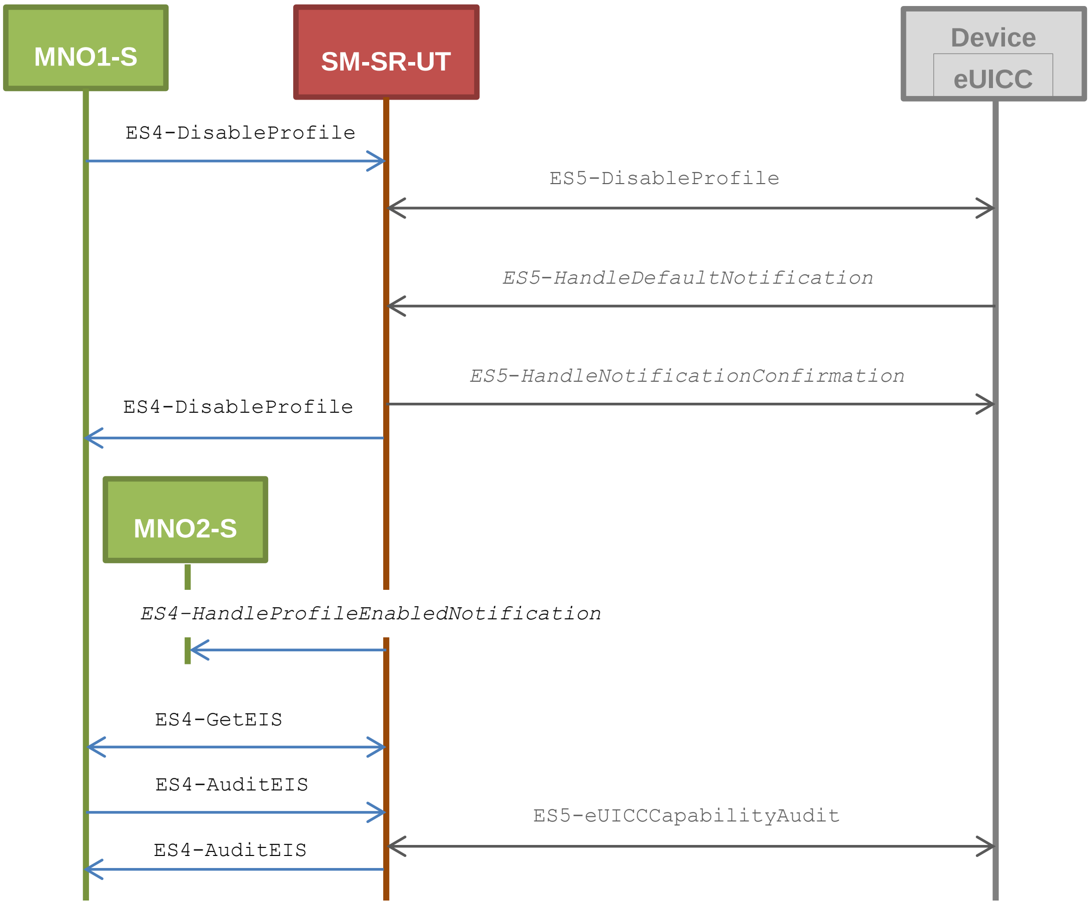
- PF_REQ2, PF_REQ5, PF_REQ6, PF_REQ7, PF_REQ25, PF_REQ28
- PROC_REQ9, PROC_REQ20, PROC_REQ22
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

Initial Conditions

- `#MNO2_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO2-S and the SM-SR-UT

5.3.4.2.1.1 Test Sequence N°1 - Nominal Case: Empty POL1 and POL2

Test Environment



Initial Conditions

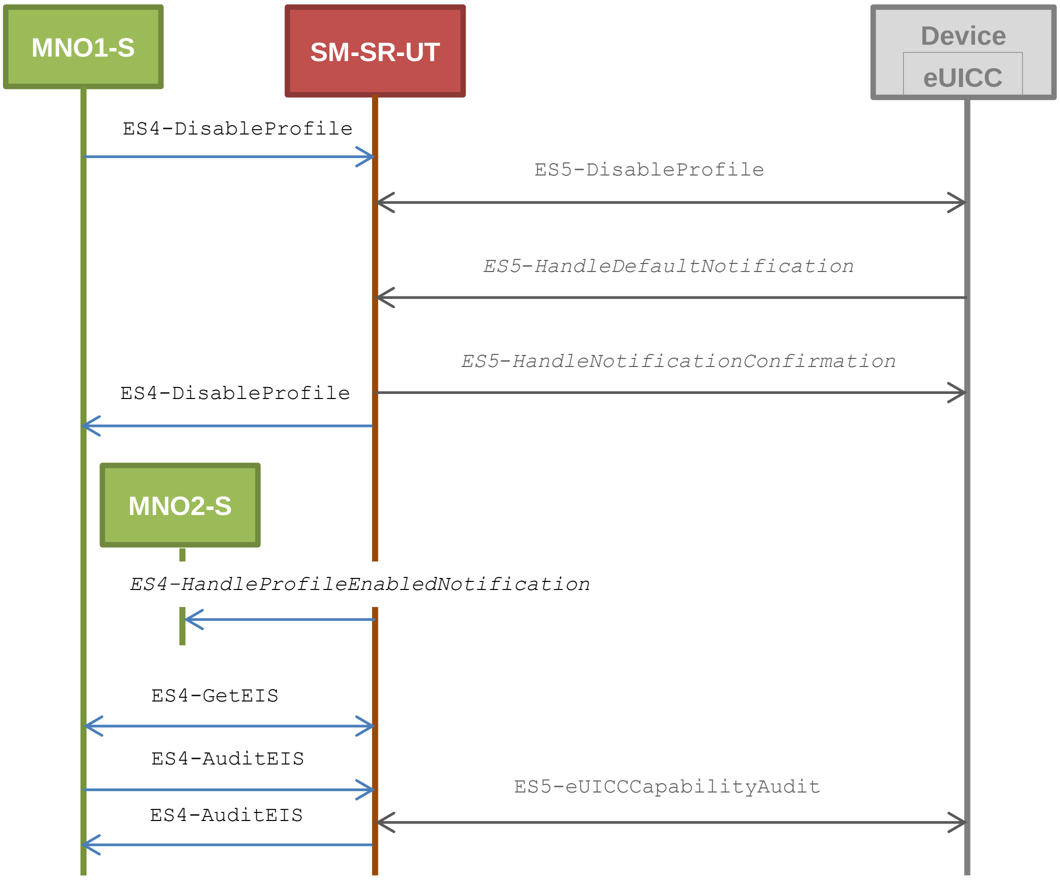
- POL1 and POL2 of the Profile identified by #NEW_ICCID do not contain any rules
    - Disabling of the Profile is allowed
    - “Profile deletion is mandatory when it is disabled” is not set

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DisableProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-DisableProfile response	The Status is equal to #SUCCESS	PF_REQ2, PF_REQ5, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9, PROC_REQ20, PROC_REQ22
4	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileEnabledNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ28, PROC_REQ9
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. the Profile identified by #NEW_ICCID is Disabled	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
9	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 6	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.1.2 Test Sequence N°2 - Nominal Case: POL1 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment



Initial Conditions

- POL1 of the Profile identified by `#NEW_ICCID` contains the rule "Profile deletion is mandatory when it is disabled"
- POL2 of the Profile identified by `#NEW_ICCID` allows disabling

Step	Direction	Sequence / Description	Expected result	REQ

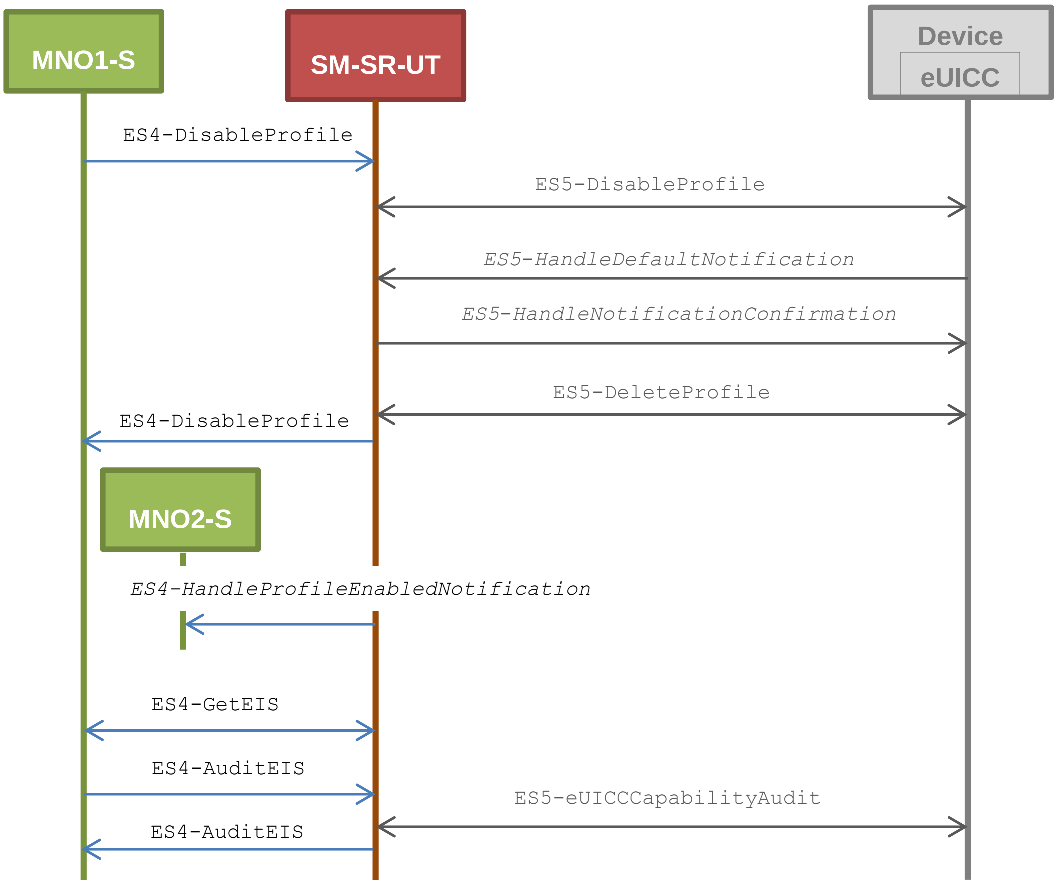
1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-UT → MNO1-S	Send the `ES4-DisableProfile` response	1- The `Status` is equal to `#WARNING` 2- The `Subject code` is equal to `#SC_POL1`	PF_REQ2, PF_REQ5, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9, PROC_REQ20, PROC_REQ22
4	SM-SR-UT → MNO2-S	Send the `ES4-HandleProfileEnabledNotification` notification	1- The `EID` parameter is equal to `#EID_RPS` 2- The ICCID is equal to `#ICCID_RPS` 3- The completion timestamp is present	PF_REQ28, PROC_REQ9
5	MNO1-S → SM-SR-UT	`SEND_REQ( ES4-GetEIS, #EID_RPS)`		
6	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. the Profile identified by `#NEW_ICCID` is not present	PM_REQ22
7	MNO1-S → SM-SR-UT	`SEND_REQ( ES4-AuditEIS, #EID_RPS)`		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 6 except that: a. the `remaining memory` and the `available memory for Profiles` are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.1.3 Test Sequence N°3 - Nominal Case: POL2 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment



Initial Conditions

- POL1 of the Profile identified by `#NEW_ICCID` does not contain any rules
  - Disabling of the Profile is allowed
  - "Profile deletion is mandatory when it is disabled" is not set
- POL2 of the Profile identified by `#NEW_ICCID` contains the rule "Profile deletion is mandatory when it is disabled"

Step	Direction	Sequence / Description	Expected result	REQ

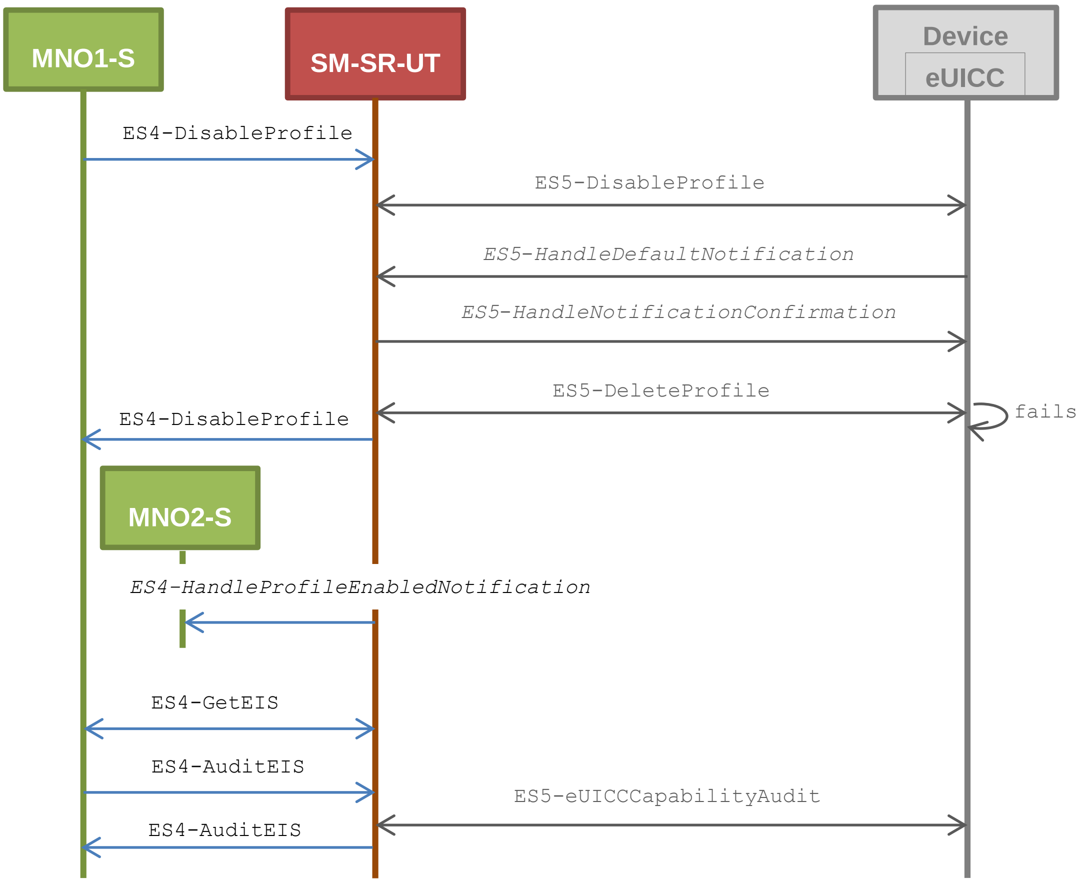
1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-UT → MNO1-S	Send the ES4-DisableProfile response	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_POL2	PF_REQ2, PF_REQ5, PF_REQ6, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9, PROC_REQ20, PROC_REQ22
4	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileEnabledNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ28, PROC_REQ9
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. the Profile identified by #NEW_ICCID is not present	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 6 except that: a. the remaining memory and the available memory for Profiles are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.1.4 Test Sequence N°4 - Nominal Case: POL1 with "Deletion not Allowed" and POL2 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment



Initial Conditions

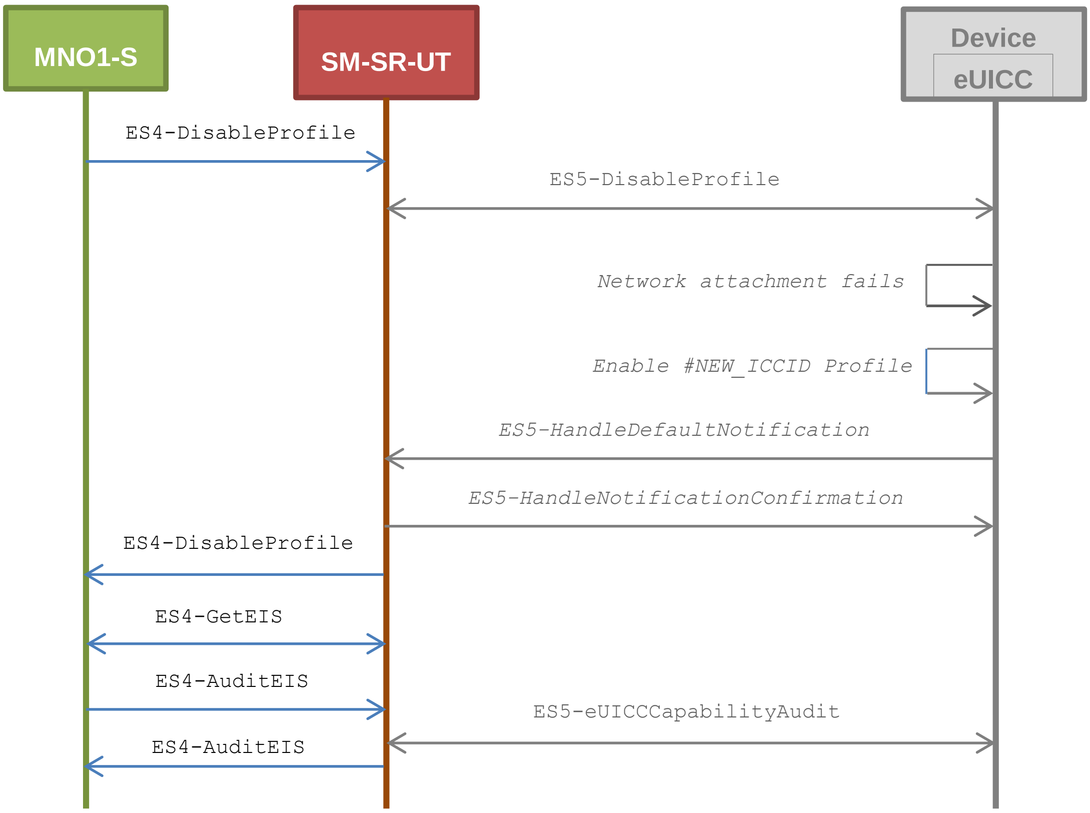
- POL1 of the Profile identified by #NEW_ICCID forbids deletion
  - Disabling of the Profile is allowed
  - Deletion of the Profile is not allowed
- POL2 of the Profile identified by #NEW_ICCID contains the rule "Profile deletion is mandatory when it is disabled"

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DisableProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-UT → MNO1-S	Send the ES4-DisableProfile response	The Status is equal to #SUCCESS (see Note1)	PF_REQ2, PF_REQ5, PF_REQ6, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9, PROC_REQ20, PROC_REQ22
4	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileEnabledNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ28, PROC_REQ9
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. the Profile identified by #NEW_ICCID is Disabled	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 6	PF_REQ2, PF_REQ7, PM_REQ26
Note 1: Even if a DELETE command is sent by the SM-SR and fails (because of POL1), the status of the disabling process shall be successful.				

5.3.4.2.1.5 Test Sequence N°5 - Error Case: Bad Connectivity Parameters

Test Environment



Initial Conditions

- The Profile, identified by `#ICCID`, shall be adapted to contain inconsistent Connectivity Parameters (e.g. `#NAN_VALUE, #LOGIN, #PWD`)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DisableProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-DisableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EUICC 3- The Reason code is equal to #RC_INACCESSIBLE	PF_REQ2, PF_REQ5, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9, PROC_REQ20, PROC_REQ22
4	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		

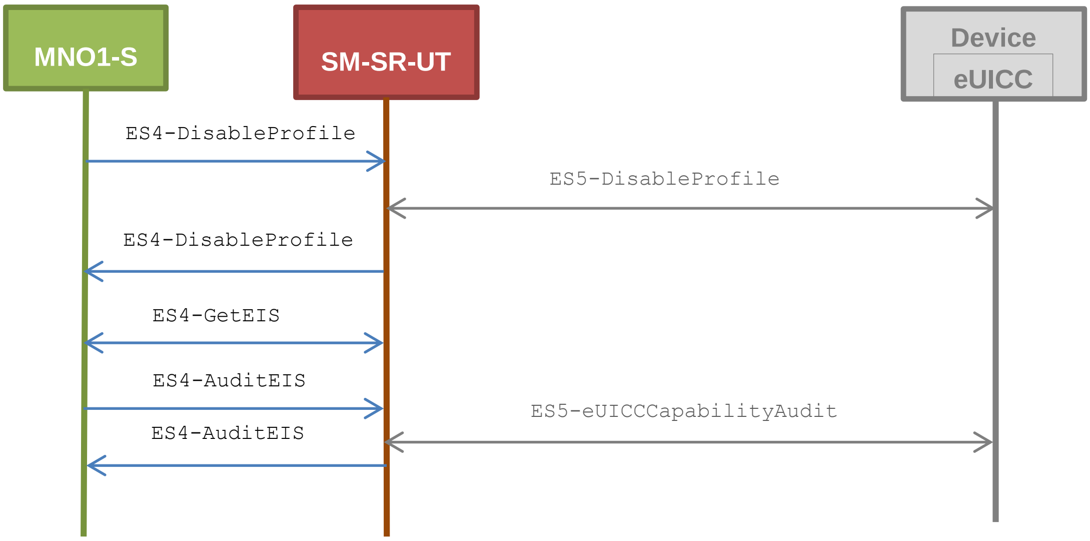
Step	Direction	Sequence / Description	Expected result	REQ
5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. the Profile identified by `#NEW_ICCID` is Enabled	PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.1.6 Test Sequence N°6 - Error Case: POL1 with "Disabling not Allowed"

Test Environment



Initial Conditions

- POL1 of the Profile identified by `#NEW_ICCID` contains the rule "Disabling not Allowed"
- POL2 of the Profile identified by `#NEW_ICCID` does not contain any rules
  - Disabling of the Profile is allowed

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the `ES4-DisableProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_POL1` 3- The `Reason code` is equal to `#RC_REFUSED` 4- The `euiccResponseData` is present and contains the POR generated by the eUICC (i.e. SW='69E1')	PF_REQ2, PF_REQ5, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9

4	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. the Profile identified by `#NEW_ICCID` is Enabled	PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.2 TC.PROC.DIS.2: ProfileDisablingViaSMDP

Test Purpose

To ensure a Profile can be Disabled by the SM-DP and the SM-SR when the MNO requests it. After the Profile disabling, an audit request is sent to the SM-SR to make sure that the Profile has been Disabled. An error case is also described:

- *the Profile with the Fall-back Attribute Set contains bad Connectivity Parameters*

Referenced Requirements

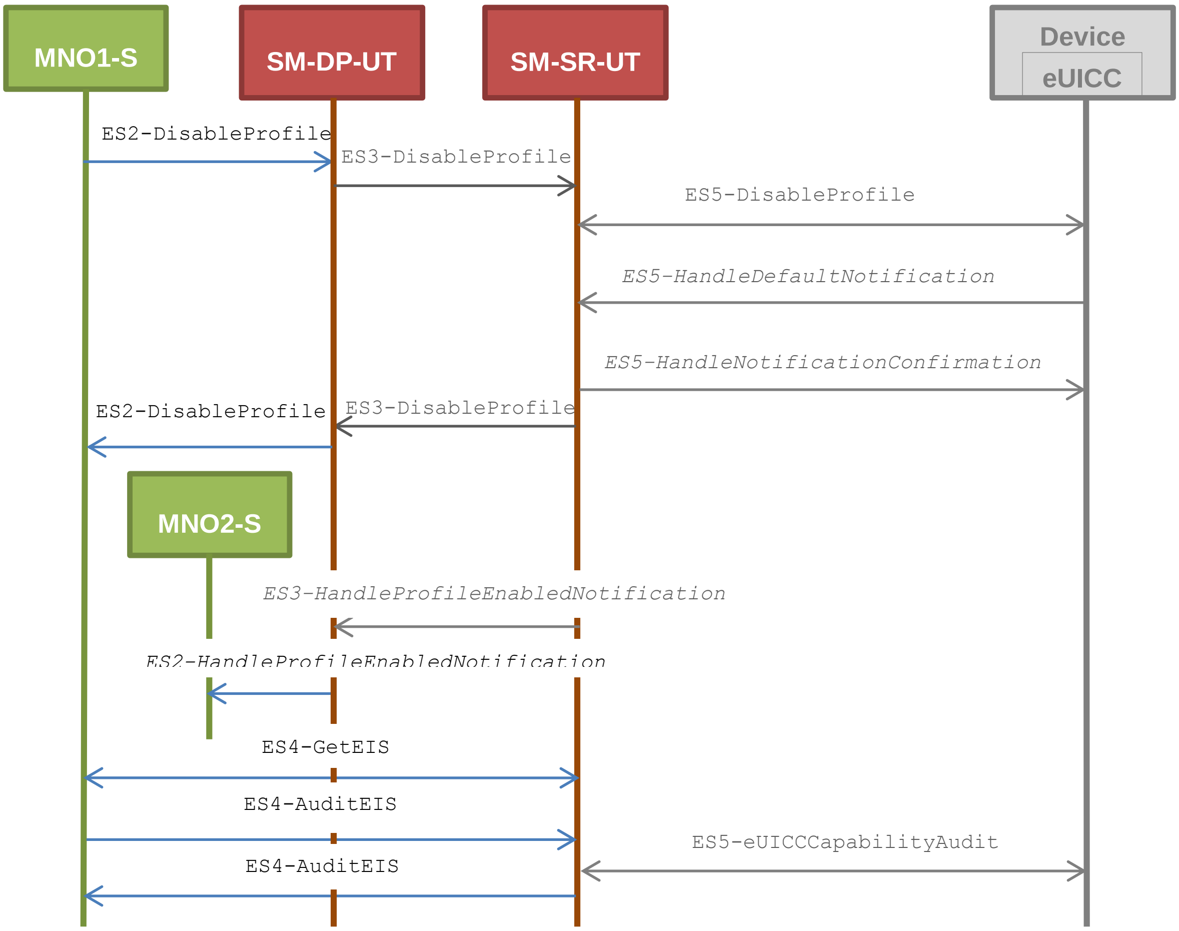
- PF_REQ2, PF_REQ5, PF_REQ7, PF_REQ13, PF_REQ16, PF_REQ19, PF_REQ22
- PROC_REQ10, PROC_REQ20, PROC_REQ22
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

Initial Conditions

- #MNO2_S_ACCESSPOINT is unknown to the SM-SR-UT
- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #MNO2_S_ID and #MNO2_S_ACCESSPOINT well known to the SM-DP-UT
- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS
- #SM_SR_ID and #SM_SR_ACCESSPOINT well known to the SM-DP-UT
- #SM_DP_ID and #SM_DP_ACCESSPOINT well known to the SM-SR-UT
- The Profile identified by #ICCID is linked to the SM-DP identified by #SM_DP_ID (the #EIS_RPS may need to be adapted on the SM-SR-UT)

5.3.4.2.2.1 Test Sequence N°1 – Nominal Case: Empty POL1 and POL2

Test Environment



Initial Conditions

- POL1 and POL2 of the Profile identified by #NEW_ICCID do not contain any rules
  - Disabling of the Profile is allowed

- "Profile deletion is mandatory when it is disabled" is not set

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DisableProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the `ES2-DisableProfile` response	The `Status` is equal to `#SUCCESS`	PF_REQ5, PF_REQ13, PF_REQ19, PF_REQ22, PROC_REQ10, PROC_REQ20, PROC_REQ22, EUICC_REQ27, EUICC_REQ29
4	SM-SR-UT → MNO2-S	Send the `ES2-HandleProfileEnabledNotification` notification	1- The `EID` parameter is equal to `#EID_RPS` 2- The ICCID is equal to `#ICCID_RPS` 3- The completion timestamp is present	PF_REQ16, PROC_REQ10

5	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

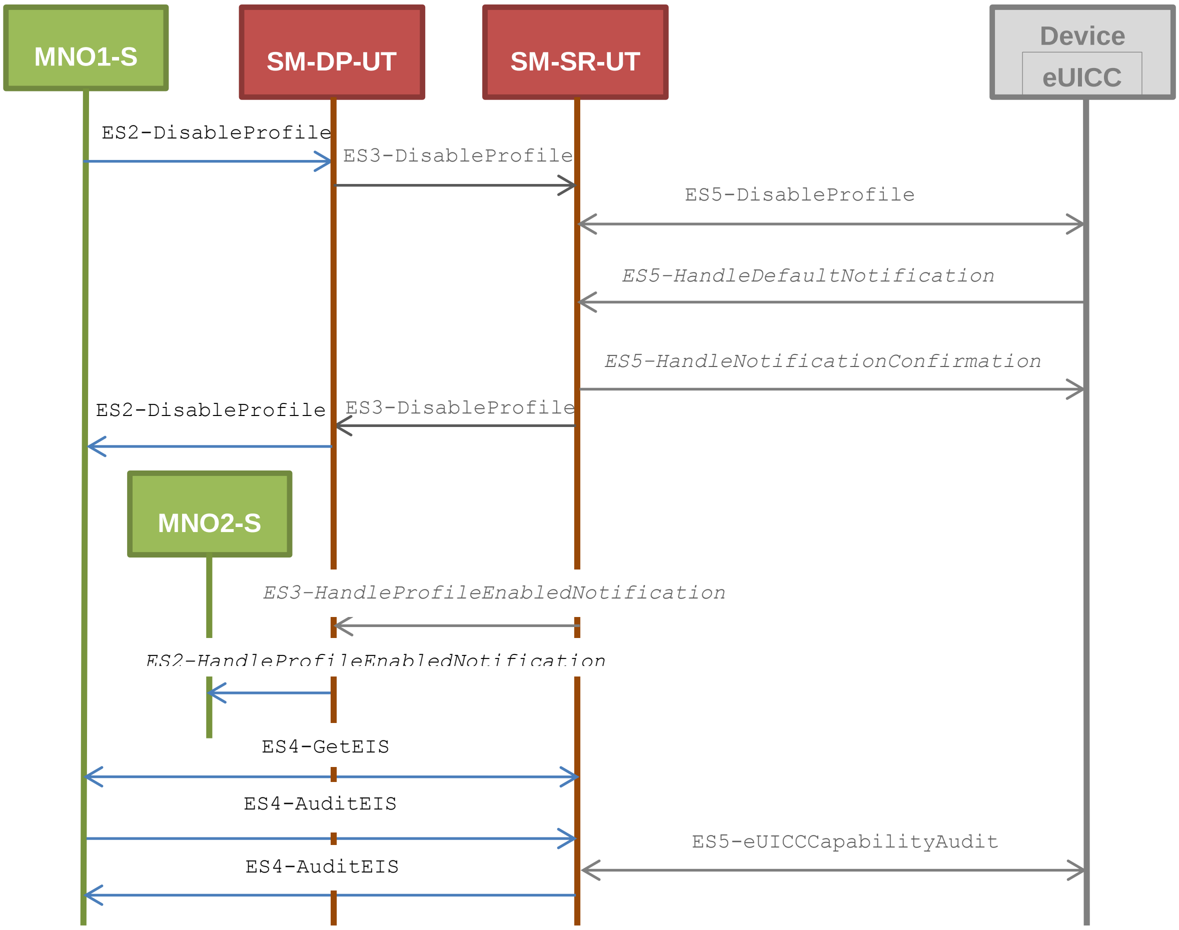
6	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. the Profile identified by `#NEW_ICCID` is Disabled	PM_REQ22

7	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 6	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.2.2 Test Sequence N°2 – Nominal Case: POL1 with “Profile Deletion is Mandatory when it is Disabled”

Test Environment



Initial Conditions

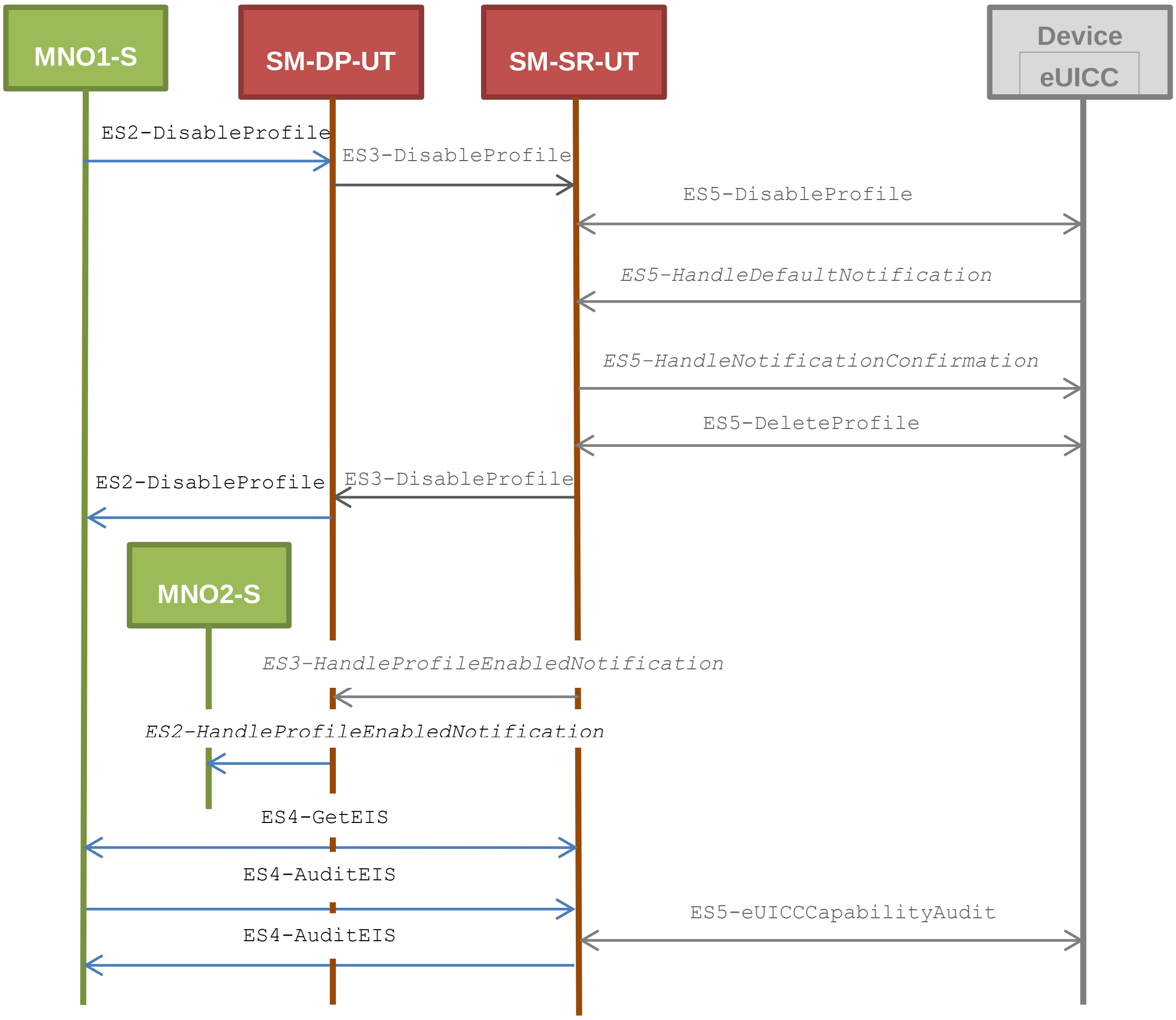
- POL1 of the Profile identified by `#NEW_ICCID` contains the rule “Profile deletion is mandatory when it is disabled”
- POL2 of the Profile identified by `#NEW_ICCID` allows disabling

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DisableProfile, #EID_RPS, {SM_SR_ID_RPS}, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-DP-UT → MNO1-S	Send the ES2-DisableProfile response	The Status is equal to #SUCCESS	PF_REQ5, PF_REQ13, PF_REQ19, PF_REQ22, PROC_REQ10, PROC_REQ20, PROC_REQ22, EUICC_REQ27, EUICC_REQ29
4	SM-SR-UT → MNO2-S	Send the ES2-HandleProfileEnabledNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ16, PROC_REQ10
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. the Profile identified by #NEW_ICCID is not present	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 6 except that: a. the remaining memory and the available memory for Profiles are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.2.3 Test Sequence N°3 – Nominal Case: POL2 with "Profile Deletion is Mandatory when it is Disabled"

Test Environment



Initial Conditions

- POL1 of the Profile identified by `#NEW_ICCID` does not contain any rules
  - Disabling of the Profile is allowed
  - "Profile deletion is mandatory when it is disabled" is not set
- POL2 of the Profile identified by `#NEW_ICCID` contains the rule "Profile deletion is mandatory when it is disabled"

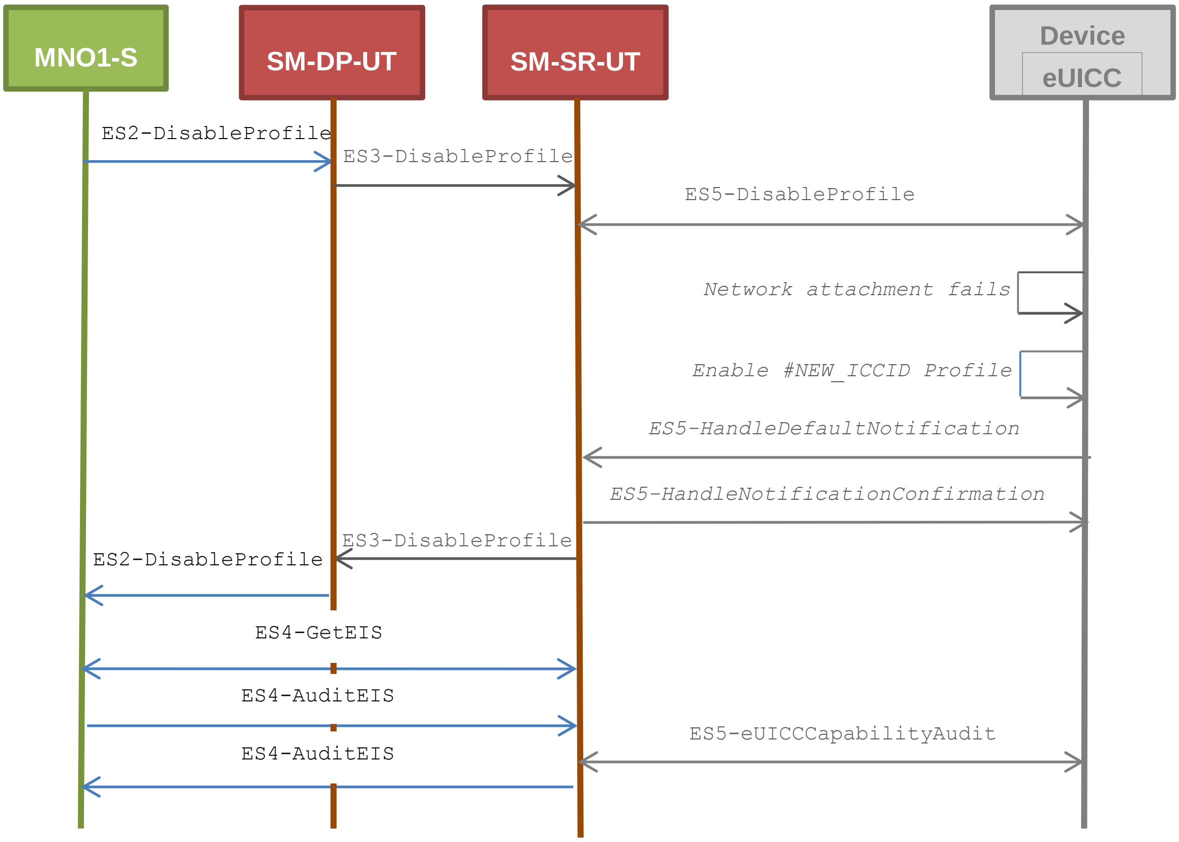
Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DisableProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the ES2-DisableProfile response	The Status is equal to #SUCCESS	PF_REQ5, PF_REQ13, PF_REQ19, PF_REQ22, PROC_REQ10, PROC_REQ20, PROC_REQ22, EUICC_REQ27, EUICC_REQ29
4	SM-SR-UT → MNO2-S	Send the ES2-HandleProfileEnabledNotification notification	1- The EID parameter is equal to #EID_RPS 2- The ICCID is equal to #ICCID_RPS 3- The completion timestamp is present	PF_REQ16, PROC_REQ10
5	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
6	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. the Profile identified by #NEW_ICCID is not present	PM_REQ22
7	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
9	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 6 except that: a. the remaining memory and the available memory for Profiles are updated (i.e. bigger than that received in step 6)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.4.2.2.4 Test Sequence N°4 – Error Case: Bad Connectivity Parameters

Test Environment



Initial Conditions

- The Profile, identified by `#ICCID`, shall be adapted to contain inconsistent Connectivity Parameters (e.g. `#NAN_VALUE, #LOGIN, #PWD`)

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DisableProfile, #EID_RPS, {SM_SR_ID_RPS}, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the ES2-DisableProfile response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_EUICC 3- The Reason code is equal to #RC_INACCESSIBLE	PF_REQ5, PF_REQ13, PF_REQ19, PF_REQ22, PROC_REQ10, PROC_REQ20, PROC_REQ22, EUICC_REQ27, EUICC_REQ29

Step	Direction	Sequence / Description	Expected result	REQ
4	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
5	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. the Profile identified by #NEW_ICCID is Enabled	PM_REQ22
6	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 5	PF_REQ2, PF_REQ7, PM_REQ26

5.3.5 Profile Deletion Process

5.3.5.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ2, PF_REQ6, PF_REQ7, PF_REQ14, PF_REQ20, PF_REQ26
- PROC_REQ11, PROC_REQ12
- PM_REQ22, PM_REQ26

5.3.5.2 Test Cases

General Initial Conditions

- #MNO1_S_ID well known to the SM-SR-UT
- #MNO1_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT
- #MNO2_S_ID well known to the SM-SR-UT
- The Profile identified by #ICCID is owned by MNO2-S and is in Enabled state
- The Profile identified by #NEW_ICCID is owned by MNO1-S and is in Disabled state

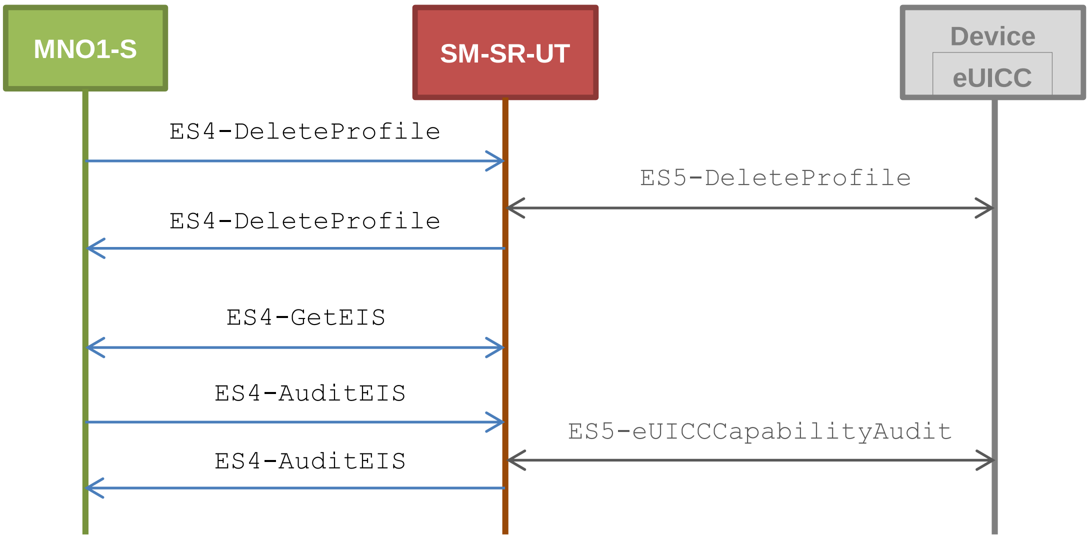
  - To download the new Profile (e.g. #PROFILE_PACKAGE), the test sequence defined in section 5.3.2.2.1.1 may be used
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by #ICCID)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the default Enabled Profile (i.e. #MNO2_CON_NAN, #MNO2_CON_LOGIN, #MNO2_CON_PWD)
- The eUICC identified by #EID has been initially provisioned on the SM-SR-UT using the #EIS_RPS

5.3.5.2.1 TC.PROC.DEL.1: ProfileDeletionByMNO

Test Purpose

To ensure a Profile can be deleted by the SM-SR when the MNO requests it. After the Profile deletion, an audit request is sent to the SM-SR to make sure that the Profile has been deleted. An error case with a POL1 defined with "Deletion not allowed" is also described.

Test Environment



Referenced Requirements

- PF_REQ2, PF_REQ6, PF_REQ7, PF_REQ26
- PROC_REQ11
- PM_REQ22, PM_REQ26

Initial Conditions

- The Profile identified by #ICCID is the Profile with the Fall-back Attribute Set

5.3.5.2.1.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- POL1 and POL2 of the Profile identified by #NEW_ICCID do not contain any rules
  - Deletion of the Profile is allowed

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-DeleteProfile response	The Status is equal to #SUCCESS	PF_REQ2, PF_REQ6, PF_REQ26, PROC_REQ11
4	MNO1-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
5	SM-SR-UT → MNO1-S	Send the ES4-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. the Profile identified by #NEW_ICCID is not present	PM_REQ22
6	MNO1-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the ES4-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS is equal to that received in step 5 except that: a. the remaining memory and the available memory for Profiles are updated (i.e. bigger than that received in step 5)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.5.2.1.2 Test Sequence N°2 - Error Case: POL1 with “Deletion not Allowed”

Initial Conditions

- POL1 of the Profile identified by #NEW_ICCID contains the rule “Deletion not Allowed”
- POL2 of the Profile identified by #NEW_ICCID does not contain any rules
  - Deletion of the Profile is allowed

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DeleteProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the `ES4-DeleteProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_POL1` 3- The `Reason code` is equal to `#RC_REFUSED` 4- The `euiccResponseData` is present and contains the POR generated by the eUICC (i.e. SW='69E1')	PF_REQ2, PF_REQ6, PF_REQ26, PROC_REQ11

4	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. the Profile identified by `#NEW_ICCID` is Disabled	PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

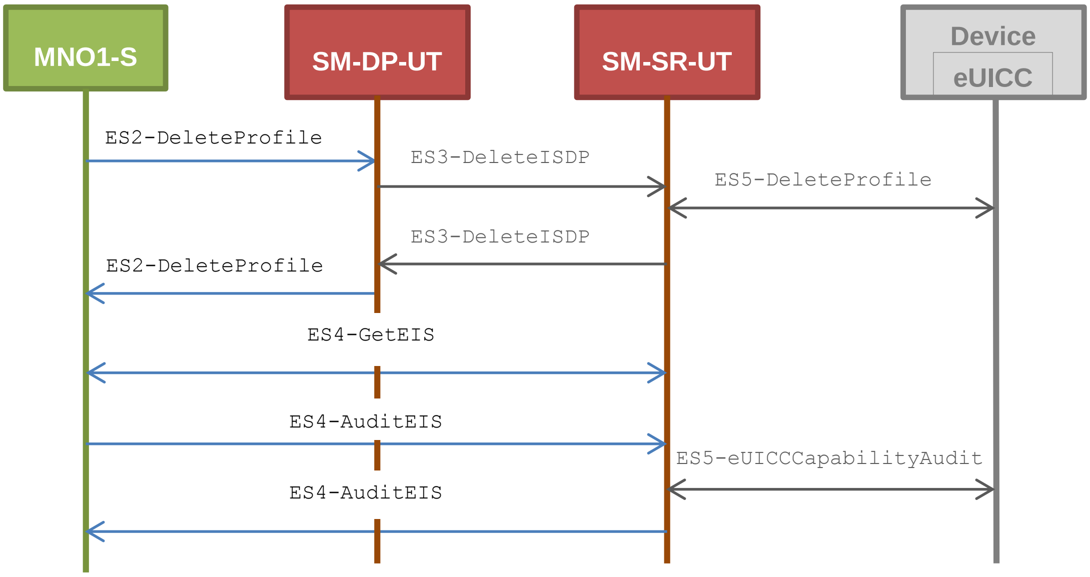
7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5	PF_REQ2, PF_REQ7, PM_REQ26

5.3.5.2.2 TC.PROC.DEL.1: ProfileDeletionViaSMDP

Test Purpose

To ensure a Profile can be deleted by the SM-DP and the SM-SR when the MNO requests it. After the Profile deletion, an audit request is sent to the SM-SR to make sure that the Profile has been deleted. An error case with a POL1 defined with "Deletion not allowed" is also described.

Test Environment



Referenced Requirements

- PF_REQ2, PF_REQ6, PF_REQ7, PF_REQ14, PF_REQ20
- PROC_REQ12
- PM_REQ22, PM_REQ26

Initial Conditions

- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- The variable {SM_SR_ID_RPS} shall be set to #SM_SR_UT_ID_RPS
- #SM_SR_ID and #SM_SR_ACCESSPOINT well known to the SM-DP-UT

5.3.5.2.2.1 Test Sequence N°1 - Nominal Case

Initial Conditions

- POL1 and POL2 of the Profile identified by #NEW_ICCID do not contain any rules
  - Deletion of the Profile is allowed

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DeleteProfile, #EID_RPS, {SM_SR_ID_RPS}, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-DP-UT → MNO1-S	Send the `ES2-DeleteProfile` response	The `Status` is equal to `#SUCCESS`	PF_REQ2, PF_REQ6, PF_REQ14, PF_REQ20, PROC_REQ12

4	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. the Profile identified by `#NEW_ICCID` is not present	PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5 except: a. the `remaining memory` and the `available memory for Profiles` are updated (i.e. bigger than that received in step 5)	PF_REQ2, PF_REQ7, PM_REQ26

5.3.5.2.2.2 Test Sequence N°2 - Error Case: POL1 with "Deletion not Allowed"

Initial Conditions

- POL1 of the Profile identified by `#NEW_ICCID` contains the rule "Deletion not Allowed"
- POL2 of the Profile identified by `#NEW_ICCID` does not contain any rules
  - Deletion of the Profile is allowed

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO1-S → SM-DP-UT	`SEND_REQ(`
		`ES2-DeleteProfile,`
		`#EID_RPS,`
		`{SM_SR_ID_RPS},`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-DP-UT → MNO1-S	Send the `ES2-DeleteProfile` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_POL1` 3- The `Reason code` is equal to `#RC_REFUSED` 4- The `euiccResponseData` is present and contains the POR generated by the eUICC (i.e. SW='69E1')	PF_REQ2, PF_REQ6, PF_REQ14, PF_REQ20, PROC_REQ12

4	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R and ECASD information are not present b. the Profile identified by `#NEW_ICCID` is Disabled	PM_REQ22

6	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
8	SM-SR-UT → MNO1-S	Send the `ES4-AuditEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to that received in step 5	PF_REQ2, PF_REQ7, PM_REQ26

5.3.6 Master Delete Process

As no interface is defined between the MNO, the SM-DP and the SM-SR in the GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2], this section is FFS. Only test cases that allow testing the eUICC are defined (see section 4.2.9).

5.3.7 SM-SR Change Process

5.3.7.1 Conformance Requirements

References

- GSMA Embedded SIM Remote Provisioning Architecture [1]
- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PF_REQ2, PF_REQ7
- EUICC_REQ24, EUICC_REQ25, EUICC_REQ33, EUICC_REQ34, EUICC_REQ35, EUICC_REQ36, EUICC_REQ37, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40, EUICC_REQ41
- PM_REQ22, PM_REQ25
- PROC_REQ13
- SEC_REQ19

5.3.7.2 Test Cases

General Initial Conditions

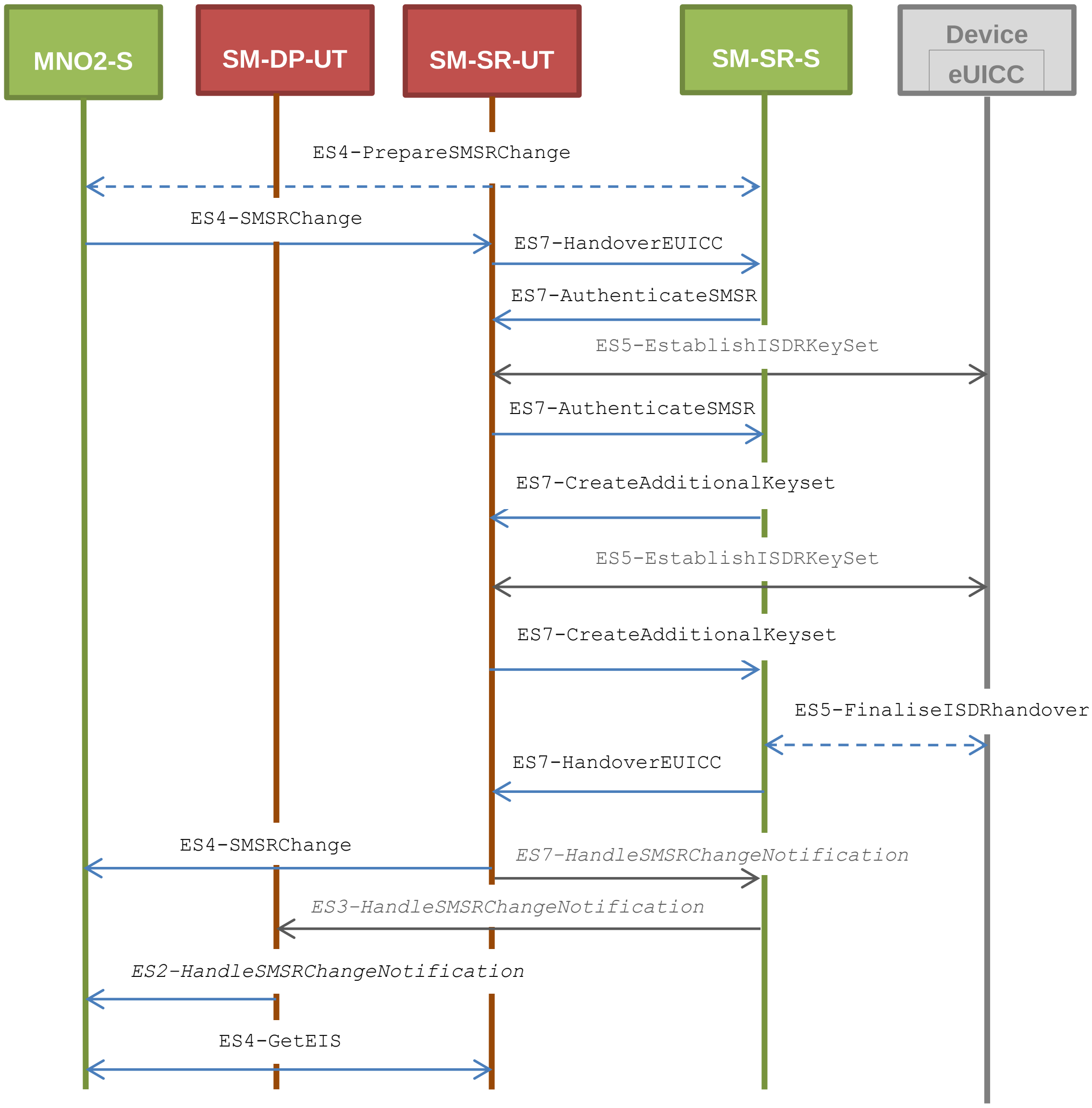
- `#MNO1_S_ID` well known to the SM-SR-UT
- `#MNO2_S_ID` well known to the SM-SR-UT
- The Profile identified by `#ICCID` is owned by MNO2-S and is in Enabled state
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by `#ICCID`)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the default Enabled Profile (i.e. `#MNO2_CON_NAN, #MNO2_CON_LOGIN, #MNO2_CON_PWD`)

5.3.7.2.1 TC.PROC.SMSRCH.1: SMSRChange

Test Purpose

To ensure the SM-SR can be changed when the MNO requests it. In this test case, the switch is from the SM-SR-UT to the SM-SR-S.

Test Environment



Note that the functions `ES4-PrepareSMSRChange` and `ES5-FinaliseISDRhandover` shall not be performed by the simulators (in the schema above, they are only informative messages).
In this test case, the Initiator Role (see GSMA Embedded SIM Remote Provisioning Architecture [1] section 2.3.1) is assumed to be played by the MNO2-S.

Referenced Requirements

- PF_REQ2
- EUICC_REQ24, EUICC_REQ33, EUICC_REQ34, EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40, EUICC_REQ41
- PM_REQ22
- PROC_REQ13
- SEC_REQ19

Initial Conditions

- `#MNO2_S_ACCESSPOINT` is unknown to the SM-SR-UT

- #MNO2_S_ID and #MNO2_S_ACCESSPOINT well known to the SM-DP-UT
- The eUICC identified by #EID has been initially provisioned on the SM-SR-UT using the #EIS_RPS
- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)

5.3.7.2.1.1 Test Sequence N°1 – Nominal Case: No DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO2-S → SM-SR-UT	SEND_REQ(ES4-SMSRChange, #EID_RPS, #TGT_SR_S_ID_RPS)		
2	SM-SR-UT → SM-SR-S	Send the ES7-HandoverEUICC request	The EIS is equal to #EIS_RPS	EUICC_REQ36, EUICC_REQ39, PROC_REQ13
3	SM-SR-S → SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #EID_RPS, #VALID_SR_CERTIF_RPS)		
4	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
5	SM-SR-UT → SM-SR-S	Send the ES7-AuthenticateSMSR response	1- The Status is equal to #SUCCESS 2- The Random Challenge is present (i.e. {RC})	PF_REQ2, EUICC_REQ24, EUICC_REQ36, EUICC_REQ39, EUICC_REQ40, PROC_REQ13
6	SM-SR-S → SM-SR-UT	SEND_REQ(ES7-CreateAdditionalKeyset, #EID_RPS, #KEY_VERSION_RPS, #INIT_SEQ_COUNTER_RPS, #ECC_KEY_LENGTH_RPS, #SC3_NO_DR_RPS, #EPHEMERAL_PK_RPS, #SIGNATURE_RPS)		
7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-UT → SM-SR-S	Send the `ES7-CreateAdditionalKeyset` response	1- The `Status` is equal to `#SUCCESS` 2- The `derivation random` is not present 3- The `receipt` (i.e. `{RECEIPT}`) is present 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tag 'A6')	PF_REQ2, EUICC_REQ24, EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, PROC_REQ13
9	SM-SR-S → SM-SR-UT	`SEND_SUCCESS_RESP( ES7-HandoverEUICC)`		
10	SM-SR-UT → SM-SR-S	Send the `ES7-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` 2- The completion timestamp is present	EUICC_REQ41
11	SM-SR-UT → MNO2-S	Send the `ES4-SMSRChange` response	The `Status` is equal to `#SUCCESS`	EUICC_REQ36, PROC_REQ13

12	SM-SR-S → SM-DP-UT	`SEND_NOTIF(`
		`ES3-HandleSMSRChangeNotification,`
		`#EIS_RPS,`
		`#TIMESTAMP_RPS)`		

13	SM-DP-UT → MNO2-S	Send the `ES2-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` 2- The completion timestamp is equal to `#TIMESTAMP_RPS`	EUICC_REQ33, EUICC_REQ34, PROC_REQ13

14	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
15	SM-SR-UT → MNO2-S	Send the `ES4-GetEIS` response	1- The Status is equal to `#FAILED` 2- The Subject code is equal to `#SC_EID` 3- The Reason code is equal to `#RC_UNKNOWN`	PM_REQ22, SEC_REQ19

5.3.7.2.1.2 Test Sequence N°2 – Nominal Case: DR, No Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-SMSRChange,`
		`#EID_RPS,`
		`#TGT_SR_S_ID_RPS)`		

2	SM-SR-UT → SM-SR-S	Send the `ES7-HandoverEUICC` request	The `EIS` is equal to `#EIS_RPS`	EUICC_REQ36, EUICC_REQ39, PROC_REQ13

3	SM-SR-S → SM-SR-UT	`SEND_REQ(`
		`ES7-AuthenticateSMSR,`
		`#EID_RPS,`
		`#VALID_SR_CERTIF_RPS)`		

4	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
5	SM-SR-UT → SM-SR-S	Send the `ES7-AuthenticateSMSR` response	1- The `Status` is equal to `#SUCCESS` 2- The `Random Challenge` is present (i.e. `{RC}`)	EUICC_REQ24, EUICC_REQ36, EUICC_REQ39, EUICC_REQ40, PROC_REQ13

6	SM-SR-S → SM-SR-UT	`SEND_REQ(`
		`ES7-CreateAdditionalKeyset,`
		`#EID_RPS,`
		`#KEY_VERSION_RPS,`
		`#INIT_SEQ_COUNTER_RPS,`
		`#ECC_KEY_LENGTH_RPS,`
		`#SC3_DR_RPS,`
		`#EPHEMERAL_PK_RPS,`
		`#SIGNATURE_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-UT → SM-SR-S	Send the `ES7-CreateAdditionalKeyset` response	1- The `Status` is equal to `#SUCCESS` 2- The `derivation random` is present (i.e. `{DR}`) 3- The `receipt` (i.e. `{RECEIPT}`) is present 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS and `{DR}` and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tags ‘A6’ and ‘85’)	EUICC_REQ24, EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, PROC_REQ13

9	SM-SR-S → SM-SR-UT	`SEND_SUCCESS_RESP(`
		`ES7-HandoverEUICC)`		

10	SM-SR-UT → SM-SR-S	Send the `ES7-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` 2- The completion timestamp is present	EUICC_REQ41
11	SM-SR-UT → MNO2-S	Send the `ES4-SMSRChange` response	The `Status` is equal to `#SUCCESS`	EUICC_REQ36, PROC_REQ13

12	SM-SR-S → SM-DP-UT	`SEND_NOTIF(`
		`ES3-HandleSMSRChangeNotification,`
		`#EIS_RPS,`
		`#TIMESTAMP_RPS)`		

13	SM-DP-UT → MNO2-S	Send the `ES2-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` 2- The completion timestamp is equal to `#TIMESTAMP_RPS`	EUICC_REQ33, EUICC_REQ34, PROC_REQ13

14	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
15	SM-SR-UT → MNO2-S	Send the `ES4-GetEIS` response	1- The Status is equal to `#FAILED` 2- The Subject code is equal to `#SC_EID` 3- The Reason code is equal to `#RC_UNKNOWN`	PM_REQ22, SEC_REQ19

5.3.7.2.1.3 Test Sequence N°3 – Nominal Case: DR, Host ID

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-SMSRChange,`
		`#EID_RPS,`
		`#TGT_SR_S_ID_RPS)`		

2	SM-SR-UT → SM-SR-S	Send the `ES7-HandoverEUICC` request	The `EIS` is equal to `#EIS_RPS`	EUICC_REQ36, EUICC_REQ39, PROC_REQ13

3	SM-SR-S → SM-SR-UT	`SEND_REQ(`
		`ES7-AuthenticateSMSR,`
		`#EID_RPS,`
		`#VALID_SR_CERTIF_RPS)`		

4	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
5	SM-SR-UT → SM-SR-S	Send the `ES7-AuthenticateSMSR` response	1- The `Status` is equal to `#SUCCESS` 2- The `Random Challenge` is present (i.e. `{RC})`	EUICC_REQ24, EUICC_REQ36, EUICC_REQ39, EUICC_REQ40, PROC_REQ13

6	SM-SR-S → SM-SR-UT	`SEND_REQ(`
		`ES7-CreateAdditionalKeyset,`
		`#EID_RPS,`
		`#KEY_VERSION_RPS,`
		`#INIT_SEQ_COUNTER_RPS,`
		`#ECC_KEY_LENGTH_RPS,`
		`#SC3_DR_HOST_RPS,`
		`#HOST_ID_RPS,`
		`#EPHEMERAL_PK_RPS,`
		`#SIGNATURE_RPS)`		

7	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-UT → SM-SR-S	Send the `ES7-CreateAdditionalKeyset` response	1- The `Status` is equal to `#SUCCESS` 2- The `derivation random` is present (i.e. `{DR}`) 3- The `receipt` (i.e. `{RECEIPT}`) is present 4- Calculate ShS from `#SM_ESK_ECKA` and `#PK_ECASD_ECKA` 5- Derive keyset from ShS and `{DR}` and retrieve the `{SCP_KENC}`, `{SCP_KMAC}` and `{SCP_KDEK}` 6- Verify the `{RECEIPT}` (i.e. it shall be generated by calculating a MAC across the tags ‘A6’ and ‘85’)	EUICC_REQ24, EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, PROC_REQ13

9	SM-SR-S → SM-SR-UT	`SEND_SUCCESS_RESP(`
		`ES7-HandoverEUICC)`		

10	SM-SR-UT → SM-SR-S	Send the `ES7-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` 2- The completion timestamp is present	EUICC_REQ41
11	SM-SR-UT → MNO2-S	Send the `ES4-SMSRChange` response	The `Status` is equal to `#SUCCESS`	EUICC_REQ36, PROC_REQ13

12	SM-SR-S → SM-DP-UT	`SEND_NOTIF(`
		`ES3-HandleSMSRChangeNotification,`
		`#EIS_RPS,`
		`#TIMESTAMP_RPS)`		

13	SM-DP-UT → MNO2-S	Send the `ES2-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` 2- The completion timestamp is equal to `#TIMESTAMP_RPS`	EUICC_REQ33, EUICC_REQ34, PROC_REQ13

14	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

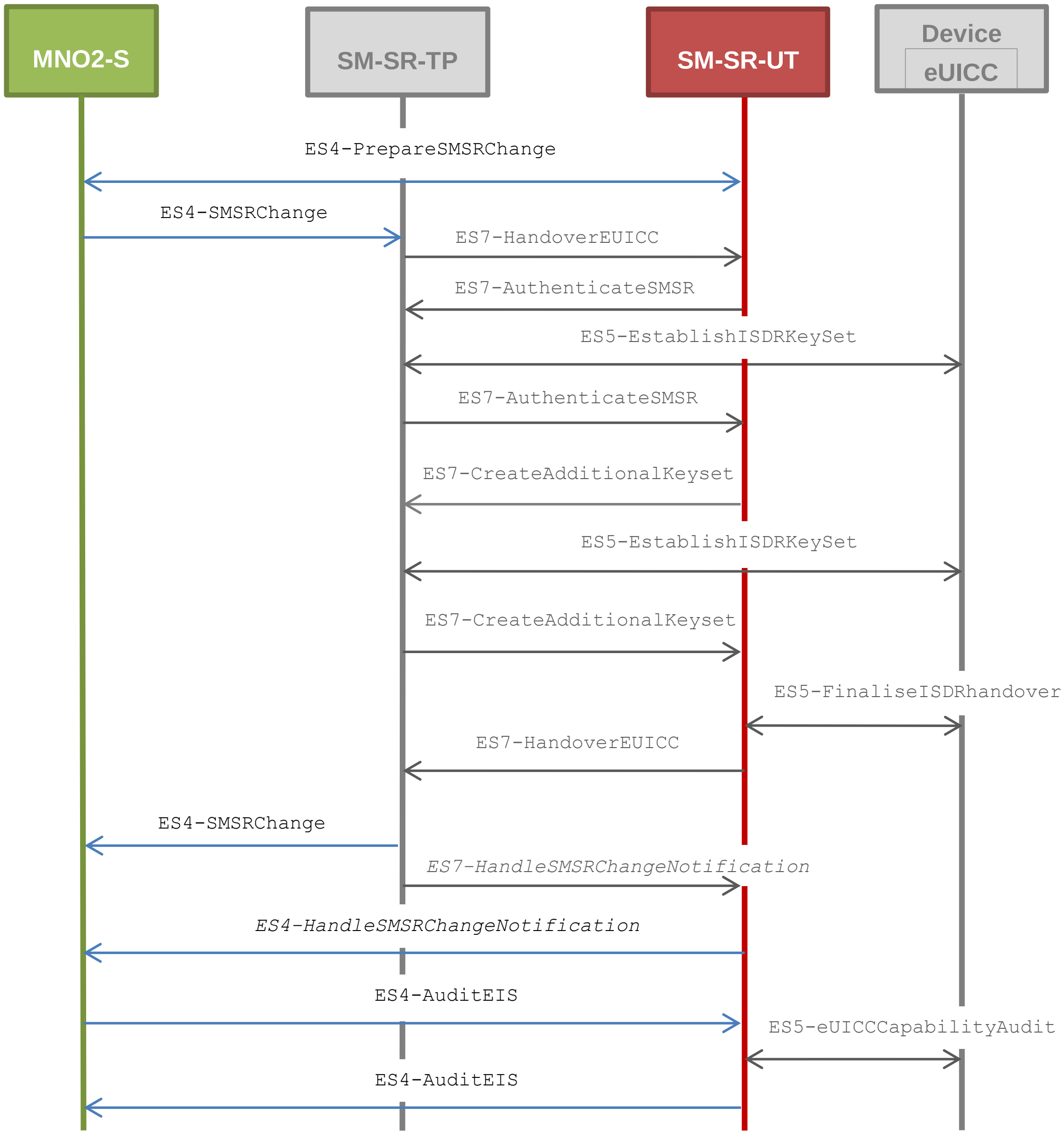
Step	Direction	Sequence / Description	Expected result	REQ
15	SM-SR-UT → MNO2-S	Send the `ES4-GetEIS` response	1- The Status is equal to `#FAILED` 2- The Subject code is equal to `#SC_EID` 3- The Reason code is equal to `#RC_UNKNOWN`	PM_REQ22, SEC_REQ19

5.3.7.2.2 TC.PROC.SMSRCH.2: SMSRChange

Test Purpose

To ensure the SM-SR can be changed when the MNO requests it. In this test case, the switch is from the SM-SR-TP to SM-SR-UT.

Test Environment



In this test case, the Initiator Role (see GSMA Embedded SIM Remote Provisioning Architecture [1] section 2.3.1) is assumed to be played by the MNO2-S.

Referenced Requirements

- PF_REQ2, PF_REQ7
- EUICC_REQ25, EUICC_REQ35, EUICC_REQ36, EUICC_REQ37, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40, EUICC_REQ41
- PM_REQ25
- PROC_REQ13

Initial Conditions

- `#MNO1_S_ID` well known to the SM-SR-TP
- `#MNO2_S_ID` well known to the SM-SR-TP
- `#MNO2_S_ACCESSPOINT` well known to the SM-SR-UT
  - A direct connection exists between the MNO2-S and the SM-SR-UT
- The eUICC identified by `#EID` has been initially provisioned on the SM-SR-TP using the `#EIS_RPS`
- The SM-SR-TP is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by `#ICCID`)
  - It means that the SM-SR-TP knows the Connectivity Parameters of the MNO's network related to the default Enabled Profile (i.e. `#MNO2_CON_NAN, #MNO2_CON_LOGIN, #MNO2_CON_PWD`)
- All necessary settings have been initialized on SM-SR-TP to accept the SM-SR change (i.e. business agreement…)

5.3.7.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO2-S →
	SM-SR-UT
		`SEND_REQ(`
		`ES4-PrepareSMSRChange,`
		`#EID_RPS,`
		`#CUR_SR_ID_RPS)`
		see Note 1		

2	SM-SR-UT → MNO2-S	Send the `ES4-PrepareSMSRChange` response	The `Status` is equal to `#SUCCESS`	EUICC_REQ35, PROC_REQ13

3	MNO2-S →
	SM-SR-TP
		`SEND_REQ(`
		`ES4-SMSRChange,`
		`#EID_RPS,`
		`#TGT_SR_UT_ID_RPS)`		

4	*Wait until a response is received (the SM-SR-TP and SM-SR-UT treatments may take several minutes)*			

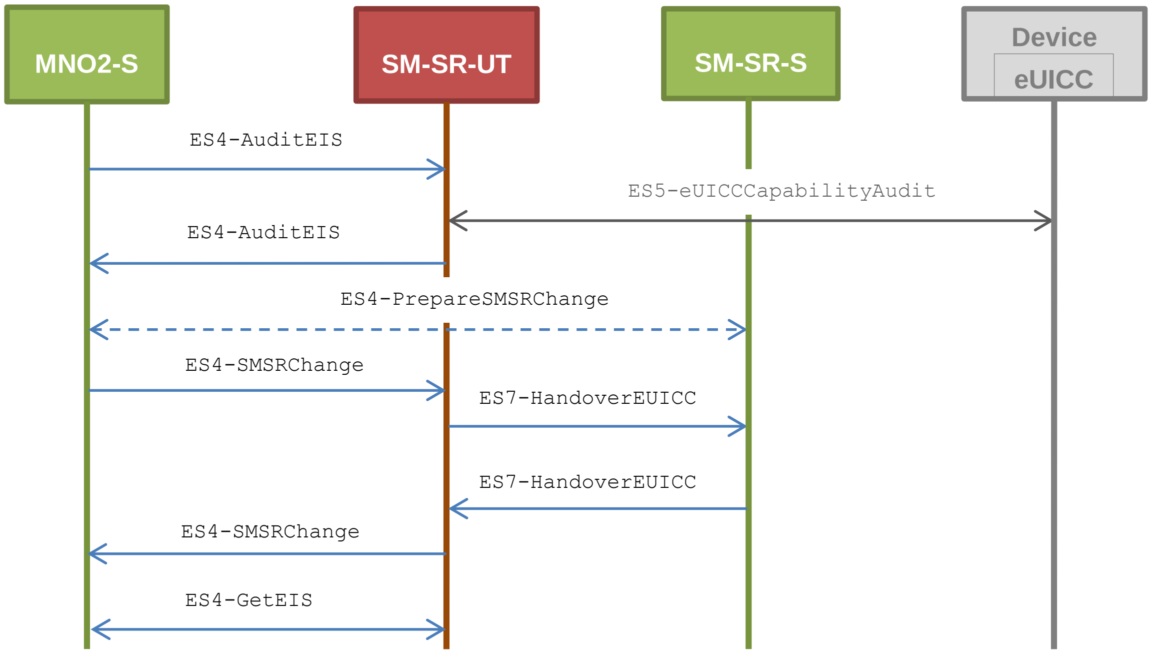
Step	Direction	Sequence / Description	Expected result	REQ
5	SM-SR-TP → MNO2-S	Send the ES4-SMSRChange response	The Status is equal to #SUCCESS	EUICC_REQ25, EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40, EUICC_REQ41, PROC_REQ13, PF_REQ2
6	SM-SR-UT → MNO2-S	SEND_NOTIF(ES4-HandleSMSRChangeNotification, #EIS_RPS, #TIMESTAMP_RPS)		EUICC_REQ37
7	MNO2-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS)		
8	*Wait until a response is received (the SM-SR-UT treatments may take several minutes)*			
9	SM-SR-UT → MNO2-S	Send the ES4-AuditEIS response see Note 2	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R and ECASD information are not present b. only Profiles related to the MNO2-S are present	PM_REQ25, PROC_REQ13, PF_REQ7, PF_REQ2
Note 1: In the #CUR_SR_ID_RPS, the SM-SR identifier is the SM-SR-TP one (not the SM-SR-UT one) *Note 2: Before performing this operation, the SM-SR-UT should use the ES5-UpdateSMSRAddressingParameters method to set the #SM_SR_DEST_ADDR (and optionally the #SM_SR_UDP_IP, #SM_SR_UDP_PORT, #SM_SR_TCP_IP, #SM_SR_TCP_PORT, #SM_SR_HTTP_URI and #SM_SR_HTTP_HOST).*				

5.3.7.2.3 TC.PROC.SMSRCH.3: SMSRChange

Test Purpose

To ensure the SM-SR change process is correctly implemented when an error occurs during the procedure.
To make sure that the audit trail contains an audit operation in the function ES7-HandoverEUICC, an audit request is sent on the current SM-SR before launching the SM-SR change process.
As the SM-SR change fails, the eUICC shall be associated to the same SM-SR (i.e. SM-SR-UT).

Test Environment



Note that the function `ES4-PrepareSMSRChange` shall not be performed by the simulators (in the schema above, this is only an informative message).
In this test case, the Initiator Role (see GSMA Embedded SIM Remote Provisioning Architecture [1] section 2.3.1) is assumed to be played by the MNO2-S.

Referenced Requirements

- PF_REQ2, PF_REQ7
- EUICC_REQ36, EUICC_REQ39
- PM_REQ22, PM_REQ25
- PROC_REQ13

Initial Conditions

- The eUICC identified by `#EID` has been initially provisioned on the SM-SR-UT using the `#EIS_RPS`
- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)

5.3.7.2.3.1 Test Sequence N°1 – Error Case: Unable to manage the eUICC

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-AuditEIS,`
		`#EID_RPS,`
		`#ICCID_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → MNO2-S	Send the `ES4-AuditEIS` response	The `Status` is equal to `#SUCCESS`	PF_REQ2, PF_REQ7, PM_REQ25

4	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-SMSRChange,`
		`#EID_RPS,`
		`#TGT_SR_S_ID_RPS)`		

5	SM-SR-UT → SM-SR-S	Send the `ES7-HandoverEUICC` request	The `EIS` is equal to `#EIS_RPS` except that: a. the `audit trail` is present and contains the operation `#AUDIT_OPERATION_RPS` (i.e. other records may be present) b. the `last audit date` is present and equal to `{CURRENT_DATE}`	EUICC_REQ36, EUICC_REQ39, PROC_REQ13

6	SM-SR-S → SM-SR-UT	`SEND_ERROR_RESP(`
		`ES7-HandoverEUICC,`
		`#FAILED,`
		`#SC_FUN_PROV,`
		`#RC_COND_USED)`		

7	SM-SR-UT → MNO2-S	Send the `ES4-SMSRChange` response	1- The `Status` is equal to `#FAILED` 2- The `Subject code` is equal to `#SC_FUN_PROV` 3- The `Reason code` is equal to `#RC_COND_USED`	EUICC_REQ36, PROC_REQ13

8	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

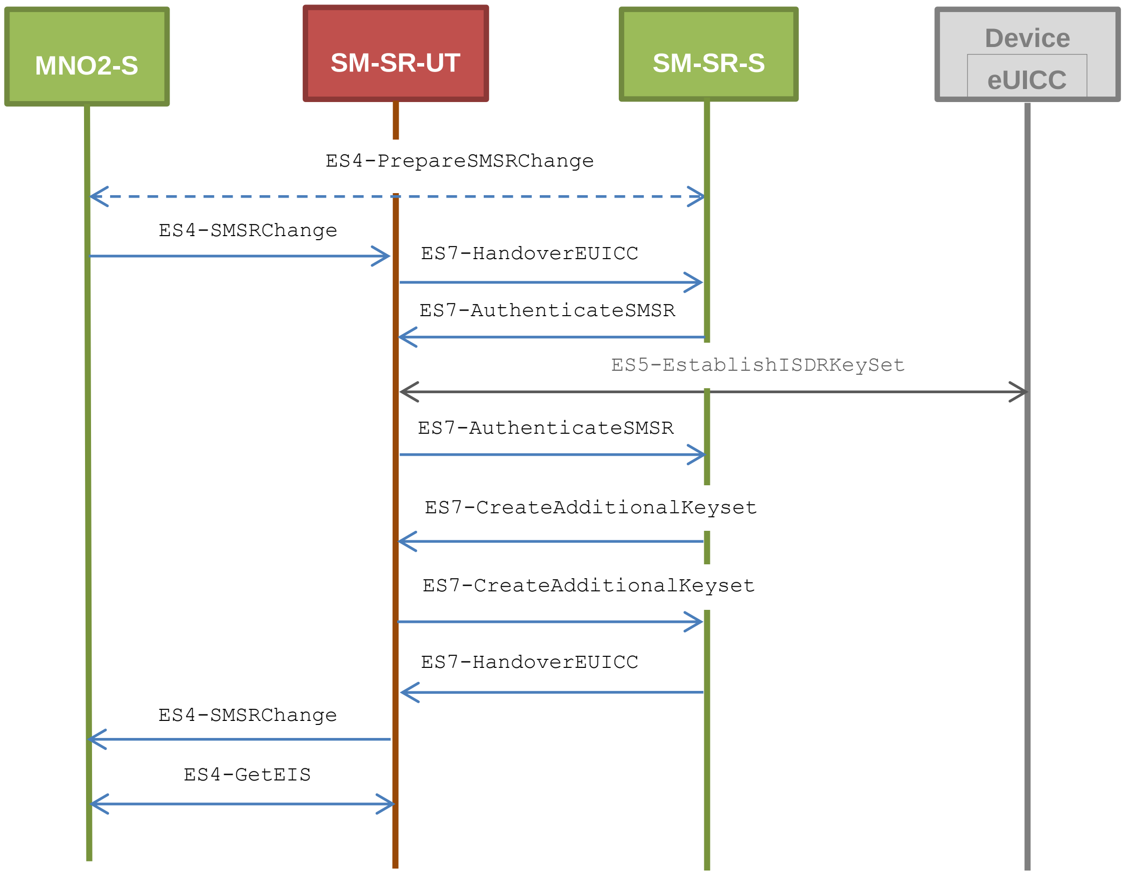
9	SM-SR-UT → MNO2-S	Send the `ES4-GetEIS` response	The `Status` is equal to `#SUCCESS`	PM_REQ22, PROC_REQ13

5.3.7.2.4 TC.PROC.SMSRCH.4: SMSRChange

Test Purpose

To ensure the SM-SR change process is correctly implemented when an error occurs during the procedure. In this particular test case, a conditional parameter (i.e. HostID) is missing in *the input parameters of the method ES7-CreateAdditionalKeyset. As the SM-SR change fails, the eUICC shall be associated to the same SM-SR (i.e. SM-SR-UT).*

Test Environment



Note that the function `ES4-PrepareSMSRChange` shall not be performed by the simulators (in the schema above, this is only an informative message).
In this test case, the Initiator Role (see GSMA Embedded SIM Remote Provisioning Architecture [1] section 2.3.1) is assumed to be played by the MNO2-S.

Referenced Requirements

- PF_REQ2
- EUICC_REQ24, EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40
- PM_REQ22
- PROC_REQ13

Initial Conditions

- The eUICC identified by `#EID` has been initially provisioned on the SM-SR-UT using the `#EIS_RPS`
- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)

5.3.7.2.4.1 Test Sequence N°1 – Error Case: Missing Host ID parameter

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO2-S → SM-SR-UT	SEND_REQ(ES4-SMSRChange, #EID_RPS, #TGT_SR_S_ID_RPS)		
2	SM-SR-UT → SM-SR-S	Send the ES7-HandoverEUICC request	The EIS is equal to #EIS_RPS	EUICC_REQ36, EUICC_REQ39, PROC_REQ13
3	SM-SR-S → SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #EID_RPS, #VALID_SR_CERTIF_RPS)		
4	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
5	SM-SR-UT → SM-SR-S	Send the ES7-AuthenticateSMSR response	1- The Status is equal to #SUCCESS 2- The Random Challenge is present (i.e. {RC})	PF_REQ2, EUICC_REQ24, EUICC_REQ36, EUICC_REQ39, EUICC_REQ40, PROC_REQ13
6	SM-SR-S → SM-SR-UT	SEND_REQ(ES7-CreateAdditionalKeyset, #EID_RPS, #KEY_VERSION_RPS, #INIT_SEQ_COUNTER_RPS, #ECC_KEY_LENGTH_RPS, #SC3_DR_HOST_RPS, #EPHEMERAL_PK_RPS, #SIGNATURE_RPS)		
7	SM-SR-UT → SM-SR-S	Send the ES7-CreateAdditionalKeyset response	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_FUNCTION 3- The Reason code is equal to #RC_COND_PARAM	EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, PROC_REQ13

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-S → SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #FAILED, #SC_FUN_PROV, #RC_COND_PARAM)		
9	SM-SR-UT → MNO2-S	Send the ES4-SMSRChange response	1- The Status is equal to #FAILED 2-The Subject code is equal to #SC_FUNCTION 3- The Reason code is equal to #RC_COND_PARAM	EUICC_REQ36, PROC_REQ13
10	MNO2-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
11	SM-SR-UT → MNO2-S	Send the ES4-GetEIS response	The Status is equal to #SUCCESS	PM_REQ22, PROC_REQ13

5.3.8 Update Connectivity Parameters Process

5.3.8.1 Conformance Requirements

References

- GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2]

Requirements

- PROC_REQ19
- PM_REQ21

5.3.8.2 Test Cases

General Initial Conditions

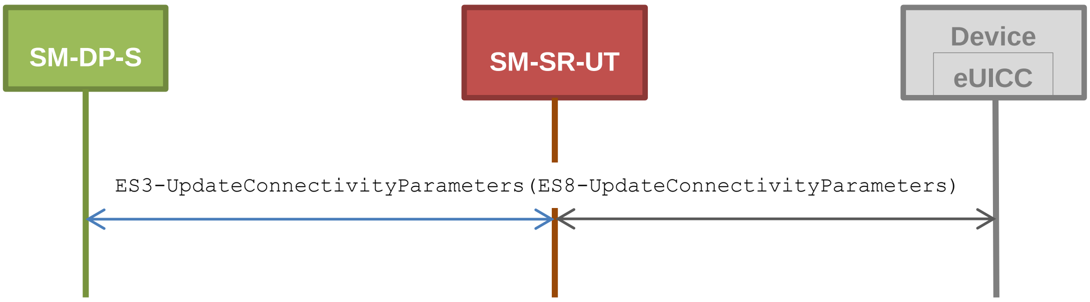
- #MNO1_S_ID well known to the SM-SR-UT
- #MNO2_S_ID well known to the SM-SR-UT
- The Profile identified by #ICCID is owned by MNO2-S and is in Enabled state
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by #ICCID)
  - It means that the SM-SR-UT knows the Connectivity Parameters of the MNO's network related to the default Enabled Profile (i.e. #MNO2_CON_NAN, #MNO2_CON_LOGIN, #MNO2_CON_PWD)
- The eUICC identified by #EID has been initially provisioned on the SM-SR-UT using the #EIS_RPS

5.3.8.2.1 TC.PROC.UCP.1: UpdateConnectivityParameters

Test Purpose

To ensure the Connectivity Parameters can be updated by the SM-SR when the SM-DP requests it.

Test Environment


SM-DP-S
SM-SR-UT
Device
eUICC
ES3-UpdateConnectivityParameters(ES8-UpdateConnectivityParameters)

Referenced Requirements

- PROC_REQ19
- PM_REQ21

Initial Conditions

- None

5.3.8.2.1.1 Test Sequence N°1 - Nominal Case: Update SMS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	SM-DP-S → SM-SR-UT	`SEND_REQ(`
		`ES3-UpdateConnectivityParameters,`
		`#EID_RPS,`
		`#ICCID_RPS,`
		`SCP03_SCRIPT(`
		`#DEFAULT_ISD_P_SCP03_KVN,`
		`[STORE_SMS_PARAM_MNO2]))`
		see Note 1		

2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → SM-DP-S	Send the `ES3-UpdateConnectivityParameters` response	The `Status` is equal to `#SUCCESS`	PROC_REQ19, PM_REQ21
Note 1: The C-APDUs generated by the method `SCP03_SCRIPT` shall be set into the RPS element `<connectivityParameters>`				

5.3.8.2.1.2 Test Sequence N°2 - Nominal Case: Update CAT_TP Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-UpdateConnectivityParameters, #EID_RPS, #ICCID_RPS, SCP03_SCRIPT(#DEFAULT_ISD_P_SCP03_KVN, [STORE_CATTP_PARAM_MNO2])) see Note 1		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → SM-DP-S	Send the ES3-UpdateConnectivityParameters response	The Status is equal to #SUCCESS	PROC_REQ19, PM_REQ21
Note 1: The C-APDUs generated by the method SCP03_SCRIPT shall be set into the RPS element <connectivityParameters>				

5.3.8.2.1.3 Test Sequence N°3 - Nominal Case: Update HTTPS Parameters

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-UpdateConnectivityParameters, #EID_RPS, #ICCID_RPS, SCP03_SCRIPT(#DEFAULT_ISD_P_SCP03_KVN, [STORE_HTTPS_PARAM_MNO2])) see Note 1		
2	*Wait until a response is received (the SM-SR-UT treatment may take several minutes)*			
3	SM-SR-UT → SM-DP-S	Send the ES3-UpdateConnectivityParameters response	The Status is equal to #SUCCESS	PROC_REQ19, PM_REQ21

Step	Direction	Sequence / Description	Expected result	REQ
Note 1: The C-APDUs generated by the method `SCP03_SCRIPT` *shall be set into the RPS element* `<connectivityParameters>`				

Annex A Reference Applications

The following Annex provides clarification on the applications to be used to execute some test cases.

A.1 Applet1

A.1.1 Description

This applet defines an application which implements `uicc.toolkit.ToolkitInterface`. The event *`EVENT_FORMATTED_SMS_PP_ENV`* is set in the Toolkit Registry entry of the applet.

A.1.2 AID

- Executable Load File AID: A0 00 00 05 59 10 10 01
- Executable Module AID: A0 00 00 05 59 10 10 01 11 22 33

A.1.3 Source Code (Java Card)

```java
package com.gsma.euicc.test.applet1;

import javacard.framework.AID;
import javacard.framework.APDU;
import javacard.framework.Applet;
import javacard.framework.ISOException;
import javacard.framework.Shareable;
import uicc.toolkit.ToolkitException;
import uicc.toolkit.ToolkitInterface;
import uicc.toolkit.ToolkitRegistrySystem;
import uicc.usim.toolkit.ToolkitConstants;

/**
 * GSMA Test Toolkit Applet1
 */
public class Applet1 extends Applet implements ToolkitConstants, ToolkitInterface {
        /**
         * Default Applet constructor
         */
        public Applet1() {
                // nothing to do
        }

        /**
         * Create an instance of the applet, the Java Card runtime environment will
         * call this static method first.
         * @param bArray the array containing installation parameters
         * @param bOffset the starting offset in bArray
         * @param bLength the length in bytes of the parameter data in bArray
         * @throws ISOException if the install method failed
         * @see javacard.framework.Applet
         */
        public static void install(byte[] bArray, short bOffset, byte bLength)
        throws ISOException {
                Applet1 applet1 = new Applet1();
```

```
            byte aidLen = bArray[bOffset];
            if (aidLen == (byte) 0) {
                applet1.register();
            } else {
                applet1.register(bArray, (short) (bOffset + 1), aidLen);
            }
            applet1.registerEvent();
        }

        /*
         * (non-Javadoc)
         * @see Applet#process(javacard.framework.APDU)
         */
        public void process(APDU apdu) throws ISOException {
            // nothing to do
        }

        /*
         * (non-Javadoc)
         * @see Applet#getShareableInterfaceObject(javacard.framework.AID, byte)
         */
        public Shareable getShareableInterfaceObject(AID clientAID, byte param) {
            if ((param == (byte) 0x01) && (clientAID == null)) {
                return ((Shareable) this);
            }
            return null;
        }

        /*
         * (non-Javadoc)
         * @see uicc.toolkit.ToolkitInterface#processToolkit(short)
         */
        public void processToolkit(short event) throws ToolkitException {
            // nothing to do
        }

        /**
         * Registration to the event EVENT_FORMATTED_SMS_PP_ENV
         */
        private void registerEvent() {
            ToolkitRegistrySystem.getEntry()
                .setEvent(EVENT_FORMATTED_SMS_PP_ENV);
        }

}
```

A.2 Applet2

A.2.1 Description

This applet is a clone of Applet1 except that the package AID and the applet AID are different.

A.2.2 AID

- Executable Load File AID: A0 00 00 05 59 10 10 02
- Executable Module AID: A0 00 00 05 59 10 10 02 11 22 33

A.2.3 Source Code (Java Card)

This source code is exactly the same as the Applet1 defined in Annex A.1 except that the package name shall be `com.gsma.euicc.test.applet2`.

A.3 Applet3

A.3.1 Description

This applet defines a “simple” application.

A.3.2 AID

- Executable Load File AID: A0 00 00 05 59 10 10 03
- Executable Module AID: A0 00 00 05 59 10 10 03 44 55 66

A.3.3 Source Code (Java Card)

```
package com.gsma.euicc.test.applet3;

import javacard.framework.APDU;
import javacard.framework.Applet;
import javacard.framework.ISOException;

/**
 * GSMA Test Applet3
 */
public class Applet3 extends Applet {
       /**
        * Default Applet constructor
        */
       public Applet3() {
              // nothing to do
       }

       /**
        * Create an instance of the applet, the Java Card runtime environment will
        * call this static method first.
        * @param bArray the array containing installation parameters
        * @param bOffset the starting offset in bArray
        * @param bLength the length in bytes of the parameter data in bArray
        * @throws ISOException if the install method failed
        * @see javacard.framework.Applet
        */
       public static void install(byte[] bArray, short bOffset, byte bLength)
       throws ISOException {
              Applet3 applet3 = new Applet3();
              byte aidLen = bArray[bOffset];
              if (aidLen == (byte) 0) {
                     applet3.register();
              } else {
                     applet3.register(bArray, (short) (bOffset + 1), aidLen);
              }
       }

       /*
        * (non-Javadoc)
        * @see Applet#process(javacard.framework.APDU)
        */
       public void process(APDU apdu) throws ISOException {
              // nothing to do
       }

}
```

Annex B Constants

B.1 Hexadecimal Constants

Here are the hexadecimal constants values used in this document:

Constant name	Value in hexadecimal string
ADMIN_HOST	6C 6F 63 61 6C 68 6F 73 74
ADMIN_URI	2F 67 73 6D 61 2F 61 64 6D 69 6E 61 67 65 6E 74
AGENT_ID	2F 2F 73 65 2D 69 64 2F 65 69 64 2F #EID 3B 2F 2F 61 61 2D 69 64 2F 61 69 64 2F 41 30 30 30 30 30 30 35 35 39 2F 31 30 31 30 46 46 46 46 46 46 46 46 38 39 30 30 30 30 30 31 30 30
BAD_SCP03_KVN	35
BAD_SPI	12 29
BAD_TOKEN	01 02 03
BEARER_DESCRIPTION	02 00 00 03 00 00 02
BUFFER_SIZE	05 78
CASD_AID	A0 00 00 01 51 53 50 43 41 53 44 00
CAT_TP_PORT	04 00
DEST_ADDR	05 85 02 82 F2
DIALING_NUMBER	33 86 99 42 11 F0
ECASD_AID	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 02 00
ECASD_TAR	00 00 02
FIRST_SCRIPT	01
HOST_ID	47 53 4D 41 5F 48 4F 53 54 5F 49 44
ICCID1	89 01 99 99 00 00 44 77 78 78
ICCID2	89 01 99 99 00 00 44 77 78 79
IP_VALUE	7F 00 00 01
ISD_P_AID1	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 see Note 1
ISD_P_AID2	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 11 00
ISD_P_AID3	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 12 00
ISD_P_ATTRIBUTE	53
ISD_P_MOD_AID	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 0E 00
ISD_P_PIX_PREFIX	10 10 FF FF FF FF 89
ISD_P_PKG_AID	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 0D 00
ISD_P_PROV_ID	47 53 4D 41
ISD_P_RID	A0 00 00 05 59
ISD_P_SDIN	49 53 44 50 53 44 49 4E
ISD_P_SIN	49 53 44 50
ISD_P_TAR1	00 00 10 see Note 1
ISD_R_AID	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 01 00

Constant name	Value in hexadecimal string
ISD_R_TAR	`00 00 01`
KEY	`11 22 33 44 55 66 77 88 99 10 11 12 13 14 15 16`
KEY_USAGE	`00 80`
LAST_SCRIPT	`03`
LOGIN	`04 6C 6F 67 69 6E`
MEMORY_QUOTA	`00 00 20 00`

MNO_AGENT_ID	`2F 2F 73 65 2D 69 64 2F 65 69 64 2F #EID 3B`
	`2F 2F 61 61 2D 69 64 2F 61 69 64 2F #MNO_SD_AID`

NAN_VALUE	`09 47 53 4D 41 65 55 49 43 43`
PWD	`04 70 61 73 73 77 6F 72 64`
RESERVED_ISD_P_AID	`A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 0F 00`
SC3_DR	`0B`
SC3_DR_HOST	`0F`
SC3_NO_DR	`09`
SCP03_KVN	`30`
SCP80_NEW_KVN	`0E` see Note 2
SPI_VALUE	`16 39`
SPI_NOTIF	`02 00`
SUB_SCRIPT	`02`
TCP_PORT	`1F 41`
TOKEN_ID	`01`
TON_NPI	`91`
UDP_PORT	`1F 40`
VIRTUAL_EID	`89 00 10 12 01 23 41 23 40 12 34 56 78 90 12 24`
VIRTUAL_EID2	`89 00 15 67 01 02 03 04 05 06 07 08 09 10 11 52`
VIRTUAL_SDIN	`00 00 00 00 01 02 03 04 05 06 07 08`
VIRTUAL_SIN	`01 02 03 04`
Note 1: Shall be different from the Profiles already installed on the eUICC. This constant depends on the eUICC *Note 2: Shall not be initialized by default on the eUICC (different than #SCP80_KVN)*	

Table 8: Hexadecimal Constants

B.2 ASCII Constants

Here are the ASCII constants values used in this document:

Constant name	Value in ASCII
CONTENT_TYPE	`Content-Type: application/vnd.globalplatform.card-content-mgt-response;version=1.0`
EUM_S_ID	`10.11.12`
FAILED	`Failed`

Constant name	Value in ASCII
HOST	`Host: localhost`
HTTP_CODE_200	`HTTP/1.1 200`
HTTP_CODE_204	`HTTP/1.1 204`
IMSI1	`234101943787656`
IMSI2	`234101943787657`
IMSI3	`234101943787658`
MNO1_S_ID	`1.2.3`
MNO2_S_ID	`11.22.33`
MSISDN1	`447112233445`
MSISDN2	`447112233446`
MSISDN3	`447112233447`
POST_URI	`POST /gsma/adminagent HTTP/1.1`
POST_URI_NOTIF	`POST /gsma/adminagent?msg=#NOTIF_PROFILE_CHANGE HTTP/1.1`
POST_URI_NOTIF2	`POST /gsma/adminagent?msg=#NOTIF_PROFILE_CHANGE2 HTTP/1.1`
PROFILE1_TYPE	`GENERIC PROFILE1 3G`
PROFILE2_TYPE	`GENERIC PROFILE2 3G`
PSK_ID	`8001028110#EID4F10#ISD_R_AID8201#SCP81_KEY_ID8301#SCP81_KVN` see Note 2
RC_ALREADY_REGISTER	`1.3`
RC_ALREADY_USED	`3.3`
RC_COND_PARAM	`2.3`
RC_COND_USED	`3`
RC_EXECUTION_ERROR	`4.2`
RC_EXPIRED	`6.3`
RC_ID_UNKNOWN	`1.1`
RC_INACCESSIBLE	`5.1`
RC_INVALID_DATA	`1.5`
RC_INVALID_DEST	`3.4`
RC_INVALID_SIGN	`1.4`
RC_MEMORY	`4.8`
RC_NOT_ALLOWED	`1.2`
RC_REFUSED	`3.8`
RC_UNKNOWN	`3.9`
SC_ECASD	`8.5.2`
SC_EID	`8.1.1`
SC_EIS	`8.6`
SC_EUICC	`8.1`
SC_FUN_PROV	`1.2`
SC_FUNCTION	`1.6`
SC_SD_AID	`8.3.1`
SC_ISDP	`8.3`
SC_ISDR	`8.4`

Constant name	Value in ASCII
SC_POL1	`8.2.2`
SC_POL2	`8.2.3`
SC_PROFILE_ICCID	`8.2.1`
SC_PROFILE	`8.2`
SC_SM_SR	`8.7`
SC_SR_CERTIF	`8.5.3`
SC_SUB_ADDR	`8.2.6`
SM_DP_S_ID	`4.5.6`
SM_SR_S_ID	`7.8.9`
SUCCESS	`Executed-Success`
TRANSFER_ENCODING	`Transfer-Encoding: chunked`
UNKNOWN_SM_SR_ID	`8888.9999.1111` see Note 1
WARNING	`Executed-WithWarning`
X_ADMIN_FROM_ISD_R	`X-Admin-From: //se-id/eid/#EID;//aa-id/aid/A000000559/1010FFFFFFFF8900000100`
X_ADMIN_FROM_MNO	`X-Admin-From: //se-id/eid/#EID;//aa-id/aid/#MNO_SD_AID`
X_ADMIN_NEXT_URI	`X-Admin-Next-URI: /gsma/adminagent`
X_ADMIN_PROTOCOL	`X-Admin-Protocol: globalplatform-remote-admin/1.0`
X_ADMIN_STATUS_OK	`X-Admin-Script-Status: ok`

Note 1: This value shall be unknown to all platforms under test.

Note 2: This Pre-Shared Key identity string shall be configured by default in the ISD-R.

Table 9: ASCII Constants

B.3 eUICC Settings

Here are the different settings that shall be given by the eUICC Manufacturer to execute the test cases defined in this document.

eUICC setting name	Description
CARD_RECOGNITION_DATA	Value of the TLV ‘66’ - Card recognition data.
DEFAULT_ISD_P_AID	The AID of the default ISD-P pre-installed on the eUICC (this ISD-P shall be Enabled).
DEFAULT_ISD_P_SCP03_KDEK	The SCP03 DEK key of the default ISD-P pre-installed on the eUICC.
DEFAULT_ISD_P_SCP03_KENC	The SCP03 ENC key of the default ISD-P pre-installed on the eUICC.
DEFAULT_ISD_P_SCP03_KMAC	The SCP03 MAC key of the default ISD-P pre-installed on the eUICC.
DEFAULT_ISD_P_SCP03_KVN	The SCP03 KVN of the default ISD-P pre-installed on the eUICC.
DEFAULT_ISD_P_TAR	The TAR of the default ISD-P pre-installed on the eUICC.
ECASD_CERTIFICATE	Value of the TLV ‘7F21’ - ECASD certificate (i.e. CERT.ECASD.ECKA).
CASD_CERTIFICATE	Value of the TLV ‘7F21’ - CASD certificate (of the default Enabled Profile).
EID	Content of the TLV ‘5A’ available on the ECASD.
EUM_PK_ECDSA	Public key of the EUM used for ECDSA.

eUICC setting name	Description
EUM_PK_CA_AUT	Public key of the EUM used to verify the MNO CASD certificate.
ISD_R_SIN	Content of the TLV '42' available on the ISD-R.
ISD_R_SDIN	Content of the TLV '45' available on the ISD-R.
PROFILE_PACKAGE	A Profile Package that contains all Profile Elements allowing the testing of the download and the network attachment processes. This Profile should follow the description defined in Annex B.7.
MNO_PSK_ID	The Pre-Shared Key identity string related to the SCP81 keyset initialized on the MNO-SD. (optional: depends if O_MNO_HTTPS is supported)
MNO_SCP80_AUTH_KEY	The value of the SCP80 message authentication key initialized on the default MNO-SD. (key identifier 02)
MNO_SCP80_DATA_ENC_KEY	The value of the SCP80 data encryption key initialized on the default MNO-SD. (key identifier 03)
MNO_SCP80_ENC_KEY	The value of the SCP80 encryption key initialized on the default MNO-SD. (key identifier 01)
MNO_SCP80_KVN	The key version number of the SCP80 keyset initialized on the default MNO-SD.
MNO_SCP81_KEY_ID	The key identifier of the PSK in the SCP81 keyset initialized on the MNO-SD. (optional: depends if O_MNO_HTTPS is supported)
MNO_SCP81_KVN	The key version number of the SCP81 keyset initialized on the MNO-SD. (optional: depends if O_MNO_HTTPS is supported)
MNO_SCP81_PSK	The value of the Pre-Shared Key initialized on the MNO-SD. (optional: depends if O_MNO_HTTPS is supported)
MNO_SD_AID	The MNO ISD AID of the default Profile pre-installed on the eUICC.
MNO_TAR	The TAR of the default MNO-SD (should be 'B2 01 00').
PK_ECASD_ECKA	Public Key of the ECASD used for ECKA (i.e. PK.ECASD.ECKA).
SCP80_DATA_ENC_KEY	The value of the SCP80 data encryption key initialized on the ISD-R. (key identifier 03)
SCP80_ENC_KEY	The value of the SCP80 encryption key initialized on the ISD-R. (key identifier 01)
SCP80_KVN	The key version number of the SCP80 keyset initialized on the ISD-R.
SCP80_AUTH_KEY	The value of the SCP80 message authentication key initialized on the ISD-R. (key identifier 02)
SCP81_KEY_ID	The key identifier of the PSK in the SCP81 keyset initialized on the ISD-R. (optional: depends if O_HTTPS is supported)
SCP81_KVN	The key version number of the SCP81 keyset initialized on the ISD-R. (optional: depends if O_HTTPS is supported)
SCP81_PSK	The value of the Pre-Shared Key initialized on the ISD-R. (optional: depends if O_HTTPS is supported)

Table 10: eUICC Settings

B.4 Platforms Settings

Here are the different platforms' settings that shall be used to execute the test cases defined in this document. The corresponding values shall be given either by the test tool provider, the platform under test or the CI.

Platform setting name	Description
ECASD_BAD_SIGN_CERT	A certificate CERT.ECASD.ECKA with an invalid signature of a simulated eUICC. The TLV '7F21' shall contain: 93 01 09 42 04 #VIRTUAL_SIN 5F 20 10 #VIRTUAL_EID 95 02 00 80 5F 25 04 20 00 01 01 5F 24 04 21 45 01 01 45 0C #VIRTUAL_SDIN 73 09 C0 01 01 C1 01 01 C2 01 01 7F 49 {L} #PK_ECASD_S_ECKA 5F 37 {L} {SIGNATURE} This signature shall not be generated using the #EUM_S_SK_ECDSA. see Note 1
EUM_S_ACCESSPOINT	The EUM-S access point allowing SM-SR-UT to communicate with a EUM simulator. see Note 1
EUM_S_CERT_ID_ECDSA	The certificate ID of the EUM-S used for ECDSA. see Note 1
EUM_S_PK_ECDSA	Public key of the EUM-S used for ECDSA. see Note 1
EUM_S_SK_ECDSA	Private key of the EUM-S used for ECDSA. see Note 1
EXPIRED_ECASD_CERT	An expired certificate CERT.ECASD.ECKA of a simulated eUICC. The TLV '7F21' shall contain: 93 01 09 42 04 #VIRTUAL_SIN 5F 20 10 #VIRTUAL_EID 95 02 00 80 5F 25 04 20 00 01 01 5F 24 04 20 00 02 02 45 0C #VIRTUAL_SDIN 73 09 C0 01 01 C1 01 01 C2 01 01 7F 49 {L} #PK_ECASD_S_ECKA 5F 37 {L} {SIGNATURE} This signature shall be generated using the #EUM_S_SK_ECDSA. see Note 1
EXPIRED_SM_SR_CERTIFICATE	An expired certificate CERT.SR.ECDSA of a simulated SM-SR. The TLV '7F21' shall contain: 93 01 01 42 01 01 5F 20 01 01 95 01 88 5F 24 04 20 00 01 01 73 03 C8 01 02 7F 49 {L} #SM_PK_ECDSA 5F 37 {L} {SIGNATURE} This signature shall be generated using the #SK_CI_ECDSA. This TLV '7F21' shall be part of the DGI '7F21'. see Note 1
INVALID_SM_DP_CERTIFICATE	An invalid certificate CERT.DP.ECDSA of a simulated SM-DP (TLV '7F21'). The #SK_CI_ECDSA shall not be used to generate the signature. The content of the TLV is the same as #VALID_SM_DP_CERTIFICATE.

Platform setting name	Description
	see Note 1
INVALID_SM_SR_CERTIFICATE	An invalid certificate CERT.DP.ECDSA of a simulated SM-DP (TLV ‘7F21’). The #SK_CI_ECDSA shall not be used to generate the signature. The content of the TLV is the same as #VALID_SM_SR_CERTIFICATE. see Note 1
MNO1_S_ACCESSPOINT	The MNO1-S access point allowing platforms under test to communicate with a MNO simulator. see Note 1
MNO2_S_ACCESSPOINT	The MNO2-S access point allowing platforms under test to communicate with a MNO simulator. see Note 1
PK_CI_ECDSA	The CI public key used for verifying the SM-SR and SM-DP certificates (i.e. PK.CI.ECDSA). see Note 3
PK_CI_ECDSA_PARAM	The CI public key parameter reference used for verifying the SM-SR and SM-DP certificates (i.e. part of the PK.CI.ECDSA). see Note 3
PK_ECASD_S_ECKA	Public Key of a virtual ECASD used for ECKA (i.e. PK.ECASD.ECKA). see Note 1
SK_CI_ECDSA	The CI private key used for signing data to generate the SM-SR and the SM-DP certificates (i.e. SK.CI.ECDSA). see Note 3
SM_DP_ACCESSPOINT	The SM-DP-UT access point allowing communication. This value depends on the transport protocol used by the SM-DP-UT. see Note 2
SM_DP_ID	The SM-DP-UT identifier. see Note 2
SM_DP_S_ACCESSPOINT	The SM-SR-S access point allowing platforms under test to communicate with a SM-DP simulator. see Note 1
SM_EPK_ECKA	Ephemeral Public Key of a simulated SM-SR (i.e. ePK.SR.ECKA), SM-DP (i.e. ePK.DP.ECKA) or MNO used for ECKA. see Note 1
SM_ESK_ECKA	Ephemeral Private Key of a simulated SM-SR (i.e. eSK.SR.ECKA), SM-DP (i.e. eSK.DP.ECKA) or MNO used for ECKA. see Note 1
SM_PK_ECDSA	Public Key of a simulated SM-SR (i.e. PK.SR.ECDSA) or SM-DP (i.e. PK.DP.ECDSA) for verifying signatures. see Note 1
SM_SK_ECDSA	Private Key of a simulated SM-SR (i.e. SK.SR.ECDSA) or SM-DP (i.e. SK.DP.ECDSA) for creating signatures. see Note 1
SM_SR_ACCESSPOINT	The SM-SR-UT access point allowing communication. This value depends on the transport protocol used by the SM-SR-UT. see Note 2
SM_SR_ID	The SM-SR-UT identifier. see Note 2
SM_SR_S_ACCESSPOINT	The SM-SR-S access point allowing platforms under test to communicate with a SM-SR simulator. see Note 1
VALID_SM_DP_CERTIFICATE	A valid certificate CERT.DP.ECDSA of a simulated SM-DP. The TLV ‘7F21’ shall contain:

Platform setting name	Description
	93 01 02 42 01 02 5F 20 01 02 95 01 88 5F 24 04 21 45 01 01 73 03 C8 01 01 7F 49 {L} #SM_PK_ECDSA 5F 37 {L} {SIGNATURE} This signature shall be generated using the #SK_CI_ECDSA. see Note 1
VALID_SM_SR_CERTIFICATE	A valid certificate CERT.SR.ECDSA of a simulated SM-SR. The TLV ‘7F21’ shall contain: 93 01 01 42 01 01 5F 20 01 01 95 01 88 5F 24 04 21 45 01 01 73 03 C8 01 02 7F 49 {L} #SM_PK_ECDSA 5F 37 {L} {SIGNATURE} This signature shall be generated using the #SK_CI_ECDSA. see Note 1
VIRTUAL_ECASD_CERT	A valid certificate CERT.ECASD.ECKA of a simulated eUICC. The TLV ‘7F21’ shall contain: 93 01 09 42 04 #VIRTUAL_SIN 5F 20 10 #VIRTUAL_EID 95 02 00 80 5F 25 04 20 00 01 01 5F 24 04 21 45 01 01 45 0C #VIRTUAL_SDIN 73 09 C0 01 01 C1 01 01 C2 01 01 7F 49 {L} #PK_ECASD_S_ECKA 5F 37 {L} {SIGNATURE} This signature shall be generated using the #EUM_S_SK_ECDSA. see Note 1
Note 1: Shall be generated by the test tool *Note 2: Shall be given by the platform under test* *Note 3: Shall be given by the CI*	

Table 11: Platforms Settings

B.5 RPS Elements

Here are the different RPS elements that shall be used to execute the test cases defined in this document.

RPS element name	Value
AUDIT_OPERATION_RPS	<Record> #EID_RPS #SM_SR_UT_ID_RPS <OperationDate>{CURRENT_DATE}</OperationDate> <OperationType>0500</OperationType> <RequesterId>#MNO2_S_ID</RequesterId>

RPS element name	Value

	`<OperationExecutionStatus>`
	`#SUCCESS`
	`</OperationExecutionStatus>`
	`<Isd-p-aid>#DEFAULT_ISD_P_AID</Isd-p-aid>`
	`#ICCID_RPS`
	`</Record>`

BIG_MEM_RPS	`<RequiredMemory>9999999</RequiredMemory>`

CATTP_CAP_RPS	`<CattpSupport>TRUE</CattpSupport>`
	`<CattpVersion>6.13.0</CattpVersion>`
	`<HttpSupport>FALSE</HttpSupport>`
	`<SecurePacketVersion>12.1.0</SecurePacketVersion>`
	`<RemoteProvisioningVersion>3.1.0</RemoteProvisioningVersion>`

CON_PARAM_RPS	`<connectivityParameters>`
	`222F80E288002A3A0727A1253507#BEARER_DESCRIPTION4709#NAN_VALUE0D05#LOGIN0D08#PWD`
	`</connectivityParameters>`
	
	see Note 6

CUR_SR_S_ID_RPS	`<CurrentSmSrid>#SM_SR_S_ID</CurrentSmSrid>`
CUR_SR_ID_RPS	`<CurrentSmSrid>#SM_SR_ID</CurrentSmSrid>`
DATA_RPS	`<Data>220E8050300008010203040102030400</Data>` see Note 6
DEFAULT_ISDP_RPS	`<Isd-p-aid>#DEFAULT_ISD_P_AID</Isd-p-aid>`

ECASD_BADKEY_RPS	`<Aid>#ECASD_AID</Aid>`
	`<Tar>#ECASD_TAR</Tar>`
	`<Sin>#VIRTUAL_SIN</Sin>`
	`<Sdin>#VIRTUAL_SDIN</Sdin>`
	`<Role>ECASD</Role>`
	`<Keyset>`
	`<Version>74</Version>`
	`<Type>CA</Type>`
	`<Certificate>`
	`<Index>02</Index>`
	`<CAId>#VIRTUAL_SDIN</CAId>`
	`<Value>#VIRTUAL_ECASD_CERT</Value>`
	`</Certificate>`
	`<Key kcv=””>`
	`<Index>01</Index>`
	`<KeyComponent`
	`type=”B0”`
	`value=”#SK_CI_ECDSA”>`
	`</KeyComponent>`
	`</Key>`
	`<Key kcv=””>`
	`<Index>01</Index>`
	`<KeyComponent`
	`type=”F0”`
	`value=”#PK_CI_ECDSA_PARAM”>`
	`</KeyComponent>`
	`</Key>`
	`</Keyset>`

RPS element name	Value
	see Note 7
ECASD_BADSIGN_RPS	<Aid>#ECASD_AID</Aid> <Tar>#ECASD_TAR</Tar> <Sin>#VIRTUAL_SIN</Sin> <Sdin>#VIRTUAL_SDIN</Sdin> <Role>ECASD</Role> <Keyset> <Version>74</Version> <Type>CA</Type> <Certificate> <Index>02</Index> <CAId>#VIRTUAL_SDIN</CAId> <Value>#ECASD_BAD_SIGN_CERT</Value> </Certificate> <Key kcv=””> <Index>01</Index> <KeyComponent type=”B0” value=”#PK_CI_ECDSA”> </KeyComponent> </Key> <Key kcv=””> <Index>01</Index> <KeyComponent type=”F0” value=”#PK_CI_ECDSA_PARAM”> </KeyComponent> </Key> </Keyset>
ECASD_RPS	<Aid>#ECASD_AID</Aid> <Tar>#ECASD_TAR</Tar> <Sin>#VIRTUAL_SIN</Sin> <Sdin>#VIRTUAL_SDIN</Sdin> <Role>ECASD</Role> <Keyset> <Version>74</Version> <Type>CA</Type> <Certificate> <Index>02</Index> <CAId>#VIRTUAL_SDIN</CAId> <Value>#VIRTUAL_ECASD_CERT</Value> </Certificate> <Key kcv=””> <Index>01</Index> <KeyComponent type=”B0” value=”#PK_CI_ECDSA”> </KeyComponent> </Key> <Key kcv=””> <Index>01</Index> <KeyComponent type=”F0” value=”#PK_CI_ECDSA_PARAM”> </KeyComponent>

RPS element name	Value
	</Key> </Keyset>
ECC_KEY_LENGTH_RPS	<ECCKeyLength>ECC-256</ECCKeyLength>
EID_RPS	<Eid>#EID</Eid>
EIS_BADCASDKEY_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid>#ISD_P_MOD_AID</Isd-p-module-aid> <Ecasd>#ECASD_BADKEY_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 1
EIS_BADCASDSIGN_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid>#ISD_P_MOD_AID</Isd-p-module-aid> <Ecasd>#ECASD_BADSIGN_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS

RPS element name	Value
	<ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 1
EIS_BADEUMSIGN_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid> #ISD_P_MOD_AID </Isd-p-module-aid> #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> <Ecasd>#ECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 2
EIS_ES1_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID

RPS element name	Value
	</Isd-p-loadfile-aid> <Isd-p-module-aid> #ISD_P_MOD_AID </Isd-p-module-aid> #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> <Ecasd>#ECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 1
EIS_ES2_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Ecasd>#ECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 1
EIS_ES3_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000

RPS element name	Value

	`</AvailableMemoryForProfiles>`
	`{SM_SR_ID_RPS}`
	`<Isd-p-loadfile-aid>`
	`#ISD_P_PKG_AID`
	`</Isd-p-loadfile-aid>`
	`<Isd-p-module-aid>#ISD_P_MOD_AID</Isd-p-module-aid>`
	`<Isdr-r>#ISD_R_RPS</Isdr-r>`
	`<Ecasd>#ECASD_RPS</Ecasd>`
	`<EuiccCapabilities>`
	`#FULL_CAP_RPS`
	`</EuiccCapabilities>`
	`<EumSignature`
	`xmlns:ds="http://www.w3.org/2000/09/xmldsig">`
	`#SIGNED_INFO_RPS`
	`<ds:SignatureValue>`
	`{SIGNATURE}`
	`</ds:SignatureValue>`
	`#KEY_INFO_RPS`
	`</EnumSignature>`
	`</Eis>`
	
	see Note 1 and Note 8

EIS_ES4_RPS	`<Eis>`
	`#VIRTUAL_EID_RPS`
	`<Eum-Id>#EUM_S_ID</Eum-Id>`
	`<ProductionDate>2014-01-01T09:30:47Z</ProductionDate>`
	`<PlatformType>JavaCard Operating System</PlatformType>`
	`<PlatformVersion>3.0.1</PlatformVersion>`
	`<RemainingMemory>800000</RemainingMemory>`
	`<AvailableMemoryForProfiles>`
	`750000`
	`</AvailableMemoryForProfiles>`
	`{SM_SR_ID_RPS}`
	`<Isd-p-loadfile-aid>`
	`#ISD_P_PKG_AID`
	`</Isd-p-loadfile-aid>`
	`<Isd-p-module-aid>`
	`#ISD_P_MOD_AID`
	`</Isd-p-module-aid>`
	`#PROFILE1_RPS`
	`<EuiccCapabilities>`
	`#FULL_CAP_RPS`
	`</EuiccCapabilities>`
	`<EumSignature`
	`xmlns:ds="http://www.w3.org/2000/09/xmldsig">`
	`#SIGNED_INFO_RPS`
	`<ds:SignatureValue>`
	`{SIGNATURE}`
	`</ds:SignatureValue>`
	`#KEY_INFO_RPS`
	`</EnumSignature>`
	`</Eis>`
	
	see Note 1

EIS_EXPIREDCASD_RPS	`<Eis>`
	`#VIRTUAL_EID_RPS`
	`<Eum-Id>#EUM_S_ID</Eum-Id>`

RPS element name	Value
	<ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid> #ISD_P_MOD_AID </Isd-p-module-aid> #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> <Ecasd>#EXPIREDECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 1
EIS_SIGNED_RPS	#VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <Isd-p-loadfile-aid>#ISD_P_PKG_AID</Isd-p-loadfile-aid> <Isd-p-module-aid>#ISD_P_MOD_AID</Isd-p-module-aid> <Ecasd>#ECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS #KEY_INFO_RPS </EnumSignature>
EIS2_BADCASDKEY_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000

RPS element name	Value
	</AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid> #ISD_P_MOD_AID </Isd-p-module-aid> #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> <Ecasd>#ECASD_BADKEY_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 1
EIS2_BADCASDSIGN_RPS	<Eis> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid> #ISD_P_MOD_AID </Isd-p-module-aid> #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> <Ecasd>#ECASD_BADSIGN_RPS </Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS

RPS element name	Value
	</EnumSignature> </Eis> see Note 1

EIS2_ES1_RPS	<Eis>
	        #VIRTUAL_EID_RPS
	        <Eum-Id>#EUM_S_ID</Eum-Id>
	        <ProductionDate>2014-01-01T09:30:47Z</ProductionDate>
	        <PlatformType>JavaCard Operating System</PlatformType>
	        <PlatformVersion>3.0.1</PlatformVersion>
	        <RemainingMemory>800000</RemainingMemory>
	        <AvailableMemoryForProfiles>
	                750000
	        </AvailableMemoryForProfiles>
	        {SM_SR_ID_RPS}
	        <Isd-p-loadfile-aid>
	                #ISD_P_PKG_AID
	        </Isd-p-loadfile-aid>
	        <Isd-p-module-aid>
	                #ISD_P_MOD_AID
	        </Isd-p-module-aid>
	        #PROFILE2_RPS
	        <Isdr-r>#ISD_R_RPS</Isdr-r>
	        <Ecasd>#ECASD_RPS</Ecasd>
	        <EuiccCapabilities>
	                #FULL_CAP_RPS
	        </EuiccCapabilities>
	        <EumSignature
	                xmlns:ds="http://www.w3.org/2000/09/xmldsig">
	                #SIGNED_INFO_RPS
	                <ds:SignatureValue>
	                        {SIGNATURE}
	                </ds:SignatureValue>
	                #KEY_INFO_RPS
	        </EnumSignature>
	</Eis>
	
	see Note 1

EIS2_SIGNED_RPS	#VIRTUAL_EID2_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <Isd-p-loadfile-aid>#ISD_P_PKG_AID</Isd-p-loadfile-aid> <Isd-p-module-aid>#ISD_P_MOD_AID</Isd-p-module-aid> <Ecasd>#ECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS #KEY_INFO_RPS </EnumSignature>
EIS3_ES1_RPS	<Eis> #VIRTUAL_EID2_RPS

RPS element name	Value
	<Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>JavaCard Operating System</PlatformType> <PlatformVersion>3.0.1</PlatformVersion> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid> #ISD_P_MOD_AID </Isd-p-module-aid> #PROFILE1_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> <Ecasd>#ECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> </Eis> see Note 3
EP_FALSE_RPS	<EnableProfile>FALSE</EnableProfile>
EP_TRUE_RPS	<EnableProfile>TRUE</EnableProfile>
EPHEMERAL_PK_RPS	<EphemeralPublicKey>#SM_EPK_ECKA</EphemeralPublicKey>
EUICC_RESP1_RPS	<EuiccResponseData>[R_AB_6985]</EuiccResponseData>
EXPIREDECASD_RPS	<Aid>#ECASD_AID</Aid> <Tar>#ECASD_TAR</Tar> <Sin>#VIRTUAL_SIN</Sin> <Sdin>#VIRTUAL_SDIN</Sdin> <Role>ECASD</Role> <Keyset> <Version>74</Version> <Type>CA</Type> <Certificate> <Index>02</Index> <CAId>#VIRTUAL_SDIN</CAId> <Value>#EXPIRED_ECASD_CERT</Value> </Certificate> </Keyset>
FULL_CAP_RPS	<CattpSupport>TRUE</CattpSupport> <CattpVersion>6.13.0</CattpVersion> <HttpSupport>TRUE</HttpSupport> <HttpVersion>1.1.3</HttpVersion> <SecurePacketVersion>12.1.0</SecurePacketVersion>

RPS element name	Value
	`<RemoteProvisioningVersion>3.1.0</RemoteProvisioningVersion>`
HOST_ID_RPS	`<HostId>#HOST_ID</HostId>`

HTTPS_CAP_RPS	`<CattpSupport>FALSE</CattpSupport>`
	`<HttpSupport>TRUE</HttpSupport>`
	`<HttpVersion>1.1.3</HttpVersion>`
	`<SecurePacketVersion>12.1.0</SecurePacketVersion>`
	`<RemoteProvisioningVersion>3.1.0</RemoteProvisioningVersion>`

ICCID_RPS	`<Iccid>#ICCID</Iccid>`
ICCID1_RPS	`<Iccid>#ICCID1</Iccid>`
ICCID2_RPS	`<Iccid>#ICCID2</Iccid>`
INIT_SEQ_COUNTER_RPS	`<InitialSequenceCounter>0</InitialSequenceCounter>`

INVALID_EIS_RPS	`<Eis>`
	`    #VIRTUAL_EID_RPS`
	`    <Eum-Id>#EUM_S_ID</Eum-Id>`
	`    <ProductionDate>2014-01-01T09:30:47Z</ProductionDate>`
	`    <PlatformType>JavaCard Operating System</PlatformType>`
	`    <PlatformVersion>3.0.1</PlatformVersion>`
	`    <RemainingMemory>500</RemainingMemory>`
	`    <AvailableMemoryForProfiles>`
	`        750000`
	`    </AvailableMemoryForProfiles>`
	`    {SM_SR_ID_RPS}`
	`    <Isd-p-loadfile-aid>`
	`        #ISD_P_PKG_AID`
	`    </Isd-p-loadfile-aid>`
	`    <Isd-p-module-aid>#ISD_P_MOD_AID</Isd-p-module-aid>`
	`    #PROFILE1_RPS`
	`    #PROFILE2_RPS`
	`    <Isdr-r>#ISD_R_RPS</Isdr-r>`
	`    <Ecasd>#ECASD_RPS</Ecasd>`
	`    <EuiccCapabilities>`
	`        #FULL_CAP_RPS`
	`    </EuiccCapabilities>`
	`    <EumSignature`
	`        xmlns:ds="http://www.w3.org/2000/09/xmldsig">`
	`        #SIGNED_INFO_RPS`
	`        <ds:SignatureValue>`
	`            {SIGNATURE}`
	`        </ds:SignatureValue>`
	`        #KEY_INFO_RPS`
	`    </EnumSignature>`
	`</Eis>`
	
	see Note 1

ISD_R_RPS	`<Aid>#ISD_R_AID</Aid>`
	`<Tar>#ISD_R_TAR</Tar>`
	`<Sin>#VIRTUAL_SIN</Sin>`
	`<Sdin>#VIRTUAL_SDIN</Sdin>`
	`<Role>ISD-R</Role>`
	`<Keyset>`
	`    <version>01</version>`
	`    <Type>SCP80</Type>`
	`    <Cntr>01</Cntr>`

RPS element name	Value
	<Key kcv=”{KEY_KCV}”> <Index>01</Index> <KeyComponent type=”88” value=”{KEY_SECURED}”> </KeyComponent> </Key> <Key kcv=”{KEY_KCV}”> <Index>02</Index> <KeyComponent type=”88” value=”{KEY_SECURED}”> </KeyComponent> </Key> <Key kcv=”{KEY_KCV}”> <Index>03</Index> <KeyComponent type=”88” value=”{KEY_SECURED}”> </KeyComponent> </Key> </Keyset>
ISDP2_RPS	<Isd-p-aid>#ISD_P_AID2</Isd-p-aid>
ISDP3_RPS	<Isd-p-aid>#ISD_P_AID3</Isd-p-aid>
KEY_INFO_RPS	<ds:KeyInfo> <ds:X509Data> <ds:X509SubjectName> #EUM_S_CERT_ID_ECDSA </ds:X509SubjectName> </ds:X509Data> </ds:KeyInfo>
KEY_VERSION_RPS	<KeyVersionNumber>#SCP80_KVN</KeyVersionNumber> see Note 4
MNO1_ID_RPS	<Mno-id>#MNO1_S_ID</Mno-id>
MNO2_ID_RPS	<Mno-id>#MNO2_S_ID</Mno-id>
MORE_TODO_RPS	<MoreToDo>TRUE</MoreToDo>
NEW_ADDR_RPS	<newSubscriptionAddress> <Msisdn>#MSISDN3</Imsi> <Imsi>#IMSI3</Imsi> </newSubscriptionAddress>
NEW_ICCID_RPS	<Iccid>#NEW_ICCID</Iccid>
NO_MORE_TODO_RPS	<MoreToDo>FALSE</MoreToDo>
POL2_DEL_RPS	<pol2> <Rule> <Subject>PROFILE</Subject> <Action>DELETE</Action> <Qualification>Not allowed</Qualification> </Rule> </pol2>
POL2_DIS_RPS	<pol2> <Rule> <Subject>PROFILE</Subject> <Action>DISABLE</Action>

RPS element name	Value

	`<Qualification>Not allowed</Qualification>`
	`</Rule>`
	`</pol2>`

POL2_EMPTY_RPS	`<pol2/>`
PROF_TYPE1_RPS	`<ProfileType>#PROFILE_TYPE1</ProfileType>`
PROF_TYPE2_RPS	`<ProfileType>#PROFILE_TYPE2</ProfileType>`

PROFILE1_RPS	`<ProfileInfo>`
	`#ICCID1_RPS`
	`#ISDP2_RPS`
	`#MNO1_ID_RPS`
	`<FallbackAttribute>TRUE</FallbackAttribute>`
	`#SUB_ADDR1_RPS`
	`<State>Disabled</State>`
	`#PROF_TYPE1_RPS`
	`<AllocatedMemory>300000</AllocatedMemory>`
	`<FreeMemory>50000</FreeMemory>`
	`#POL2_DEL_RPS`
	`</ProfileInfo>`

PROFILE2_RPS	`<ProfileInfo>`
	`#ICCID2_RPS`
	`#ISDP3_RPS`
	`#MNO2_ID_RPS`
	`<FallbackAttribute>FALSE</FallbackAttribute>`
	`#SUB_ADDR2_RPS`
	`<State>Enabled</State>`
	`#PROF_TYPE2_RPS`
	`<AllocatedMemory>100000</AllocatedMemory>`
	`<FreeMemory>50000</FreeMemory>`
	`#POL2_DEL_RPS`
	`</ProfileInfo>`

SC3_NO_DR_RPS	`<ScenarioParameter>#SC3_NO_DR</ScenarioParameter>`
SC3_DR_RPS	`<ScenarioParameter>#SC3_DR</ScenarioParameter>`
SC3_DR_HOST_RPS	`<ScenarioParameter>#SC3_DR_HOST</ScenarioParameter>`
SIGNATURE_RPS	`<Signature>{SIGNATURE}</Signature>` see Note 5

SIGNED_INFO_RPS	`<ds:SignedInfo>`
	`<ds:CanonicalizationMethod Algorithm=”http://www.w3.org/2001/10/xml-exc-c14n”/>`
	`<ds:SignatureMethod Algorithm=”http://www.w3.org/2001/04/xmldsig-more#rsa-sha256”/>`
	`<ds:Reference>`
	`<ds:DigestMethod Algorithm=”http://www.w3.org/2001/04/xmlenc#sha256”/>`
	`<ds:DigestValue>{DIGEST}</ds:DigestValue>`
	`</ds:Reference>`
	`</ds:SignedInfo>`

SD_ISDP2_RPS	`<sd-aid>#ISD_P_AID2</sd-aid>`
SM_SR_S_ID_RPS	`<SmSr-id>#SM_SR_S_ID</SmSr-id>`
SM_SR_UT_ID_RPS	`<SmSr-id>#SM_SR_ID</SmSr-id>`
SMALL_MEM_RPS	`<RequiredMemory>999</RequiredMemory>`

RPS element name	Value

SUB_ADDR1_RPS	`<SubscriptionAddress>`
	`<Msisdn>#MSISDN1</Imsi>`
	`<Imsi>#IMSI1</Imsi>`
	`</SubscriptionAddress>`

SUB_ADDR2_RPS	`<SubscriptionAddress>`
	`<Msisdn>#MSISDN2</Imsi>`
	`<Imsi>#IMSI2</Imsi>`
	`</SubscriptionAddress>`

SUB_ADDR3_RPS	`<SubscriptionAddress>`
	`<Msisdn>#MSISDN3</Imsi>`
	`<Imsi>#IMSI3</Imsi>`
	`</SubscriptionAddress>`

TGT_SR_S_ID_RPS	`<Target-SmSr-id>#SM_SR_S_ID</Target-SmSr-id>`
TGT_SR_UT_ID_RPS	`<Target-SmSr-id>#SM_SR_ID</Target-SmSr-id>`
TIMESTAMP_RPS	`<completionTimestamp>{CURRENT_DATE}</completionTimestamp>`

VALID_SR_CERTIF_RPS	`<smsrCertificate>`
	`'7F21'{L}#VALID_SM_SR_CERTIFICATE`
	`</smsrCertificate>`

VIRTUAL_EID_RPS	`<Eid>#VIRTUAL_EID</Eid>`
VIRTUAL_EID2_RPS	`<Eid>#VIRTUAL_EID2</Eid>`
Note 1: The `{SIGNATURE}` shall be based on the `#EIS_SIGNED_RPS` and generated with the `#EUM_S_SK_ECDSA` *Note 2: The `{SIGNATURE}` shall be based on the `#EIS_SIGNED_RPS` and NOT generated with the `#EUM_S_SK_ECDSA`* *Note 3: The `{SIGNATURE}` shall be based on the `#EIS2_SIGNED_RPS` and generated with the `#EUM_S_SK_ECDSA`* *Note 4: The `#SCP80_KVN` shall be converted in Integer* *Note 5: The `{SIGNATURE}` shall use the `{RC}` (see the method `STORE_ISDR_KEYS` defined in Annex D to have more details on the way to generate the signature)* *Note 6: As this RPS element is used to execute non-nominal tests, the content of the C-APDUs should not be executed on the eUICC (i.e. the C-APDUs do not have to be relevant)* *Note 7: The SK.CI.ECDSA is used instead of PK.CI.ECDSA* *Note 8: The ISD-R definition shall not contain the keyset information.*	

Table 12: RPS Elements

B.6 Profiles Information

Here is the different Profiles information used to execute the test cases defined in the section 5.2.6 of this Test Plan. This information is related to:

- the Profiles pre-installed on the eUICC
- the Profile that is dynamically loaded on the eUICC

The different values shall be either provided by the eUICC Manufacturer or the MNO owning the new Profile.

Profile information	Description
EIS_RPS	The eUICC Information Set (RPS format) related to the eUICC. The different data shall be consistent with the state of the eUICC after the manufacturing. The eUICC Manufacturer shall give, at least, these values: • EID (i.e. `#EID`) • EUM Identifier • production date • platform type • platform version • remaining memory • available memory for Profiles • all Profiles pre-installed information with (for each one) o ICCID (i.e. `#ICCID` if the Profile is Enabled) o ISD-P AID (i.e. `#DEFAULT_ISD_P_AID` if the Profile is Enabled) o MSISDN (i.e. `#MSISDN` if the Profile is Enabled) o Fall-back Attribute o state o Profile type o allocated memory o POL2 • ISD-R information with o AID (i.e. `#ISD_R_AID`) o SIN o SDIN o SCP80 and/or SCP81 keysets information • ECASD information with o AID (i.e. `#ECASD_AID`) o SIN o SDIN o certificate (i.e. `#ECASD_CERTIFICATE`) o the CI public key (i.e. `#PK_CI_ECDSA`, `#PK_CI_ECDSA_PARAM`) • eUICC capabilities o supported CAT_TP version and/or supported HTTPS version ▪ depends if O_HTTPS and O_CAT_TP are supported o supported secured packet version o supported remote provisioning version The tool provider shall format the data (i.e. RPS) and add: • the SM-SR-UT Identifier (i.e. `#SM_SR_ID`) • the ISD-P Executable Load File AID (i.e. `#ISD_P_PKG_AID`) • the ISD-P Executable Module AID (i.e. `#ISD_P_MOD_AID`) • the MNO Identifier of the pre-installed Profiles (i.e. `#MNO2_S_ID` shall be set on the default Enabled Profile) • the signature using the `#EUM_S_PK_ECDSA` and encrypt the secure channel keyset values using a key agreed by the SM-SR-UT
ICCID	The ICCID of the default Profile pre-installed on the eUICC.
MSISDN	The MSISDN of the default Profile pre-installed on the eUICC. A network connectivity shall be available with this mobile subscription.

Profile information	Description
NEW_ICCID	The ICCID of the new Profile dynamically downloaded on the eUICC. This ICCID shall not be present on the #EIS_RPS.
NEW_MSISDN	The MSISDN of the new Profile dynamically downloaded on the eUICC. This MSISDN shall not be present on the #EIS_RPS. A network connectivity shall be available with this mobile subscription.
MNO1_CON_NAN	The NAN, of the new Profile dynamically downloaded on the eUICC, which allows MNO's network connection.
MNO1_CON_LOGIN	The NAN related login, of the new Profile dynamically downloaded on the eUICC, which allows MNO's network connection.
MNO1_CON_PWD	The NAN related password, of the new Profile dynamically downloaded on the eUICC, which allows MNO's network connection.
MNO1_CON_TON_NPI	The TON and NPI of the MNO that owns the new Profile dynamically downloaded on the eUICC.
MNO1_CON_DIAL_NUM	The dialing number of the MNO that owns the new Profile dynamically downloaded on the eUICC.
MNO2_CON_NAN	The NAN, of the Enabled Profile pre-installed on the eUICC, which allows MNO's network connection.
MNO2_CON_LOGIN	The NAN related login, of the Enabled Profile pre-installed on the eUICC, which allows MNO's network connection.
MNO2_CON_PWD	The NAN related password, of the Enabled Profile pre-installed on the eUICC, which allows MNO's network connection.
MNO2_CON_TON_NPI	The TON and NPI of the MNO that owns the Enabled Profile pre-installed on the eUICC.
MNO2_CON_DIAL_NUM	The dialing number of the MNO that owns the Enabled Profile pre-installed on the eUICC.
SM_SR_DEST_ADDR	The destination address of the SM-SR-UT.
SM_SR_UDP_IP	The UDP IP of the SM-SR-UT related to the CAT_TP implementation.
SM_SR_UDP_PORT	The UDP port of the SM-SR-UT related to the CAT_TP implementation.
SM_SR_TCP_IP	The TCP IP of the SM-SR-UT related to the HTTPS implementation.
SM_SR_TCP_PORT	The TCP port of the SM-SR-UT related to the HTTPS implementation.
SM_SR_HTTP_URI	The URI of the SM-SR-UT related to the HTTPS implementation.
SM_SR_HTTP_HOST	The HOST of the SM-SR-UT related to the HTTPS implementation.

Table 13: Profiles Information

B.7 Profile Package Description

Here is a description of the Profile Package content that should be used during the testing of the Profile download process (see section 4.2.18). Some parts of this PEs list may be adapted according to the eUICC implementation.

This Profile, defined in Table 14**: Profile Package Content**, contains the following Components:

- MF and USIM ADF
- PIN and PUK codes
- NAA using Milenage algorithm
- MNO-SD supporting SCP80 in 3DES
- SSD supporting SCP80 in 3DES
- RFM application

The parameters below have been chosen to personalize the Profile:

- Profile type: "GSMA Profile Package"
- ICCID: '89019990001234567893'
- IMSI: 234101943787656
- MNO-SD AID / TAR: 'A000000151000000' / 'B20100'
- UICC RFM application AID / TAR: 'A00000055910100001' / 'B00000'
- USIM RFM application AID / TAR: ' A00000055910100002' / 'B00020'
- Executable Load File AID for SD: 'A0000001515350'
- Executable Module AID for SD: 'A000000151000000'
- SSD AID / TAR: 'A00000055910100102736456616C7565' / '6C7565'
- All access rules are defined in the Table 15

Note that all these parameters may be freely adapted if necessary.

B.7.1 Profile Package Content

The `#PROFILE_PACKAGE` should be the result of the concatenation of the different PEs described below (respecting the order).

ASN.1 format	DER TLV format
PE_HEADER	
headerValue ProfileElement ::= header : {	A0 4F
major-version 2,	80 01 02
minor-version 0,	81 01 00
profileType "GSMA Profile Package",	82 14 47534D412050726F66696C65205061636B616765
iccid '89019990001234567893'H,	83 0A 89019990001234567893
eUICC-Mandatory-services {	A5 06
usim NULL,	81 00
milenage NULL,	84 00
javacard NULL	8B 00
},	
eUICC-Mandatory-GFSTEList {	A6 10
-- *see Note 1*	
id-MF,	06 06 67810F010201
id-USIM	06 06 67810F010204
},	
-- *These SMS connectivity parameters may be freely changed*	
connectivityParameters 'A0090607#TON_NPI#DIALING_NUMBER'H	87 0B A0090607913386994211F0
}	
PE_MF	
mfValue ProfileElement ::= mf : {	B0 8201F9
mf-header {	A0 05
mandated NULL,	80 00
identification 1	81 01 01
},	
templateID id-MF,	81 06 67810F010201
mf {	A2 07
fileDescriptor : {	A1 05
pinStatusTemplateDO '01020A'H	C6 03 01020A

```
    }
  },
  ef-pl {                                                            A3 05
    fileDescriptor : {                                                 A1 03
    -- EF PL modified to use Access Rule 15 within EF ARR
      securityAttributesReferenced '0F'H                                 8B 01 0F
    }
  },
  ef-iccid {                                                         A4 0C
    -- swapped ICCID: 98109909002143658739
    fillFileContent '98109909002143658739'H                            83 0A 98109909002143658739
  },
  ef-dir {                                                           A5 28
    fileDescriptor {                                                   A1 0A
      -- Shareable Linear Fixed File
      -- 4 records, record length: 38 bytes
      fileDescriptor '42210026'H,                                        82 04 42210026
      efFileSize '0098'H                                                 80 02 0098
    },
    -- USIM AID: A0000000871002FF33FF018900000100
    fillFileContent                                                    83 1A
      '61184F10A0000000871002FF33FF01890000010050045553494D'H            61184F10A0000000871002FF33FF01890000010050045553494D
  },
  ef-arr {                                                           A6 82019E
    fileDescriptor {                                                   A1 0A
      -- Shareable Linear Fixed File
      -- 15 records, record length: 37 bytes
      -- ARR created with content defined in Annex B.7.2
      -- plus one additional record for use with EF PL
      fileDescriptor '42210025'H,                                        82 04 42210025
      efFileSize '022B'H                                                 80 02 022B
    },
    -- see Table 15 to see the access rules definitions
    fillFileContent '#ACCESS_RULE1'H,                                  83 1B #ACCESS_RULE1
    fillFileOffset 10,                                                 82 01 0A
    fillFileContent '#ACCESS_RULE2'H,                                  83 16 #ACCESS_RULE2
    fillFileOffset 15,                                                 82 01 0F
    fillFileContent '#ACCESS_RULE3'H,                                  83 0B #ACCESS_RULE3
```

fillFileOffset 26, fillFileContent '#ACCESS_RULE4'H, fillFileOffset 27, fillFileContent '#ACCESS_RULE5'H, fillFileOffset 15, fillFileContent '#ACCESS_RULE6'H, fillFileOffset 15, fillFileContent '#ACCESS_RULE7'H, fillFileOffset 4, fillFileContent '#ACCESS_RULE8'H, fillFileOffset 4, fillFileContent '#ACCESS_RULE9'H, fillFileOffset 10, fillFileContent '#ACCESS_RULE10'H, fillFileOffset 21, fillFileContent '#ACCESS_RULE11'H, fillFileOffset 16, fillFileContent '#ACCESS_RULE12'H, fillFileOffset 21, fillFileContent '#ACCESS_RULE13'H, fillFileOffset 15, fillFileContent '#ACCESS_RULE14'H, fillFileOffset 26, fillFileContent '8001019000800102A010A40683010195 0108A406830102950108800158A40683 010A950108'H } }	82 01 1A 83 0A #ACCESS_RULE4 82 01 1B 83 16 #ACCESS_RULE5 82 01 0F 83 16 #ACCESS_RULE6 82 01 0F 83 21 #ACCESS_RULE7 82 01 04 83 21 #ACCESS_RULE8 82 01 04 83 1B #ACCESS_RULE9 82 01 0A 83 10 #ACCESS_RULE10 82 01 15 83 15 #ACCESS_RULE11 82 01 10 83 10 #ACCESS_RULE12 82 01 15 83 16 #ACCESS_RULE13 82 01 0F 83 0B #ACCESS_RULE14 82 01 1A 83 25 8001019000800102A010A40683010195 0108A406830102950108800158A40683 010A950108
PE_PUK	

pukVal ProfileElement ::= pukCodes : {
puk-Header {
mandated NULL,
identification 3
},
pukCodes {
	A3 3F
	A0 05
	80 00
	81 01 03
	
	A1 36

```
    {
      keyReference pukAppl1,
      pukValue '3030303030303030'H,
      -- maxNumOfAttemps:9, retryNumLeft:9
      maxNumOfAttemps-retryNumLeft 153
    },
    {
      keyReference pukAppl2,
      pukValue '3132333435363738'H
    },
    {
      keyReference secondPUKAppl1,
      pukValue '3932393435363738'H,
      -- maxNumOfAttemps:8, retryNumLeft:8
      maxNumOfAttemps-retryNumLeft 136
    }
  }
}
```

```
    30 11
      80 01 01
      81 08 3030303030303030

      82 02 0099

    30 0D
      80 01 02
      81 08 3132333435363738

    30 12
      80 02 0081
      81 08 3932393435363738

      82 02 0088
```

PE_PIN

```
pinVal ProfileElement ::= pinCodes : {
  pin-Header {
    mandated NULL,
    identification 4
  },
  pinCodes pinconfig : {
    {
      keyReference pinAppl1,
      pinValue '31323334FFFFFFFF'H,
      unblockingPINReference pukAppl1
    },
    {
      keyReference pinAppl2,
      pinValue '30303030FFFFFFFF'H,
      unblockingPINReference pukAppl2
    },
    {
```

```
A2 41
  A0 05
    80 00
    81 01 04

  A1 38 A0 36
    30 10
      80 01 01
      81 08 31323334FFFFFFFF
      82 01 01

    30 10
      80 01 02
      81 08 30303030FFFFFFFF
      82 01 02

    30 10
```

keyReference adm1,	80 01 0A
pinValue '35363738FFFFFFFF'H,	81 08 35363738FFFFFFFF
pinAttributes 1	83 01 01
}	
}	
}	
PE_USIM	
usimValue ProfileElement ::= usim : {	B3 7C
usim-header {	A0 05
mandated NULL,	80 00
identification 2	81 01 02
},	
templateID id-USIM,	81 06 67810F010204
adf-usim {	A2 1D
fileDescriptor : {	A1 1B
fileID '7FF1'H,	83 02 7FF1
dfName 'A0000000871002FF33FF018900000100'H,	84 10 A0000000871002FF33FF018900000100
pinStatusTemplateDO '01810A'H	C6 03 01810A
}	
},	
ef-imsi {	A3 0B
-- numerical format: 234101943787656	
fillFileContent '082943019134876765'H	83 09 082943019134876765
},	
ef-arr {	A4 06
fileDescriptor {	A1 04
linkPath '2F06'H	C7 02 2F06
}	
},	
ef-ust {	A8 0F
-- Service Dialling Numbers, Short Message Storage…	
fillFileContent '0A2E178CE73204000000000000'H	83 0D 0A2E178CE73204000000000000
},	
ef-spn {	AD 13
-- ASCII format: "GSMA eUICC"	
fillFileContent '0247534D41206555494343FFFFFFFFFFFF'H	83 11 0247534D41206555494343FFFFFFFFFFFF

```
  },
  ef-est {                                          AE 03
    -- Services deactivated
    fillFileContent '00'H                             83 01 00
  },
  ef-acc {                                          B2 04
    -- Access class 4
    fillFileContent '0040'H                           83 02 0040
  },
  ef-ecc {                                          B6 06
    -- Emergency Call Code 911
   fillFileContent '19F1FF01'H                        83 04 19F1FF01
  }
}
```

PE_USIM_PIN

```
usimPin ProfileElement ::= pinCodes : {             A2 25
  pin-Header {                                        A0 05
    mandated NULL,                                      80 00
    identification 40                                   81 01 28
  },
  pinCodes pinconfig : {                              A1 1C A0 1A
    {                                                   30 18
      keyReference secondPINAppl1,                        80 02 0081
      pinValue '39323338FFFFFFFF'H                        81 08 39323338FFFFFFFF
     unblockingPINReference secondPUKAppl1,               82 02 0081
      -- PIN is Enabled
      pinAttributes 1,                                    83 01 01
      -- maxNumOfAttemps:2, retryNumLeft:2
      maxNumOfAttemps-retryNumLeft 34                     84 01 22
    }
  }
}
```

PE_NAA

```
akaParamValue ProfileElement ::= akaParameter : {   A4 3A
```

`  aka-header {`	`  A0 05`
`    mandated NULL,`	`    80 00`
`    identification 5`	`    81 01 05`
`  },`	
`  algoConfiguration algoParameter : {`	`  A1 31 A1 2F`
`    algorithmID milenage,`	`    80 01 01`
`    -- RES and MAC 64 bits, CK and IK 128 bits`	
`    algorithmOptions '01'H,`	`    81 01 01`
`    key '000102030405060708090A0B0C0D0E0F'H,`	`    82 10 000102030405060708090A0B0C0D0E0F`
`    opc '0102030405060708090A0B0C0D0E0F00'H,`	`    83 10 0102030405060708090A0B0C0D0E0F00`
`    -- rotationConstants uses default: '4000204060'H`	
`    -- xoringConstants uses default value`	
`    authCounterMax '010203'H`	`    86 03 010203`
`  }`	
`  -- sqnOptions uses default: '02'H`	
`  -- sqnDelta uses default: '000010000000'H`	
`  -- sqnAgeLimit uses default: '000010000000'H`	
`  -- sqnInit uses default: all bytes zero`	
`}`	
PE_MNO_SD	
`mnoSdValue ProfileElement ::= securityDomain : {`	`A6 81B5`
`  sd-Header {`	`  A0 05`
`    mandated NULL,`	`    80 00`
`    identification 6`	`    81 01 06`
`  },`	
`  instance {`	`  A1 3E`
`    applicationLoadPackageAID 'A0000001515350'H,`	`    4F 07 A0000001515350`
`    classAID 'A000000151535041'H,`	`    4F 08 A000000151535041`
`    instanceAID 'A000000151000000'H,`	`    4F 08 A000000151000000`
`    applicationPrivileges '82FC80'H,`	`    82 03 82FC80`
`    -- Secured`	
`    lifeCycleState '0F'H,`	`    83 01 0F`
`    -- SCP80 supported`	
`    applicationSpecificParametersC9 '81028000'H,`	`    C9 04 81028000`
`    -- other parameters may be necessary`	
`    applicationParameters {`	`    EA 11`

```
      -- TAR: B20100, MSL: 12
      uiccToolkitApplicationSpecificParametersField                  80 0F
         '01000001000000002011203B2010000'H                            01000001000000002011203B2010000
    }
  },
  keyList {                                                        A2 6C
    {                                                                30 22
      -- C-ENC + R-ENC
      keyUsageQualifier '38'H,                                         95 01 38
      -- ENC key
      keyIdentifier '01'H,                                             82 01 01
      keyVersionNumber '01'H,                                          83 01 01
      keyCompontents {                                                 30 17
        {                                                                30 15
          -- DES mode implicitly known (as an example)
          keyType '80'H,                                                   80 01 80
          -- This value may be freely changed
          keyData '112233445566778899AABBCCDDEEFF10'H                      86 10 112233445566778899AABBCCDDEEFF10
        }
      }
    },
    {                                                                30 22
      -- C-MAC + R-MAC
      keyUsageQualifier '34'H,                                         95 01 34
      -- MAC key
      keyIdentifier '02'H,                                             82 01 02
      keyVersionNumber '01'H,                                          83 01 01
      keyCompontents {                                                 30 17
        {                                                                30 15
          -- DES mode implicitly known (as an example)
          keyType '80'H,                                                   80 01 80
          -- This value may be freely changed
          keyData '112233445566778899AABBCCDDEEFF10'H                      86 10 112233445566778899AABBCCDDEEFF10
        }
      }
    },
    {                                                                30 22
      -- C-DEK + R-DEK
```

`      keyUsageQualifier 'C8'H,`	`      95 01 C8`
`      -- data ENC key`	
`      keyIdentifier '03'H,`	`      82 01 03`
`      keyVersionNumber '01'H,`	`      83 01 01`
`      keyCompontents {`	`      30 17`
`        {`	`        30 15`
`          -- DES mode implicitly known (as an example)`	
`          keyType '80'H,`	`          80 01 80`
`          -- This value may be freely changed`	
`          keyData '112233445566778899AABBCCDDEEFF10'H`	`          86 10 112233445566778899AABBCCDDEEFF10`
`        }`	
`      }`	
`    }`	
`  }`	
`}`	
PE_SSD	
`ssdValue ProfileElement ::= securityDomain : {`	`A6 81C7`
`  sd-Header {`	`  A0 05`
`    mandated NULL,`	`    80 00`
`    identification 7`	`    81 01 07`
`  },`	
`  instance {`	`  A1 50`
`    applicationLoadPackageAID 'A0000001515350'H,`	`    4F 07 A0000001515350`
`    classAID 'A000000151535041'H,`	`    4F 08 A000000151535041`
`    instanceAID 'A000000559101001027364566l6C7565'H,`	`    4F 10 A0000005591010010273645666l6C7565`
`    -- MNO-SD`	
`    extraditeSecurityDomainAID 'A000000151000000'H,`	`    4F 08 A000000151000000`
`    -- Security Domain + Trusted Path`	
`    applicationPrivileges '808000'H,`	`    82 03 808000`
`    -- Personalized`	
`    lifeCycleState '0F'H,`	`    83 01 0F`
`    -- SCP80 supported`	
`    applicationSpecificParametersC9 '81028000'H,`	`    C9 04 81028000`
`    applicationParameters {`	`    EA 11`
`      -- TAR: 6C7565, MSL: 12`	
`      uiccToolkitApplicationSpecificParametersField`	`      80 0F`

'0100000100000000020112036C756500'H	0100000100000000020112036C756500
}	
},	
keyList {	A2 6C
{	30 22
-- C-ENC + R-ENC	
keyUsageQualifier '38'H,	95 01 38
-- ENC key	
keyIdentifier '01'H,	82 01 01
keyVersionNumber '01'H,	83 01 01
keyCompontents {	30 17
{	30 15
-- DES mode implicitly known (as an example)	
keyType '80'H,	80 01 80
-- This value may be freely changed	
keyData '11223344556677881122334455667788'H	86 10 11223344556677881122334455667788
}	
}	
},	
{	30 22
-- C-MAC + R-MAC	
keyUsageQualifier '34'H,	95 01 34
-- MAC key	
keyIdentifier '02'H,	82 01 02
keyVersionNumber '01'H,	83 01 01
keyCompontents {	30 17
{	30 15
-- DES mode implicitly known (as an example)	
keyType '80'H,	80 01 80
-- This value may be freely changed	
keyData '11223344556677881122334455667788'H	86 10 11223344556677881122334455667788
}	
}	
},	
{	30 22
-- C-DEK + R-DEK	
keyUsageQualifier 'C8'H,	95 01 C8
-- data ENC key	

keyIdentifier '03'H,	82 01 03
keyVersionNumber '01'H,	83 01 01
keyCompontents {	30 17
{	30 15
-- DES mode implicitly known (as an example)	
keyType '80'H,	80 01 80
-- This value may be freely changed	
keyData '11223344556677881122334455667788'H	86 10 11223344556677881122334455667788
}	
}	
}	
}	
}	
PE_RFM_UICC	
rfmUicc ProfileElement ::= rfm : {	A7 20
rfm-header {	A0 03
identification 9	81 01 09
},	
-- Instance AID	
instanceAID ' A00000055910100001'H,	4F 09 A00000055910100001
tarList {	A0 05 04 03
'B00000'H	B00000
},	
-- cryptographic checksum + counter higher	
minimumSecurityLevel '12'H,	81 01 12
-- full access	
uiccAccessDomain '00'H,	04 01 00
-- full access	
uiccAdminAccessDomain '00'H	04 01 00
}	
PE_RFM_USIM	
rfmUsim ProfileElement ::= rfm : {	A7 40
rfm-header {	A0 03
identification 10	81 01 0A

},	
-- Instance AID	
instanceAID 'A00000055910100002'H,	4F 09 A00000055910100002
tarList {	A0 05
'B00020'H	04 03 B00020
},	
-- cryptographic checksum + counter higher	
minimumSecurityLevel '12'H,	81 01 12
-- full access	
uiccAccessDomain '00'H,	04 01 00
-- full access	
uiccAdminAccessDomain '00'H,	04 01 00
adfRFMAccess {	30 1E
adfAID 'A0000000871002FF33FF018900000100'H,	80 10 A0000000871002FF33FF018900000100
-- UICC access condition: ADM1	
adfAccessDomain '02000100'H,	81 04 02000100
-- UICC access condition: ADM1	
adfAdminAccessDomain '02000100'H	82 04 02000100
}	
}	
PE_END	
endValue ProfileElement ::= end : {	AA 07
end-header {	A0 05
mandated NULL,	80 00
identification 99	81 01 63
}	
}	
Note: The rule related to the usage of curly brackets defined in section 2.2.2 shall not apply for the elements described in the column “ASN.1 format” of this table. *Note 1: The following OIDs are used:* *id-MF OBJECT IDENTIFIER ::=* *{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) mf(1)}* *id-USIM OBJECT IDENTIFIER ::=* *{joint-iso-itu-t(2) international-organizations(23) simalliance(143) euicc-profile(1) template(2) usim(4)}* *These OIDs allow identifying the templates used to accelerate the creation of the file system in the Profile as defined in the SIMAlliance Profile Package specification [16].*	

Table 14: Profile Package Content

B.7.2 Access Rules

Here are the access rules used in the Profile Package content defined in Profile Package **Content**.

Access rule name	File access conditions						Hexadecimal value
	READ	UPDATE	INCREASE	ACTIVATE	DEACTIVATE	DELETE	
ACCESS_RULE1	ALWAYS	PIN1	NEVER	ADM1	ADM1	ADM1	8001019000 800102A406830101950108 800158A40683010A950108
ACCESS_RULE2	PIN1	ADM1	NEVER	ADM1	ADM1	ADM1	800101A406830101950108 80015AA40683010A950108
ACCESS_RULE3	ADM1	ADM1	NEVER	ADM1	ADM1	ADM1	80015BA40683010A950108
ACCESS_RULE4	ALWAYS	NEVER	NEVER	NEVER	NEVER	ADM1	8001019000 80015A9700
ACCESS_RULE5	PIN1	PIN1	NEVER	ADM1	ADM1	ADM1	800103A406830101950108 800158A40683010A950108
ACCESS_RULE6	PIN1	ADM1	NEVER	PIN1	ADM1	ADM1	800111A406830101950108 80014AA40683010A950108
ACCESS_RULE7	PIN1	PIN1	PIN1	ADM1	ADM1	ADM1	800103A406830101950108 800158A40683010A950108 840132A406830101950108
ACCESS_RULE8	PIN1	PIN2	NEVER	ADM1	ADM1	ADM1	800101A406830101950108 800102A406830181950108 800158A40683010A950108
ACCESS_RULE9	ALWAYS	PIN1	NEVER	PIN1	PIN1	ADM1	8001019000 80011AA406830101950108 800140A40683010A950108
ACCESS_RULE10	ALWAYS	ADM1	NEVER	ADM1	ADM1	ADM1	8001019000 80015AA40683010A950108

ACCESS_RULE11	ALWAYS	NEVER	NEVER	ADM1	ADM1	NEVER	8001019000 800118A40683010A950108 8001429700
ACCESS_RULE12	PIN1	NEVER	NEVER	NEVER	NEVER	NEVER	800101A406830101950108 80015A9700
ACCESS_RULE13	PIN1	PIN1	NEVER	PIN1	ADM1	ADM1	800113A406830101950108 800148A40683010A950108
Access rule name	**MF/ADF/DF access conditions**						**Hexadecimal value**
	DELETE self	**TERMINATE**	**ACTIVATE**	**DEACTIVATE**	**CREATE DF**	**CREATE EF**	
ACCESS_RULE14	ADM1	NEVER	ADM1	ADM1	ADM1	ADM1	80015EA40683010A950108
Note: These access rules strictly follow the definition provided in the SIMAlliance Profile Package specification [16] (section 9.9)							

Table 15: Access Rules

B.7.3 Additional Profile Elements

Here are additional Profile Elements that shall be added to the Profile Package content defined above in order to execute the tests defined in section 5.2:

- `#PE_APPLET1`: This PE allows loading and instantiating the Applet 1 defined in section A.1
- `#PE_APPLET3`: This PE allows loading and instantiating the Applet 3 defined in section A.3
- `#PE_EF1122`: This PE allows creating an EF with the identifier ‘1122’. This transparent file is 16 bytes long, activated and present under the MF ‘3F00’

ASN.1 format	DER TLV format
PE_APPLET1	
`applet1 ProfileElement ::= application : {`	`A8 {L}`
`  app-Header {`	`  A0 05`
`    mandated NULL,`	`    80 00`
`    identification 20`	`    81 01 14`
`  },`	
`  loadBlock {`	`  A1 {L}`
`    loadPackageAID 'A000000559101001'H,`	`    4F 08 A000000559101001`
`    loadBlockObject '{LFDB_APPLET1}'H`	`    C4 {L} {LFDB_APPLET1}`
`  },`	
`  instanceList {`	`  A2 41`
`    {`	`    30 3F`
`      applicationLoadPackageAID 'A000000559101001'H,`	`      4F 08 A000000559101001`
`      classAID 'A000000559101001112233'H,`	`      4F 0B A000000559101001112233`
`      instanceAID 'A00000055910100111223301'H,`	`      4F 0C A00000055910100111223301`
`      applicationPrivileges '000000'H,`	`      82 03 000000`
`      -- Selectable`	
`      lifeCycleState '07'H,`	`      83 01 07`
`      applicationSpecificParametersC9 '00'H,`	`      C9 01 00`
`      applicationParameters {`	`      EA 0D`
`        uiccToolkitApplicationSpecificParametersField`	`        80 0B`
`          -- TAR: 112233`	
`          '0100000000000311223300'H`	`          0100000000000311223300`
`      }`	
`    }`	
`  }`	
`}`	
see Note 1	

PE_APPLET3	

applet3 ProfileElement ::= application : {
  app-Header {
    mandated NULL,
    identification 21
  },
  loadBlock {
    loadPackageAID 'A000000559101003'H,
    loadBlockObject '{LFDB_APPLET3}'H
  },
  instanceList {
    {
      applicationLoadPackageAID 'A000000559101003'H,
      classAID 'A000000559101003445566'H,
      instanceAID 'A0000005591010034455 6601'H,
      applicationPrivileges '000000'H,
      -- Selectable
      lifeCycleState '07'H,
      applicationSpecificParametersC9 '00'H
    }
  }
}
see Note 1
	A8 {L}
	  A0 05
	    80 00
	    81 01 15
	
	  A1 {L}
	    4F 08 A000000559101003
	    C4 {L} {LFDB_APPLET3}
	
	  A2 32
	    30 30
	      4F 08 A000000559101003
	      4F 0B A000000559101003445566
	      4F 0C A0000005591010034455 6601
	      82 03 000000
	
	      83 01 07
	      C9 01 00

PE_EF1122	

`ef1122 ProfileElement ::= genericFileManagement : {`
`  gfm-header {`
`    mandated NULL,`
`    identification 22`
`  },`
`  fileManagementCMD {`
`    {`
`      createFCP {`
`        -- Transparent File`
`        fileDescriptor '0121'H,`
`        fileID '1122'H,`
`        -- reference to the #ACCESS_RULE1`
`        securityAttributesReferenced '2F0601'H,`
`        efFileSize '0010'H,`
`        shortEFID ''H`
`      },`
`      fillFileContent '1122334455'H`
`    }`
`  }`
`}`
see Note 2
	`A1 27`
	`  A0 05`
	`    80 00`
	`    81 01 16`
	
	`  A1 1E`
	`    30 1C`
	`      62 13`
	
	`        82 02 0121`
	`        83 02 1122`
	
	`        8B 03 2F0601`
	`        80 02 0010`
	`        88 00`
	
	`      81 05 1122334455`

Note: The rule related to the usage of curly brackets defined in section 2.2.2 shall not apply for the elements described in the column "ASN.1 format". *Note 1: This PE shall be added just before the `#PE_END`.* *Note 2: This PE shall be added just after the `#PE_PIN`.*	

Table 16: Additional Profile Elements

Annex C Dynamic Content

Here are the different dynamic values used in the test cases defined in this document. These values should be either calculated by the test tools or generated dynamically by an entity under test.

Variable name	Description
ACK_NUM	CAT_TP PDU acknowledgment number (2 bytes long) as defined in ETSI TS 102 127 [7].
CARD_CHALLENGE	Pseudo-random value (8 bytes long).
CARD_CRYPTOGRAM	Card cryptogram as defined in GlobalPlatform Card Specification - Amendment D [11] (8 bytes long).
CC	Cryptographic Checksum as defined in ETSI TS 102 225 [4] (8 bytes long).
CNTR	Counter coded on 5 bytes as defined in ETSI TS 102 225 [4].
COMMAND_SCRIPT	List of commands to execute formatted in expanded format as defined in ETSI TS 102 226 [6].
CPI	Command Packet Identifier as defined in ETSI TS 102 225 [4].
CS	CAT_TP PDU checksum (2 bytes long) as defined in ETSI TS 102 127 [7].
CURRENT_DATE	The current date formatted as specified by W3C: YYYY-MM-DDThh:mm:ssTZD
DATA	CAT_TP PDU data as defined in ETSI TS 102 127 [7].
DATA_LENGTH	CAT_TP PDU data length as defined in ETSI TS 102 127 [7].
DEST_PORT	CAT_TP PDU destination port (2 bytes long) as defined in ETSI TS 102 127 [7].
DIGEST	SHA-256 of the data to sign.
DR	Derivation Random as defined in GlobalPlatform Card Specification v.2.2 Amendment E [12] (Confidential Setup of Secure Channel Keys using ECKA).
FUNC_CALL_ID	Identification of a function call. This identifier enables to manage function call retry policies. As consequence, it shall be unique.
FUNCTION_REC_ID	Depending of the direction of the test step, this value shall be either: • #SM_DP_ID or • #SM_SR_ID or • #SM_DP_S_ID or • #SM_SR_S_ID or • #MNO1_S_ID or • #MNO2_S_ID or • #EUM_S_ID
FUNCTION_REQ_ID	Depending of the direction of the test step, this value shall be either: • #SM_DP_ID or • #SM_SR_ID or • #SM_DP_S_ID or • #SM_SR_S_ID or • #MNO1_S_ID or • #MNO2_S_ID or • #EUM_S_ID
HL	CAT_TP PDU header length (1 byte) as defined in ETSI TS 102 127 [7].
HOST_CHALLENGE	Random value (8 bytes long).
HOST_CRYPTOGRAM	Host cryptogram as defined in GlobalPlatform Card Specification - Amendment D [11] (8 bytes long).
IDENTIFICATION_DATA	CAT_TP off-card entity identification data as defined in ETSI TS 102 127 [7].

Variable name	Description
KEY_DIV_DATA	Key diversification data as defined in GlobalPlatform Card Specification - Amendment D [11] (10 bytes long).
KEY_KCV	The Key Check Value of the #KEY.
KEY_LENGTH	Symmetric key length that shall be at least 16 bytes long.
KEY_SECURED	The #KEY encrypted with a transport key agreed between the SM-DP and the SM-SR (as defined in GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2]).
KEYS_ENCRYPTED	Encrypted secure channel keys used during the confidential setup. The value of each plain key is #KEY.
KIC	SC80 Key and algorithm Identifier for ciphering as defined in ETSI TS 102 225 [4].
KID	SCP80 Key and algorithm Identifier for RC/CC/DS as defined in ETSI TS 102 225 [4].
L	Exact length of the corresponding tag or of the remaining data.
LC	Exact length of a command data.
LFDB_APPLET1	Load File Data Block of the Applet1 defined in Annex A.
LFDB_APPLET3	Load File Data Block of the Applet3 defined in Annex A.
LOAD_APPLET1	List of C-APDUs that allows loading the Applet1 defined in Annex A. The script is composed of one INSTALL FOR LOAD and several LOAD commands.
LOAD_APPLET2	List of C-APDUs that allows loading the Applet2 defined in Annex A. The script is composed of one INSTALL FOR LOAD and several LOAD commands.
LOAD_APPLET3	List of C-APDUs that allows loading the Applet3 defined in Annex A. The script is composed of one INSTALL FOR LOAD and several LOAD commands.
MAC	C-MAC as defined in GlobalPlatform Card Specification – Amendment D [11].
MAX_PDU_SIZE	CAT_TP maximum PDU size (2 bytes long) as defined in ETSI TS 102 127 [7].
MAX_SDU_SIZE	CAT_TP maximum SDU size (2 bytes long) as defined in ETSI TS 102 127 [7].
NB_APP	Number of applications installed.
NON_VOLATILE_MEMORY	Non volatile memory available.
NOTIF_NUMBER	The notification sequence number as defined in GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2].
PCNTR	Padding Counter coded on 1 byte as defined in ETSI TS 102 225 [4].
PK_CASD_CT	Symmetric or asymmetric key (depending of the implementation choice) of the MNO CASD.
PROFILE_PART1	The first part of the Profile Elements list defined by #PROFILE_PACKAGE. This part of the Profile Package shall be split according the eUICC capabilities.
PROFILE_PARTi	An intermediate part of the Profile Elements list defined by #PROFILE_PACKAGE. Each middle part of the Profile Package shall be split according the eUICC capabilities.
PROFILE_PARTn	The last part of the Profile Elements list defined by #PROFILE_PACKAGE. This part of the Profile Package shall be split according the eUICC capabilities.
RC	Random Challenge as defined in GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2].
REASON_CODE	CAT_TP reason code as defined in ETSI TS 102 127 [7].
RECEIPT	Receipt as defined in GlobalPlatform Card Specification v.2.2 Amendment E [12] (Confidential Setup of Secure Channel Keys using ECKA).
REL_MESSAGE_ID	Identifier of the initial message request.
REQ_MESSAGE_ID	Identifier of the message to send. It shall be unique and composed of the domain portion of the tool provider and an integer (or a date).

Variable name	Description
SCP_KDEK	The new SCP DEK key generated on the ISD-R or the ISD-P.
SCP_KENC	The new SCP ENC key generated on the ISD-R or the ISD-P.
SCP_KMAC	The new SCP MAC key generated on the ISD-R or the ISD-P.
SCP03_SEQ_NUM	The SCP03 sequence number (3 bytes long).
SEQ_NUM	CAT_TP PDU sequence number (2 bytes long) as defined in ETSI TS 102 127 [7].
SIGNATURE	A signature used for key set establishment.
SM_SR_ID_RPS	The SM-SR identifier structure used in off-card interfaces. Depending of the test, this value shall be either: • #SM_SR_UT_ID_RPS or • #SM_SR_S_ID_RPS
SRC_PORT	CAT_TP PDU source port (2 bytes long) as defined in ETSI TS 102 127 [7].
TOKEN_KEY	The AES token key value (key version number = ‘70’) of the ISD-P (16 bytes long).
TOKEN_VALUE	The token generated with the {TOKEN_KEY} (16 bytes long).
UDH	User Data Header as defined in 3GPP TS 23.040 [5].
VOLATILE_MEMORY	Volatile memory available.
WIN_SIZE	CAT_TP PDU window size port (2 bytes long) as defined in ETSI TS 102 127 [7].

Table 17: Dynamic Content

Annex D Methods

Here are the methods' descriptions used in this document:

<table>
<tr><th>Method name</th><th>Explanation</th></tr>
<tr><td>

ENVELOPE_SMS_PP

</td><td>

Generate an SMS envelope.

Parameters:

- *SPI*
- *TAR*
- *COMMAND1; COMMAND2…* (i.e. APDUs or TLVs)
- *CHAINING_OPT* (optional parameter)

Here is the content of the envelope SMS-PP download to send:

```
'80 C2 00 00 {LC}
   D1 {L}
     82 02 83 81
     86 02 80 01
     8B {L}
       40 05 81 12 50 F3 96 F6 22 22 22 22 22 22 22
       {L} {UDH}' +
       SCP80_PACKET(SPI, TAR, COMMAND1;COMMAND2…, CHAINING_OPT)
```

See Annex C for the definition of `{UDH}`.

The method `SCP80_PACKET` is defined below.

If the SMS content length is higher than the SMS maximum size, it shall be split into several envelopes: SMS concatenation shall be used.

Note that the first Transport Layer Protocol values present under the tag '8B' (referenced by the 3GPP TS 23.040 specification [5]) are informative: they may be freely adapted by the test tool provider if needed.

</td></tr>
<tr><td>

HTTPS_CONTENT

</td><td>

Generate an HTTPS POST message containing APDU commands. This method is used to ask the ISD-R or the MNO-SD to execute some scripts.

Parameters:

- *APDU1; APDU2…*

Here is the TLS record (TLS_APPLICATION) content (in ASCII) to send:

```
#HTTP_CODE_200
X-Admin-Protocol: globalplatform-remote-admin/1.0
Content-Type: application/vnd.globalplatform.card-content-
mgt;version=1.0
#X_ADMIN_NEXT_URI

{COMMAND_SCRIPT}
```

`{COMMAND_SCRIPT}` shall be:

```
'AE 80' +
  '22 {L}' + APDU1 +
  '22 {L}' + APDU2 + … +
  '00 00'
```

</td></tr>
</table>

Method name	Explanation

HTTPS_CONTENT_ISDP	Generate an HTTPS POST message containing some commands (i.e. ADPUs or TLVs) to the ISD-P.
	
	Parameters:
	• *ISD_P_TARGETED_AID*
	• *COMMAND1; COMMAND2…*(i.e. APDUs or TLVs)
	• *CHAINING_OPT* (optional parameter)
	
	Here is the TLS record (TLS_APPLICATION) content (in ASCII) to send:
	`#HTTP_CODE_200`
	`X-Admin-Protocol: globalplatform-remote-admin/1.0`
	`Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0`
	`#X_ADMIN_NEXT_URI`
	`X-Admin-Targeted-Application:` *ISD_P_TARGETED_AID*
	
	`{COMMAND_SCRIPT}`
	
	• If the commands list is composed of APDUs:
	173.
	`{COMMAND_SCRIPT}` shall contain the list of APDUs formatted using the expanded format with indefinite length as defined in ETSI TS 102 226 [6].
	174.
	If *CHAINING_OPT* is not set, the `{COMMAND_SCRIPT}` shall be:
	`'AE 80' +`
	`'22 {L}' +` *COMMAND1* `+`
	`'22 {L}' +` *COMMAND2* `+ … +`
	`'00 00'`
	
	If *CHAINING_OPT* is set, the `{COMMAND_SCRIPT}` shall be:
	`'AE 80' +`
	`'83 01' +` *CHAINING_OPT* `+`
	`'22 {L}' +` *COMMAND1* `+`
	`'22 {L}' +` *COMMAND2* `+ … +`
	`'00 00'`
	175.
	• If the commands list is composed of TLVs (e.g. SCP03t commands):
	
	`{COMMAND_SCRIPT}` shall contain the list of TLVs formatted using the expanded format with indefinite length as defined in ETSI TS 102 226 [6].
	
	If *CHAINING_OPT* is not set, the `{COMMAND_SCRIPT}` shall be:
	`'AE 80' +`
	COMMAND1 `+`
	COMMAND2 `+ … +`
	`'00 00'`
	
	If *CHAINING_OPT* is set, the `{COMMAND_SCRIPT}` shall be:
	`'AE 80' +`
	`'83 01' +` *CHAINING_OPT* `+`
	COMMAND1 `+`
	COMMAND2 `+ … +`

Method name	Explanation
	'00 00'
SCP03_SCRIPT	Generate an SCP03 script with the APDUs in parameters. Parameters: - *KVN* - *APDU1; APDU2;…;APDUn* Here is the SCP03 script to generate: '80 50' + *KVN* + '00 08 {HOST_CHALLENGE} 00' '84 82 33 00 10 {HOST_CRYPTOGRAM} {MAC}' '{APDU1_SECURED}' '{APDU2_SECURED}' '…' '{APDUn_SECURED}' See Annex C for the definition of {HOST_CHALLENGE}, {HOST_CRYPTOGRAM} and {MAC}. The {APDUx_SECURED} is the command *APDUx* secured according GlobalPlatform Card Specification - Amendment D [11]. If it is not defined differently in the test step, these following SCP03 keys shall be used: - #DEFAULT_ISD_P_SCP03_KENC - #DEFAULT_ISD_P_SCP03_KMAC - #DEFAULT_ISD_P_SCP03_KDEK In order to retrieve the SCP03 sequence counter (i.e. {SCP03_SEQ_NUM}), it is assumed that a INITIALIZE UPDATE APDU command may be used every time it is necessary.
SCP03_SUB_SCRIPT	Generate the next part of an SCP03 script. Parameters: - *APDU1; APDU2;…APDUn* Here is the SCP03 script to generate: '{APDU1_SECURED}' '{APDU2_SECURED}' '…' '{APDUn_SECURED}' The {APDUx_SECURED} is the command *APDUx* secured according GlobalPlatform Card Specification - Amendment D [11]. The SCP03 session keys of the previous generated script shall be used.

Method name	Explanation

SCP03T_SCRIPT	Generate an SCP03t script with the PEs in parameters encoded in TLV structures using DER.
	
	Parameters:
	• *KVN*
	• *PE_TLVs*
	
	The *PE_TLVs* shall be split in several parts: each of these sub-parts (named `PE_TLV1, PE_TLV2 … PE_TLVn` here after) shall have a size which does not exceed 1007 bytes (considering that the maximum length of a SCP03t TLV command shall be 1020 bytes).
	
	Here is the SCP03t script to generate:
	
	`'84 0A' + KVN + '00 {HOST_CHALLENGE}'`
	`'85 11 33 {HOST_CRYPTOGRAM} {MAC}'`
	`'86 {L} {PE_TLV1_SECURED}'`
	`'86 {L} {PE_TLV2_SECURED}'`
	`'…'`
	`'86 {L} {PE_TLVn_SECURED}'`
	
	See Annex C for the definition of `{HOST_CHALLENGE}, {HOST_CRYPTOGRAM}` and `{MAC}`.
	
	The `{PE_TLVx_SECURED}` is the `PE_TLVx` secured according GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2] (section 4.1.3.3).
	
	If it is not defined differently in the test step, these following SCP03 keys shall be used:
	• `#DEFAULT_ISD_P_SCP03_KENC`
	• `#DEFAULT_ISD_P_SCP03_KMAC`
	• `#DEFAULT_ISD_P_SCP03_KDEK`
	
	In order to retrieve the SCP03 sequence counter (i.e. `{SCP03_SEQ_NUM}`), it is assumed that a INITIALIZE UPDATE TLV command may be used every time it is necessary.

SCP03T_SUB_SCRIPT	Generate the next part of an SCP03t script. Parameters: • *PE_TLVs* The *PE_TLVs* shall be split in several parts: each of these sub-parts (named `PE_TLV1, PE_TLV2 … PE_TLVn` here after) shall have a size which does not exceed 1007 bytes (considering that the maximum length of a SCP03t TLV command shall be 1020 bytes). Here is the SCP03t script to generate:

Method name	Explanation
	'86 {L} {PE_TLV1_SECURED}' '86 {L} {PE_TLV2_SECURED}' '…' '86 {L} {PE_TLVn_SECURED}' The {PE_TLVx_SECURED} is the PE_TLVx secured according GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2] (section 4.1.3.3). The SCP03 session keys of the previous generated script shall be used.
SCP80_PACKET	Generate an SCP80 secured packet with the commands (i.e. ADPUs or TLVs) in parameters. Parameters: • *SPI* • *TAR* • *COMMAND1; COMMAND2…*(i.e. APDUs or TLVs) • *CHAINING_OPT* (optional parameter) Here is the content of the command packet to generate: '{CPI} {L} 15' + *SPI* + '{KIC} {KID}' + *TAR* + '{CNTR} {PCNTR} {CC} {COMMAND_SCRIPT}' See Annex C for the definition of {CPI}, {KIC}, {KID}, {CNTR}, {PCNTR} and {CC}. For KIC and KID, if the KVN to use is '06' (for example), the value shall be '62' (AES in CBC mode). The KVN used shall be either #SCP80_KVN or #MNO_SCP80_KVN (depending of the targeted SD). Note that if the TAR is equal to #MNO_TAR, the algorithm used may be also Triple DES in outer-CBC depending of the Profile (i.e. KIC and KID shall be adapted in consequence). {CNTR} shall be incremented each time this function is called. • If the commands list is composed of one TLV which is either [OPEN_SCP81_SESSION] or [OPEN_SCP81_MNO_SESSION] (i.e. SCP81 administration session triggering parameters): {COMMAND_SCRIPT} shall contain the TLV command. • If the commands list is composed of APDUs: {COMMAND_SCRIPT} shall contain the list of APDUs formatted using the expanded format with definite length as defined in ETSI TS 102 226 [6]. If *CHAINING_OPT* is not set, the {COMMAND_SCRIPT} shall be: 'AA {L}' + '22 {L}' + *COMMAND1* + '22 {L}' + *COMMAND2* … If *CHAINING_OPT* is set, the {COMMAND_SCRIPT} shall be: 'AA {L}' + '83 01' + *CHAINING_OPT* + '22 {L}' + *COMMAND1* + '22 {L}' + *COMMAND2* …

Method name	Explanation

	• If the commands list is composed of TLVs (e.g. SCP03t commands):
	176.
	`{COMMAND_SCRIPT}` shall contain the list of TLVs formatted using the expanded format with definite length as defined in ETSI TS 102 226 [6].
	
	If *CHAINING_OPT* is not set, the `{COMMAND_SCRIPT}` shall be:
	`'AA {L}' +`
	`  `*`COMMAND1`* `+`
	`  `*`COMMAND2`* `…`
	
	If *CHAINING_OPT* is set, the `{COMMAND_SCRIPT}` shall be:
	`'AA {L}' +`
	`  '83 01' + `*`CHAINING_OPT`*` +`
	`  `*`COMMAND1`* `+`
	`  `*`COMMAND2`* `…`
	
	In any cases, this packet shall be secured according the `SPI` value.
	If it is not defined differently in the test step, these following SCP80 keys shall be used:
	• `#SCP80_ENC_KEY`
	• `#SCP80_AUTH_KEY`
	• `#SCP80_DATA_ENC_KEY`

SEND_ERROR_RESP	Send a secured error response message for a given request using network to an off-card entity.
	
	Parameters:
	• *FUNCTION_NAME*
	• *STATUS*
	• *SUBJECT_CODE*
	• *REASON_CODE*
	• *OUT_DATA1, OUT_DATA2…* (optional parameter)
	
	Here is the content of the response to answer:
	
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<RPSMessage xmlns="http://namespaces.gsma.org/esim-messaging/1"`
	`    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
	`    MessageVersion="1.0.0">`
	
	`    <RPSHeader>`
	`        <SenderEntity>`
	`            <EntityId>{FUNCTION_REQ_ID}</EntityId>`
	`        </SenderEntity>`
	`        <SenderName>{TOOL_NAME}</SenderName>`
	
	`        <ReceiverEntity>`
	`            <EntityId>{FUNCTION_REC_ID}</EntityId>`
	`        </ReceiverEntity>`
	
	`        <MessageId>{REQ_MESSAGE_ID}</MessageId>`
	`        <RelatesTo>{REL_MESSAGE_ID}</RelatesTo>`
	`        <MessageType>`*`FUNCTION_NAME`*`</MessageType>`
	`        <MessageDate>{CURRENT_DATE}</MessageDate>`
	`    </RPSHeader>`
	
	`    <RPSBody>`
	`        <`*`FUNCTION_NAME`*`>`
	`            <ProcessingStart>{CURRENT_DATE}</ProcessingStart>`

<table>
<tr><th>Method name</th><th>Explanation</th></tr>
<tr><td></td><td>

```
            <ProcessingEnd>{CURRENT_DATE}</ProcessingEnd>
            <FunctionExecutionStatus>
                <Status>STATUS</Status>
                <StatusCodeData>
                     <Subject>SUBJECT_CODE</Subject>
                     <Reason>REASON_CODE</Reason>
                </StatusCodeData>
           </FunctionExecutionStatus>
            OUT_DATA1
            OUT_DATA2
            …
        </FUNCTION_NAME>
    </RPSBody>
</RPSMessage>
```

See Annex C for the definition of `{CURRENT_DATE}`, `{FUNCTION_REQ_ID}` and `{FUNCTION_REC_ID}`.

The mapping of this function into message shall be compliant with the Annex A of the GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2].
To transport the message, the technology of the entity under test shall be used (mail, file, Web Services…).

Depending of the receiver of this message, the endpoint shall be either the `#SM_DP_ACCESSPOINT` or the `#SM_SR_ACCESSPOINT`.

</td></tr>
<tr><td>

SEND_NOTIF

</td><td>

Send a secured notification message using network to an off-card entity.

Parameters:

- *NOTIF_NAME*
- *IN_DATA1; IN_DATA2…*

Here is the message to send:

```
<?xml version="1.0" encoding="UTF-8"?>
<RPSMessage xmlns="http://namespaces.gsma.org/esim-messaging/1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    MessageVersion="1.0.0">

    <RPSHeader>
        <SenderEntity>
            <EntityId>{FUNCTION_REQ_ID}</EntityId>
            <EntityName>{TOOL_NAME}</EntityName>
        </SenderEntity>
        <SenderName>{TOOL_NAME}</SenderName>

        <ReceiverEntity>
            <EntityId>{FUNCTION_REC_ID}</EntityId>
        </ReceiverEntity>

        <MessageId>{MESSAGE_ID}</MessageId>
        <MessageType>NOTIF_NAME</MessageType>
        <MessageDate>{CURRENT_DATE}</MessageDate>
    </RPSHeader>

    <RPSBody>
        <NOTIF_NAME>
            <FunctionCallIdentifier>
                 {FUNC_CALL_ID}
            </FunctionCallIdentifier>
            IN_DATA1
```

</td></tr>
</table>

<table>
<tr><th>Method name</th><th>Explanation</th></tr>
<tr><td></td><td>

```
                IN_DATA2
                …
            </NOTIF_NAME>
        </RPSBody>
</RPSMessage>
```

See Annex C for the definition of `{CURRENT_DATE}`, `{FUNCTION_REQ_ID}` and `{FUNCTION_REC_ID}`.

To transport the message, the technology of the entity under test shall be used (mail, file, Web Services…).

Depending of the receiver of this message, the endpoint shall be either the `#SM_DP_ACCESSPOINT` or the `#SM_SR_ACCESSPOINT`.

</td></tr>
<tr><td><i>SEND_REQ</i></td><td>

Send a secured request message using network to an off-card entity.

Parameters:

- *FUNCTION_NAME*
- *IN_DATA1; IN_DATA2…*

Here is the content of the request to send:

```
<?xml version="1.0" encoding="UTF-8"?>
<RPSMessage xmlns="http://namespaces.gsma.org/esim-messaging/1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    MessageVersion="1.0.0">

    <RPSHeader>
        <SenderEntity>
            <EntityId>{FUNCTION_REQ_ID}</EntityId>
            <EntityName>{TOOL_NAME}</EntityName>
        </SenderEntity>
        <SenderName>{TOOL_NAME}</SenderName>

        <ReceiverEntity>
            <EntityId>{FUNCTION_REC_ID}</EntityId>
        </ReceiverEntity>

        <MessageId>{MESSAGE_ID}</MessageId>
        <MessageType>FUNCTION_NAME</MessageType>
        <MessageDate>{CURRENT_DATE}</MessageDate>
    </RPSHeader>

    <RPSBody>
        <FUNCTION_NAME>
            <FunctionCallIdentifier>
               {FUNC_CALL_ID}
            </FunctionCallIdentifier>
            IN_DATA1
            IN_DATA2
            …
        </FUNCTION_NAME>
    </RPSBody>
</RPSMessage>
```

See Annex C for the definition of `{CURRENT_DATE}`, `{FUNC_CALL_ID}`, `{FUNCTION_REQ_ID}` and `{FUNCTION_REC_ID}`.

</td></tr>
</table>

Method name	Explanation
	The mapping of this function into message shall be compliant with the Annex A of the GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2]. To transport the message, the technology of the entity under test shall be used (mail, file, Web Services…). Depending of the receiver of this message, the endpoint shall be either the `#SM_DP_ACCESSPOINT` or the `#SM_SR_ACCESSPOINT`. If needed, the attribute `ResponseEndpoint` may be used.

SEND_SUCCESS_RESP	Send a secured success response message for a given request using network to an off-card entity.
	Parameters:
	• *FUNCTION_NAME*
	• *OUT_DATA1; OUT_DATA2…* (optional parameter)
	Here is the content of the response to answer:
	<?xml version="1.0" encoding="UTF-8"?>
	<RPSMessage xmlns="http://namespaces.gsma.org/esim-messaging/1"
	    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	    MessageVersion="1.0.0">
	    <RPSHeader>
	        <SenderEntity>
	            <EntityId>{FUNCTION_REQ_ID}</EntityId>
	        </SenderEntity>
	        <SenderName>{TOOL_NAME}</SenderName>
	        <ReceiverEntity>
	            <EntityId>{FUNCTION_REC_ID}</EntityId>
	        </ReceiverEntity>
	        <MessageId>{REQ_MESSAGE_ID}</MessageId>
	        <RelatesTo>{REL_MESSAGE_ID}</RelatesTo>
	        <MessageType>*FUNCTION_NAME*</MessageType>
	        <MessageDate>{CURRENT_DATE}</MessageDate>
	    </RPSHeader>
	    <RPSBody>
	        <*FUNCTION_NAME*>
	            <ProcessingStart>{CURRENT_DATE}</ProcessingStart>
	            <ProcessingEnd>{CURRENT_DATE}</ProcessingEnd>
	            <FunctionExecutionStatus>
	                <Status>#SUCCESS</Status>
	            </FunctionExecutionStatus>
	            OUT_DATA1
	            OUT_DATA2
	            …
	        </*FUNCTION_NAME*>
	    </RPSBody>
	</RPSMessage>
	See Annex C for the definition of `{CURRENT_DATE}`, `{FUNCTION_REQ_ID}` and `{FUNCTION_REC_ID}`.
	The mapping of this function into message shall be compliant with the Annex A of the GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2].
	To transport the message, the technology of the entity under test shall be used (mail, file, Web Services…).

<table>
<tr><th>Method name</th><th>Explanation</th></tr>
<tr><td></td><td>Depending of the receiver of this message, the endpoint shall be either the #SM_DP_ACCESSPOINT or the #SM_SR_ACCESSPOINT.</td></tr>
<tr><td><i>STORE_ISDP_KEYS</i></td><td>Generate the APDU command allowing the creation or the update of the ISD-P keys (scenario#3 based on ECKA EG (ElGamal) scheme as defined in GlobalPlatform Card Specification Amendment E [12]).

Parameters:
• *SC3_PARAM*
• *RANDOM_CHALLENGE*

Here is the content of the APDU to generate:<pre>
- CLA = 80
- INS = E2
- P1 = 89
- P2 = 01
- LC = {LC}
- Data =
   '3A 02 {L}
      A6 {L}
        90 02 03' + SC3_PARAM +
       '95 01 10
        80 01 88
        81 01 10
        82 01 01
        83 01 #SCP03_KVN
        91 00
        84 {L} #HOST_ID (present only if SC3_PARAM=#SC3_DR_HOST)
    7F 49 {L} #SM_EPK_ECKA'
    5F 37 {L} {SIGNATURE}
- LE = 00
</pre>The following TLV-encoded data shall be signed with #SM_SK_ECDSA to generate the {SIGNATURE}:<pre>
'3A 02 {L}
    A6 {L}
      90 02 03' + SC3_PARAM +
     '95 01 10
      80 01 88
      81 01 10
      82 01 01
      83 01 #SCP03_KVN
      91 00
      84 {L} #HOST_ID (present only if SC3_PARAM=#SC3_DR_HOST)
7F 49 {L} #SM_EPK_ECKA
00 85 {L}' + RANDOM_CHALLENGE
</pre></td></tr>
<tr><td><i>STORE_ISDR_KEYS</i></td><td>Generate the APDU command allowing the creation or the update of the ISD-R keys (scenario#3 based on ECKA EG (ElGamal) scheme as defined in GlobalPlatform Card Specification Amendment E [12]).

Parameters:
• *SC3_PARAM*
• *RANDOM_CHALLENGE*

Here is the content of the APDU to generate:<pre>
- CLA = 80
- INS = E2
</pre></td></tr>
</table>

Method name	Explanation
	- P1 = 89 - P2 = 01 - LC = {LC} - Data = '3A 02 {L} A6 {L} 90 02 03' + *SC3_PARAM* + '95 01 10 80 01 88 81 01 10 82 01 01 83 01 #SCP80_KVN 91 05 00 00 00 00 01 84 {L} #HOST_ID (present only if *SC3_PARAM*=#SC3_DR_HOST) 7F 49 {L} #SM_EPK_ECKA' 5F 37 {L} {SIGNATURE} - LE = 00 The following TLV-encoded data shall be signed with #SM_SK_ECDSA to generate the {SIGNATURE}: '3A 02 {L} A6 {L} 90 02 03' + *SC3_PARAM* + '95 01 10 80 01 88 81 01 10 82 01 01 83 01 #SCP80_KVN 91 05 00 00 00 00 01 84 {L} #HOST_ID (present only if *SC3_PARAM*=#SC3_DR_HOST) 7F 49 {L} #SM_EPK_ECKA 00 85 {L}' + *RANDOM_CHALLENGE*
STORE_MNO_KEYS_2B	Generate the APDU command that allows updating the MNO keys using the scenario#2.B as defined in GlobalPlatform Card Specification v.2.2.1 - UICC Configuration [13]. Parameters: - *CASD_PUBLIC_KEY* Here is the content of the APDU to generate: - CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {LC} - Data = 00 A6 18 A6 16 90 01 04 95 01 10 80 01 80 (MNO-SD shall be configured with 3DES keys) 81 01 10 83 01 #MNO_SCP80_KVN 91 05 00 00 00 00 01 80 10 {L} {KEYS_ENCRYPTED}

Method name	Explanation
	The `{KEYS_ENCRYPTED}` shall be encrypted with the *`CASD_PUBLIC KEY`*.

STORE_MNO_KEYS_3	Generate the APDU command that allows updating the MNO keys using the scenario#3 based on ECKA EG (ElGamal) scheme as defined in GlobalPlatform Card Specification Amendment E [12].
	Parameters:
	- *None*
	Here is the content of the APDU to generate:
	`- CLA = 80`
	`- INS = E2`
	`- P1 = 89`
	`- P2 = 00`
	`- LC = {LC}`
	`- Data =`
	`   00 A6 1C`
	`      A6 1A`
	`        90 02 03 01`
	`        95 01 10`
	`        80 01 80 (or '88' if the MNO-SD is configured with AES keys)`
	`        81 01 10`
	`        82 01 01`
	`        83 01 #MNO_SCP80_KVN`
	`        91 05 00 00 00 00 01`
	`    7F 49 {L} #SM_EPK_ECKA`
	`- LE = 00`

Table 18: Methods

Annex E Commands and Responses

Here are all the commands and responses used in this document.

E.1 Commands

Name	Content in hexadecimal string
BAD_MASTER_DEL_ISDP1	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = 33 - Data = 4F 10 #ISD_P_AID1 B6 1A 42 04 #ISD_P_SIN 45 08 #ISD_P_SDIN 5F 20 04 #ISD_P_PROV_ID 93 01 #TOKEN_ID 9E 03 #BAD_TOKEN - LE = 00
BAD_STORE_POL1	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 06 - Data = 3A 06 03 81 01 07
DELETE_ISDP1	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = 12 - Data = 4F 10 #ISD_P_AID1 - LE = 00
DELETE_SCP80_KEYSETS	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 00 - LC = 05 - Data = F2 03 #SCP03_KVN 01 03 - LE = 00
DELETE1_KEYSETS	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 00 - LC = 05 - Data = F2 03 #SCP80_KVN 01 03 - LE = 00
DELETE2_KEYSETS	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 00 - LC = 0A - Data = F2 03 #SCP80_KVN 01 03 F2 03 #SCP81_KVN 01 05 - LE = 00

Name	Content in hexadecimal string
DISABLE_ISDP1	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 15 - Data = 3A 04 12 4F 10 #ISD_P_AID1
ENABLE_ISDP1	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 15 - Data = 3A 03 12 4F 10 #ISD_P_AID1
GET_DATA_5A	- CLA = 80 - INS = CA - P1 = 00 - P2 = 5A - LE = 00
GET_DATA_BF30_CERT	- CLA = 80 - INS = CA - P1 = BF - P2 = 30 - LC = 04 - Data = 5C 02 7F 21 - LE = 00
GET_DATA_BF30_REC	- CLA = 80 - INS = CA - P1 = BF - P2 = 30 - LC = 03 - Data = 5C 01 66 - LE = 00
GET_DATA_C1	- CLA = 80 - INS = CA - P1 = 00 - P2 = C1 - LE = 00
GET_DATA_CASD_CERT	- CLA = 80 - INS = CA - P1 = 7F - P2 = 21 - LE = 00
GET_DATA_E0	- CLA = 80 - INS = CA - P1 = 00 - P2 = E0 - LE = 00
GET_DATA_FF21	- CLA = 80 - INS = CA - P1 = FF - P2 = 21 - LE = 00

Name	Content in hexadecimal string
GET_DEFAULT_ISDP	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 17 - Data = 4F 10 #DEFAULT_ISD_P_AID 5C 03 4F 9F 70 - LE = 00
GET_FALLBACK	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 09 - Data = 4F 00 #ISD_P_ATTRIBUTE 01 01 5C 02 4F #ISD_P_ATTRIBUTE - LE = 00
GET_ISDP1	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 17 - Data = 4F 10 #ISD_P_AID1 5C 03 4F 9F 70 - LE = 00
GET_ISDP1_MEM	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 19 - Data = 4F 10 #ISD_P_AID1 5C 05 4F 9F 70 8F 91 - LE = 00
GET_ISDP_DISABLED	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 0B - Data = 4F 00 9F 70 01 1F 5C 03 4F 9F 70 - LE = 00
GET_ISDP_ENABLED	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 0B - Data = 4F 00 9F 70 01 3F 5C 03 4F 9F 70 - LE = 00
GET_ISDP_LIST	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 07 - Data = 4F 00 5C 03 4F 9F 70 - LE = 00

Name	Content in hexadecimal string

GET_MNO_ISD	`- CLA = 80`
	`- INS = F2`
	`- P1 = 80`
	`- P2 = 02`
	`- LC = 07`
	`- Data = 4F 00 5C 03 4F 9F 70`
	`- LE = 00`

GET_MNO_SD	`- CLA = 80`
	`- INS = F2`
	`- P1 = 40`
	`- P2 = 02`
	`- LC = {L}`
	`- Data = 4F {L} #MNO_SD_AID 5C 01 4F`
	`- LE = 00`

GET_STATUS_ISDR	`- CLA = 80`
	`- INS = F2`
	`- P1 = 40`
	`- P2 = 02`
	`- LC = 12`
	`- Data = 4F 10 #ISD_R_AID`
	`- LE = 00`

INSTALL_AID_ECASD	`- CLA = 80`
	`- INS = E6`
	`- P1 = 0C`
	`- P2 = 00`
	`- LC = 2C`
	`- Data =`
	`       08 A0 00 00 05 59 10 10 03`
	`       0B A0 00 00 05 59 10 10 03 44 55 66`
	`       10 #ECASD_AID`
	`       01 00`
	`       02 C9 00`
	`       00`
	`-LE = 00`

INSTALL_TAR_ISDR	`- CLA = 80`
	`- INS = E6`
	`- P1 = 0C`
	`- P2 = 00`
	`- LC = 37`
	`- Data =`
	`       08 A0 00 00 05 59 10 10 01`
	`       0B A0 00 00 05 59 10 10 01 11 22 33`
	`       0C A0 00 00 05 59 10 10 01 11 22 33 01`
	`       01 00`
	`       11 EA 0D 80 0B 01 00 00 00 00 00 03 #ISD_R_TAR 00`
	`          C9 00`
	`       00`
	`-LE = 00`

Name	Content in hexadecimal string
INSTALL_APPLET2	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 37 - Data = 08 A0 00 00 05 59 10 10 02 0B A0 00 00 05 59 10 10 02 11 22 33 0C A0 00 00 05 59 10 10 02 11 22 33 01 01 00 11 EA 0D 80 0B 01 00 00 00 00 00 03 11 22 33 00 C9 00 00 -LE = 00
INSTALL_APPLET3	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 28 - Data = 08 A0 00 00 05 59 10 10 03 0B A0 00 00 05 59 10 10 03 44 55 66 0C A0 00 00 05 59 10 10 03 44 55 66 01 01 00 02 C9 00 00 -LE = 00
INSTALL_ISDP	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 3F - Data = 10 #ISD_P_PKG_AID 10 #ISD_P_MOD_AID 10 #ISD_P_AID1 03 80 C0 00 06 C9 04 81 02 03 70 00 -LE = 00
INSTALL_ISDP_MEM	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 47 - Data = 10 #ISD_P_PKG_AID 10 #ISD_P_MOD_AID 10 #ISD_P_AID1 03 80 C0 00 0E EF 06 83 04 #MEMORY_QUOTA C9 04 81 02 03 70 00 - LE = 00
INSTALL_PERSO_RES_ISDP	- CLA = 80 - INS = E6 - P1 = 20 - P2 = 00 - LC = 16 - Data = 00 00 10 #RESERVED_ISD_P_AID 00 00 00 - LE = 00

Name	Content in hexadecimal string
INSTALL_PERSO_ISDP1	- CLA = 80 - INS = E6 - P1 = 20 - P2 = 00 - LC = 16 - Data = 00 00 10 #ISD_P_AID1 00 00 00 - LE = 00
LOCK_DEFAULT_ISDP	- CLA = 80 - INS = F0 - P1 = 40 - P2 = 80 - LC = 10 - Data = #DEFAULT_ISD_P_AID
LOCK_ISDR	- CLA = 80 - INS = F0 - P1 = 80 - P2 = 7F - LC = 10 - Data = #ISD_R_AID

MASTER_DEL_ISDP1	- CLA = 80
	- INS = E4
	- P1 = 00
	- P2 = 40
	- LC = 40
	- Data =
	  4F 10 #ISD_P_AID1
	  B6 1A
	    42 04 #ISD_P_SIN
	    45 08 #ISD_P_SDIN
	    5F 20 04 #ISD_P_PROV_ID
	    93 01 #TOKEN_ID
	  9E 10 {TOKEN_VALUE}
	- LE = 00

NOTIF_CONFIRMATION	- CLA = 80 - INS = E2 - P1 = 89 - P2 = 00 - LC = 07 - Data = 3A 08 04 4E 02 {NOTIF_NUMBER} - LE = 00

NOTIF_PROFILE_CHANGE	E1 {L}
	  4C 10 #EID
	  4D 01 02
	  4E 02 {NOTIF_NUMBER}
	  2F 10 #ISD_P_AID1
	
	see Note 1

NOTIF_PROFILE_CHANGE2	E1 {L}
	  4C 10 #EID
	  4D 01 02
	  4E 02 {NOTIF_NUMBER}
	  2F 10 #DEFAULT_ISD_P_AID
	
	see Note 1

Name	Content in hexadecimal string
NOTIF_ROLL_BACK	E1 {L} 4C 10 #EID 4D 01 03 4E 02 {NOTIF_NUMBER} 2F 10 #DEFAULT_ISD_P_AID see Note 1
OPEN_CHANNEL_FOR_BIP	- CLA = 80 - INS = EC - P1 = 01 - P2 = 01 - LC = 25 - Data = 35 07 #BEARER_DESCRIPTION 3C 03 01 #UDP_PORT 39 02 #BUFFER_SIZE 47 0A #NAN_VALUE 3E 05 21 #IP_VALUE
OPEN_CHANNEL_FOR_CATTP	- CLA = 80 - INS = EC - P1 = 01 - P2 = 02 - LC = 05 - Data = 3C 03 00 #CAT_TP_PORT
OPEN_SCP81_MNO_SESSION	81 {L} 83 {L} 84 25 35 07 #BEARER_DESCRIPTION 39 02 #BUFFER_SIZE 47 0A #NAN_VALUE 3C 03 02 #TCP_PORT 3E 05 21 #IP_VALUE 89 {L} 8A 09 #ADMIN_HOST 8B {L} #MNO_AGENT_ID 8C 10 #ADMIN_URI 85 {L} {L} #MNO_PSK_ID 02 #MNO_SCP81_KVN #MNO_SCP81_KEY_ID
OPEN_SCP81_SESSION	81 {L} 83 {L} 84 25 35 07 #BEARER_DESCRIPTION 39 02 #BUFFER_SIZE 47 0A #NAN_VALUE 3C 03 02 #TCP_PORT 3E 05 21 #IP_VALUE 89 {L} 8A 09 #ADMIN_HOST 8B {L} #AGENT_ID 8C 10 #ADMIN_URI

Name	Content in hexadecimal string
SELECT_APPLET3	- CLA = 00 - INS = A4 - P1 = 04 - P2 = 00 - LC = 0C - Data = A0 00 00 05 59 10 10 03 44 55 66 01 - LE = 00
SELECT_CASD	- CLA = 00 - INS = A4 - P1 = 04 - P2 = 00 - LC = 0C - Data = #CASD_AID - LE = 00
SELECT_ECASD	- CLA = 00 - INS = A4 - P1 = 04 - P2 = 00 - LC = 10 - Data = #ECASD_AID - LE = 00
SELECT_FILE_1122	- CLA = 00 - INS = A4 - P1 = 00 - P2 = 04 - LC = 02 - Data = 11 22 - LE = 00
SET_FALLBACK	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 15 - Data = 3A 05 12 4F 10 #ISD_P_AID1
STORE_CATTP_PARAM	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 16 - Data = 3A 07 13 A4 11 3C 03 01 #UDP_PORT 3C 03 00 #CAT_TP_PORT 3E 05 21 #IP_VALUE
STORE_CATTP_PARAM_MNO	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 2D - Data = 3A 07 2A A2 28 35 07 #BEARER_DESCRIPTION 47 0A #NAN_VALUE 0D 06 #LOGIN 0D 09 #PWD

Name	Content in hexadecimal string
STORE_CATTP_PARAM_MNO2	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A2 {L} 35 07 #BEARER_DESCRIPTION 47 {L} #MNO2_CON_NAN 0D {L} #MNO2_CON_LOGIN 0D {L} #MNO2_CON_PWD
STORE_DP_CERTIF	- CLA = 80 - INS = E2 - P1 = 09 - P2 = 00 - LC = {LC} - Data = 3A 01 {L} #VALID_SM_DP_CERTIFICATE - LE = 00

STORE_HTTPS_PARAM	- CLA = 80
	- INS = E2
	- P1 = 90
	- P2 = 00
	- LC = {L}
	- Data =
	  A5 {L}
	   84 0C
	      3C 03 02 #TCP_PORT
	      3E 05 21 #IP_VALUE
	   89 {L}
	      8A 09 #ADMIN_HOST
	      8B {L} #AGENT_ID
	      8C 10 #ADMIN_URI

STORE_HTTPS_PARAM_MNO	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 2D - Data = 3A 07 2A A1 28 35 07 #BEARER_DESCRIPTION 47 0A #NAN_VALUE 0D 06 #LOGIN 0D 09 #PWD
STORE_HTTPS_PARAM_MNO2	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A1 {L} 35 07 #BEARER_DESCRIPTION 47 {L} #MNO2_CON_NAN 0D {L} #MNO2_CON_LOGIN 0D {L} #MNO2_CON_PWD

Name	Content in hexadecimal string
STORE_INVALID_DP_CERTIF	- CLA = 80 - INS = E2 - P1 = 09 - P2 = 00 - LC = {LC} - Data = 3A 01 {L} #INVALID_SM_DP_CERTIFICATE - LE = 00
STORE_INVALID_SR_CERTIF	- CLA = 80 - INS = E2 - P1 = 09 - P2 = 00 - LC = {LC} - Data = 3A 01 {L} #INVALID_SM_SR_CERTIFICATE - LE = 00
STORE_POL1_DEL_AUTO	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 06 - Data = 3A 06 03 81 01 04
STORE_POL1_DEL_DIS	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 06 - Data = 3A 06 03 81 01 03
STORE_POL1_DIS	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 06 - Data = 3A 06 03 81 01 01
STORE_POL1_NO_RULE	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 06 - Data = 3A 06 03 81 01 00
STORE_PROV_ID	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 0A - Data = 00 70 07 5F 20 04 #ISD_P_PROV_ID
STORE_SDIN	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 0D - Data = 00 70 0A 45 08 #ISD_P_SDIN

Name	Content in hexadecimal string
STORE_SIN	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 09 - Data = 00 70 06 42 04 #ISD_P_SIN
STORE_SMS_PARAM	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 0C - Data = 3A 07 09 A3 07 81 05 #DEST_ADDR

STORE_SMSCATTP_PARAM	- CLA = 80
	- INS = E2
	- P1 = 88
	- P2 = 00
	- LC = 38
	- Data =
	   3A 07 35
	      A0 09 06 07 #TON_NPI #DIALING_NUMBER
	      A2 28
	         35 07 #BEARER_DESCRIPTION
	         47 0A #NAN_VALUE
	         0D 06 #LOGIN
	         0D 09 #PWD

STORE_SMSHTTPS_PARAM	- CLA = 80
	- INS = E2
	- P1 = 88
	- P2 = 00
	- LC = 38
	- Data =
	   3A 07 35
	      A0 09 06 07 #TON_NPI #DIALING_NUMBER
	      A1 28
	         35 07 #BEARER_DESCRIPTION
	         47 0A #NAN_VALUE
	         0D 06 #LOGIN
	         0D 09 #PWD

STORE_SMS_PARAM_MNO	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 0E - Data = 3A 07 0B A0 09 06 07 #TON_NPI #DIALING_NUMBER
STORE_SMS_PARAM_MNO2	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A0 {L} 06 {L} #MNO2_CON_TON_NPI #MNO2_CON_DIAL_NUM

Name	Content in hexadecimal string

STORE_SR_CERTIF	`- CLA = 80`
	`- INS = E2`
	`- P1 = 09`
	`- P2 = 00`
	`- LC = {LC}`
	`- Data = 3A 01 {L} #VALID_SM_SR_CERTIFICATE`
	`- LE = 00`

TERMINAL_PROFILE	`- CLA = 80`
	`- INS = 10`
	`- P1 = 00`
	`- P2 = 00`
	`- LC = 1F`
	`- Data =`
	`FF FF FF FF FF FF 1F FF FF 03 02 FF FF 9F FF EF DF FF`
	`0F FF 0F FF FF 0F FF 03 00 3F 7F FF 03`

Note 1: The AID tag that allows identifying the ISD-P may be either ‘2F’ or ‘AF’. The different TLV data objects within the tag ‘E1’ may be returned with a different order. Moreover, the TLV notification may also contain proprietary tags. However, the entire TLV shall fit into one SMS-MO if the notification is sent over SMS, and shall not exceed the size of 240 bytes if sent by HTTPs or CAT_TP.	

Table 19: Commands

E.2 Responses

Name	Content in hexadecimal string

R_AB_009000	`AB 09`
	`80 02 00 01`
	`23 03 00 90 00`
	see Note 2

R_AB_026982	`AB 08`
	`80 02 00 02`
	`23 02 69 82`
	see Note 2

R_AB_026A80	`AB 0D`
	`80 02 00 02`
	`23 03 00 90 00`
	`23 02 6A 80`
	see Note 2

R_AB_029000	`AB 0D`
	`80 02 00 02`
	`23 03 00 90 00`
	`23 02 90 00`
	see Note 2

R_AB_02RC	`AB {L}`
	`80 02 00 02`
	`23 {L} 85 {L} {RC}`
	`90 00`
	see Note 2

Name	Content in hexadecimal string
R_AB_02RECEIPT	AB {L} 80 02 00 02 23 {L} 86 {L} {RECEIPT} 90 00 see Note 2
R_AB_02RECEIPT_DR	AB {L} 80 02 00 02 23 {L} 85 {L} {DR} 86 {L} {RECEIPT} 90 00 see Note 2
R_AB_036982	AB 0D 80 02 00 03 23 03 00 90 00 23 02 69 82 see Note 2
R_AB_03RC	AB {L} 80 02 00 03 23 03 00 90 00 23 {L} 85 {L} {RC} 90 00 see Note 2
R_AB_6985	AB 08 80 02 00 01 23 02 69 85 see Note 2
R_AB_69E1	AB 08 80 02 00 01 23 02 69 E1 see Note 2
R_AB_6A80	AB 08 80 02 00 01 23 02 6A 80 see Note 2
R_AB_6A88	AB 08 80 02 00 01 23 02 6A 88 see Note 2
R_AB_9000	AB 08 80 02 00 01 23 02 90 00 see Note 2

Name	Content in hexadecimal string

R_AB_BF30_ECASD	AB {L}
	  80 02 00 01
	  23 {L}
	    BF 30 {L}
	      7F 21 {L}
	        7F 21 {L} #ECASD_CERTIFICATE
	    90 00
	see Note 2

R_AB_BF30_REC	AB {L}
	  80 02 00 01
	  23 {L}
	    BF 30 {L}
	      66 {L} #CARD_RECOGNITION_DATA
	    90 00
	see Note 2

R_AB_E0_SCP80	AB 1C
	  80 02 00 01
	  23 16
	     E0 12
	        C0 04 01 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 02 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 03 #SCP80_KVN 88 {KEY_LENGTH}
	     90 00
	see Note 1
	see Note 2

R_AB_E0_SCP80_SCP81	AB 22
	  80 02 00 01
	  23 1C
	     E0 18
	        C0 04 01 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 02 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 03 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 #SCP81_KEY_ID #SCP81_KVN 85 {KEY_LENGTH}
	     90 00
	see Note 1
	see Note 2
	see Note 5

R_AB_E3_ISDP_3F	AB 20
	  80 02 00 01
	  23 1A
	    E3 16
	      4F 10 #DEFAULT_ISD_P_AID
	      9F 70 01 3F
	    90 00
	see Note 2

Name	Content in hexadecimal string
R_AB_E3_ISDP_LIST1	AB 3C 80 02 00 02 23 1A E3 16 4F 10 #ISD_P_AID1 9F 70 01 3F 90 00 23 1A E3 16 4F 10 #DEFAULT_ISD_P_AID 9F 70 01 1F 90 00 see Note 2
R_AB_E3_ISDP_LIST2	AB 3C 80 02 00 02 23 1A E3 16 4F 10 #ISD_P_AID1 9F 70 01 1F 90 00 23 1A E3 16 4F 10 #DEFAULT_ISD_P_AID 9F 70 01 3F 90 00 see Note 2
R_AB_E3_ISDP_LIST3	AB 38 80 02 00 01 23 32 E3 16 4F 10 #ISD_P_AID1 9F 70 01 1F E3 16 4F 10 #DEFAULT_ISD_P_AID 9F 70 01 3F 90 00 see Note 2
R_AB_E3_ISDP1_07	AB 20 80 02 00 01 23 1A E3 16 4F 10 #ISD_P_AID1 9F 70 01 07 90 00 see Note 2
R_AB_E3_ISDP1_0F	AB 20 80 02 00 01 23 1A E3 16 4F 10 #ISD_P_AID1 9F 70 01 0F 90 00 see Note 2

Name	Content in hexadecimal string

R_AB_E3_ISDP1_1F	AB 20
	  80 02 00 01
	  23 1A
	    E3 16
	      4F 10 #ISD_P_AID1
	      9F 70 01 1F
	    90 00
	see Note 2

R_AB_E3_ISDP1_3F	AB 20
	  80 02 00 01
	  23 1A
	    E3 16
	      4F 10 #ISD_P_AID1
	      9F 70 01 3F
	    90 00
	see Note 2

R_AB_E3_ISDP1_E1	AB 1F
	  80 02 00 01
	  23 19
	    E3 15
	      4F 10 #ISD_P_AID1
	      #ISD_P_ATTRIBUTE 01 01
	    90 00
	see Note 2

R_AB_E3_ISDP1_MEM	AB 2C
	  80 02 00 01
	  23 26
	    E3 22
	      4F 10 #ISD_P_AID1
	      9F 70 01 07
	      8F 04 #MEMORY_QUOTA
	      91 04 #MEMORY_QUOTA
	    90 00
	see Note 2
	see Note 4

R_AB_FF21	AB {L}
	  80 02 00 01
	  23 {L}
	    FF 21 {L}
	      81 {L} {NB_APP}
	      82 {L} {NON_VOLATILE_MEMORY}
	      83 {L} {VOLATILE_MEMORY}
	    90 00
	see Note 2

R_AB_MNO_SD	AB {L}
	  80 02 00 01
	  23 {L}
	    E3 {L}
	      4F {L} #MNO_SD_AID
	      9F 70 01 0F
	    90 00
	see Note 2
	see Note 3

Name	Content in hexadecimal string
R_AB_NOTIF	AB 0A 80 02 00 01 23 04 80 00 90 00 see Note 2
R_AB_NOTIF1	AB 1C 80 02 00 01 23 16 80 12 4F 10 #DEFAULT_ISD_P_AID 90 00 see Note 2
R_AB_NOTIF2	AB 1C 80 02 00 01 23 16 80 12 4F 10 #ISD_P_AID1 90 00 see Note 2
R_AB_RECEIPT	AB {L} 80 02 00 01 23 {L} 86 {L} {RECEIPT} 90 00 see Note 2
R_AB_SCP03T_01	AB 2C 80 02 00 03 [R_SCP03T_INITUP_OK] [R_SCP03T_EXTAUTH_OK] 9F 46 01 01 see Note 2
R_AB_SCP03T_02	AB 2C 80 02 00 03 [R_SCP03T_INITUP_OK] [R_SCP03T_EXTAUTH_OK] 9F 46 01 02 see Note 2
R_AB_SCP03T_EA_01	AB 2A 80 02 00 02 [R_SCP03T_INITUP_OK] 9F 45 01 01 see Note 2
R_AB_SCP03T_EA_02	AB 2A 80 02 00 02 [R_SCP03T_INITUP_OK] 9F 45 01 02 see Note 2
R_AB_SCP03T_IU_01	AB 08 80 02 00 01 9F 44 01 01 see Note 2

Name	Content in hexadecimal string

R_AB_SCP03T_IU_03	AB 08
	  80 02 00 01
	  9F 44 01 03
	see Note 2

R_AF_009000	AF 80
	  23 03 00 90 00
	  00 00

R_AF_029000	AF 80
	  23 03 00 90 00
	  23 02 90 00
	  00 00

R_AF_02RC	AF 80
	  23 03 00 90 00
	  23 {L} 85 {L} {RC} 90 00
	  00 00

R_AF_6A88	AF 80
	  23 02 6A 88
	  00 00

R_AF_9000	AF 80
	  23 02 90 00
	  00 00

R_AF_BF30_CERT	AF 80
	  23 {L}
	    BF 30 {L}
	      7F 21 {L}
	        7F 21 {L} #ECASD_CERTIFICATE
	    90 00
	  00 00

R_AF_BF30_REC	AF 80
	  23 {L}
	    BF 30 {L}
	      66 {L} #CARD_RECOGNITION_DATA
	    90 00
	  00 00

R_AF_E0_SCP80_SCP81	AF 80
	  23 1C
	     E0 18
	        C0 04 01 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 02 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 03 #SCP80_KVN 88 {KEY_LENGTH}
	        C0 04 #SCP81_KEY_ID #SCP81_KVN 85 {KEY_LENGTH}
	     90 00
	  00 00
	see Note 1
	see Note 5

R_AF_E3_ISDP_3F	AF 80
	  23 1A
	    E3 16
	      4F 10 #DEFAULT_ISD_P_AID
	      9F 70 01 3F
	    90 00
	  00 00

Name	Content in hexadecimal string

R_AF_E3_ISDP_LIST3	AF 80
	  23 32
	    E3 16
	      4F 10 #ISD_P_AID1
	      9F 70 01 1F
	    E3 16
	      4F 10 #DEFAULT_ISD_P_AID
	      9F 70 01 3F
	    90 00
	  00 00

R_AF_E3_ISDP1_07	AF 80
	  23 1A
	    E3 16
	      4F 10 #ISD_P_AID1
	      9F 70 01 07
	    90 00
	  00 00

R_AF_E3_ISDP1_0F	AF 80
	  23 1A
	    E3 16
	      4F 10 #ISD_P_AID1
	      9F 70 01 0F
	    90 00
	  00 00

R_AF_E3_ISDP1_1F	AF 80
	  23 1A
	    E3 16
	      4F 10 #ISD_P_AID1
	      9F 70 01 1F
	    90 00
	  00 00

R_AF_E3_ISDP1_E1	AF 80
	  23 19
	    E3 15
	      4F 10 #ISD_P_AID1
	      #ISD_P_ATTRIBUTE 01 01
	    90 00
	  00 00

R_AF_FF21	AF 80
	  23 {L}
	    FF 21 {L}
	      81 {L} {NB_APP}
	      82 {L} {NON_VOLATILE_MEMORY}
	      83 {L} {VOLATILE_MEMORY}
	    90 00
	  00 00

R_AF_NOTIF	AF 80
	  23 04
	    80 00
	    90 00
	  00 00

R_AF_RC	AF 80
	  23 {L} 85 {L} {RC}
	     90 00
	  00 00

Name	Content in hexadecimal string

R_AF_RECEIPT	`AF 80`
	`  23 {L} 86 {L} {RECEIPT}`
	`     90 00`
	`  00 00`

R_CASD	`7F 21 {L}`
	`  7F 21 {L} #CASD_CERTIFICATE 90 00`

R_PROF_PKG_OK	`30 07`
	`  A0 05`
	`    30 03`
	`      80 01 00`

R_SCP03T_EMPTY	`86 00`
R_SCP03T_EXTAUTH_OK	`85 00`

R_SCP03T_INITUP_OK	`84 20`
	`  {KEY_DIV_DATA}`
	`  #SCP03_KVN 03 70`
	`  {CARD_CHALLENGE}`
	`  {CARD_CRYPTOGRAM}`
	`  {SCP03_SEQ_NUM}`

Note 1: Key Information Data Structure – Extended as defined in GlobalPlatform Card Specification [3] may also be returned. The order of the tags 'C0' (i.e. key information data) shall not be checked.

Note 2: In this table, the expanded remote responses using definite length contain a number of executed commands (i.e. value of the BER-TLV tag '80') coded on 2 bytes (i.e. short number) as an example. But, it may be also coded on '01' byte as defined in ETSI TS 102 226 [6]. As a consequence, the expected response scripting template tag (i.e. 'AB') shall be adapted according the eUICC implementation.

Note 3: Depending on the support of the GlobalPlatform Amendment C specification [14] in the Profile linked to the MNO-SD, the lifecycle state may be encoded with two bytes instead of one (that is, the contactless activation state shall be encoded in the second byte). In addition, other tags (e.g. 'C5' – Privileges) may be returned in the R-APDU as the tag '5C' (i.e. tag list) present in the related GET STATUS command may not be supported by the MNO-SD. The content of the tag '9F70' – Lifecycle state is set with '0F' (i.e. SECURED) as an example: it shall not be checked in the response.

Note 4: The values of the tags '8F' (i.e. cumulative granted non-volatile Memory) and '91' (cumulative remaining non-volatile memory) may be also encoded in 2 bytes. In addition, they may be lower or equal to `#MEMORY_QUOTA`.

Note 5: Other keys with an identifier from 1 to 5 may be also present under the keyset identified by `#SCP81_KVN`.

Table 20: Responses

Annex F Bearer Independent Protocol

Here is a sequence explaining the BIP communication between the Device and the eUICC.

Direction	Sequence / Description
	TRIGGERING EVT
eUICC → Device	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* OPEN CHANNEL
Device → eUICC	TERMINAL RESPONSE
eUICC → Device	*PROACTIVE COMMAND PENDING:* SEND DATA
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* SEND DATA containing the data to send to the off-card entity
Device → eUICC	TERMINAL RESPONSE
Several SEND DATA commands may be used to send the complete data	
Device → eUICC	ENVELOPE EVENT DOWNLOAD
eUICC → Device	*PROACTIVE COMMAND PENDING:* RECEIVE DATA
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* RECEIVE DATA
Device → eUICC	TERMINAL RESPONSE containing the data sent by the off-card entity
Several RECEIVE DATA commands may be used to retrieve the complete data	
eUICC → Device	*PROACTIVE COMMAND PENDING:* SEND DATA
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* SEND DATA containing the data to send to the off-card entity
Device → eUICC	TERMINAL RESPONSE
Several SEND DATA commands may be used to send the complete data	
Device → eUICC	ENVELOPE EVENT DOWNLOAD
eUICC → Device	*PROACTIVE COMMAND PENDING:* RECEIVE DATA
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* RECEIVE DATA
Device → eUICC	TERMINAL RESPONSE containing the data sent by the off-card entity
Several RECEIVE DATA commands may be used to retrieve the complete data	

Direction	Sequence / Description
eUICC → Device	*PROACTIVE COMMAND PENDING:* SEND DATA
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* SEND DATA containing the data to send to the off-card entity
Device → eUICC	TERMINAL RESPONSE
Several SEND DATA commands may be used to send the complete data	
Device → eUICC	ENVELOPE EVENT DOWNLOAD
eUICC → Device	*PROACTIVE COMMAND PENDING:* RECEIVE DATA
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* RECEIVE DATA
Device → eUICC	TERMINAL RESPONSE containing the message sent by the off-card entity to close the session
Before closing the channel, the card may send a confirmation	
eUICC → Device	*PROACTIVE COMMAND PENDING:* CLOSE CHANNEL
Device → eUICC	FETCH
eUICC → Device	*PROACTIVE COMMAND:* CLOSE CHANNEL
Device → eUICC	TERMINAL RESPONSE
Note: It is assumed that some proactive commands TIMER MANAGEMENT or MORE TIME may be sent by the eUICC at any time	

Table 21: BIP Exchanges

Annex G CAT_TP PDUs

Here are the different CAT_TP PDUs that shall be used by the CAT_TP entities during a test sequence. The values in square brackets depend on the context and the CAT_TP implementation. The other values need to be checked.

PDU	Value in hexadecimal string

ACK_DATA	40 00 00 12
	{SRC_PORT}
	{DEST_PORT}
	{DATA_LENGTH}
	{SEQ_NUM}
	{ACK_NUM}
	{WIN_SIZE}
	{CS}
	{DATA}
	
	Or
	
	44 00 00 12
	{SRC_PORT}
	{DEST_PORT}
	{DATA_LENGTH}
	{SEQ_NUM}
	{ACK_NUM}
	{WIN_SIZE}
	{CS}
	{DATA}
	
	See Annex C for the definition of {SRC_PORT}, {DEST_PORT}, {SEQ_NUM}, {ACK_NUM}, {WIN_SIZE}, and {CS}.
	
	{DATA} is either a command packet or a response packet as defined in ETSI TS 102 225 [4].
	
	If the data length is higher to the Maximum PDU size, the ACK_DATA shall be segmented (1st byte = '44') and the data shall be split in several PDUs.
	The command packet length shall not be higher than the Maximum SDU size.

ACK_NO_DATA	40 00 00 12
	{SRC_PORT}
	{DEST_PORT}
	00 00
	{SEQ_NUM}
	{ACK_NUM}
	{WIN_SIZE}
	{CS}
	
	See Annex C for the definition of {SRC_PORT}, {DEST_PORT}, {SEQ_NUM}, {ACK_NUM}, {WIN_SIZE}, and {CS}.

ACK_NUL	48 00 00 12

PDU	Value in hexadecimal string

	{SRC_PORT}
	{DEST_PORT}
	00 00
	{SEQ_NUM}
	{ACK_NUM}
	{WIN_SIZE}
	{CS}
	
	See Annex C for the definition of {SRC_PORT}, {DEST_PORT}, {SEQ_NUM}, {ACK_NUM}, {WIN_SIZE}, and {CS}.

RST	10 00 00 13
	{SRC_PORT}
	{DEST_PORT}
	00 00
	{SEQ_NUM}
	{ACK_NUM}
	{WIN_SIZE}
	{CS}
	{REASON_CODE}
	
	See Annex C for the definition of {SRC_PORT}, {DEST_PORT}, {SEQ_NUM}, {ACK_NUM}, {WIN_SIZE}, {CS} and {REASON_CODE}.

SYN	80 00 00 {HL}
	{SRC_PORT}
	#CAT_TP_PORT
	00 00
	{SEQ_NUM}
	00 00
	{WIN_SIZE}
	{CS}
	{MAX_PDU_SIZE}
	{MAX_SDU_SIZE}
	#EID (optional: it may contain another value)
	
	See Annex C for the definition of {HL}, {SRC_PORT}, {SEQ_NUM}, {WIN_SIZE}, {CS}, {MAX_PDU_SIZE} and {MAX_SDU_SIZE}.
	
	{WIN_SIZE} shall be taken into account by the off-card entity.
	
	{MAX_SDU_SIZE} and {MAX_PDU_SIZE} shall be taken into account by the off-card entity.

SYN_ACK	C0 00 00 {HL}
	#CAT_TP_PORT
	{DEST_PORT}
	00 00
	{SEQ_NUM}
	{ACK_NUM}
	{WIN_SIZE}
	{CS}

PDU	Value in hexadecimal string

	{MAX_PDU_SIZE}
	{MAX_SDU_SIZE}
	{IDENTIFICATION_DATA}
	
	See Annex C for the definition of {HL}, {DEST_PORT}, {SEQ_NUM}, {ACK_NUM}, {WIN_SIZE}, {CS}, {MAX_PDU_SIZE} and {MAX_SDU_SIZE}.
	
	{IDENTIFICATION_DATA} is the off-card entity identification data which can be freely chosen.

Table 22: CAT_TP PDUs

Annex H TLS Records

Here are the different TLS records that shall be used by the TLS entities. All values defined in the tables below are hexadecimal strings. The values in square brackets depend on the context and the TLS implementation. The other values need to be checked.

TLS_CLIENT_HELLO		
Content type: Handshake		16
Version: TLS 1.2		03 03
Length		{L}
Protocol message	Message type: ClientHello	01
	Length	{L}
	Version: TLS 1.2	03 03
	Random value	AA BB CC ……01 02
	Session id length	00
	Cipher suite length	{L}
	TLS_PSK_WITH_AES_128_CBC_SHA256	00 AE
	TLS_PSK_WITH_AES_128_GCM_SHA256	00 A8
	Compression length	01
	Compression method: no compression	00
	Extension message length	00 05
	Extension-type: max fragment length	00 01
	Extension data length	00 01
	Max fragment length: 2^9	01
Note 1: TLS_PSK_WITH_AES_128_CBC_SHA256 and/or TLS_PSK_WITH_AES_128_GCM_SHA256 shall be present. Other cipher suites may be present. *Note 2: The TLS record length is coded with 2 bytes.* *Note 3: The protocol message length is coded with 3 bytes.* *Note 4: The cipher suites length is coded with 2 bytes.* *Note 5: The random value present in the table above is informative.*		

TLS_SERVER_HELLO		
Content type: Handshake		16
Version: TLS 1.2		03 03
Length		{L}
Protocol message	Message type: ServerHello	02
	Length	{L}
	Version: TLS 1.2	03 03
	Random value	AA BB CC ……01 02
	Session id length	{L}
	Session id	AA BB CC …
	TLS_PSK_WITH_AES_128_GCM_SHA256	00 A8
	Compression method: no compression	00

	Extension message length	00 05
	Extension-type: max fragment length	00 01
	Extension data length	00 01
	Max fragment length: 2^9	01
Note 1: The cipher suite may be also TLS_PSK_WITH_AES_128_CBC_SHA256. *Note 2: The TLS record length is coded with 2 bytes.* *Note 3: The protocol message length is coded with 3 bytes.* *Note 4: The random value and the session ID present in the table above are informative.*		

TLS_SERVER_HELLO_DONE		
Content type: Handshake		16
Version: TLS 1.2		03 03
Length		00 04
Protocol message	Message type: ServerHelloDone	0E
	Length	00 00 00
Note: this TLS record may be concatenated to the TLS_SERVER_HELLO message		

TLS_1_1_SERVER_HELLO		
Content type: Handshake		16
Version: TLS 1.1		03 02
Length		{L}
Protocol message	Message type: ServerHello	02
	Length	{L}
	Version: TLS 1.1	03 02
	Random value	AA BB CC ……01 02
	Session id length	{L}
	Session id	AA BB CC …
	TLS_PSK_WITH_AES_128_CBC_SHA256	00 AE
	Compression method: no compression	00
	Extension message length	00 05
	Extension-type: max fragment length	00 01
	Extension data length	00 01
	Max fragment length: 2^9	01
Note 1: The TLS record length is coded with 2 bytes. *Note 2: The protocol message length is coded with 3 bytes.* *Note 3: The random value and the session ID present in the table above are informative.*		

TLS_1_1_SERVER_HELLO_DONE		
Content type: Handshake		16
Version: TLS 1.1		03 02
Length		00 04
Protocol message	Message type: ServerHelloDone	0E
	Length	00 00 00
Note: this TLS record may be concatenated to the TLS_1_1_SERVER_HELLO message		

TLS_CLIENT_KEY_EXCHANGE		
Content type: Handshake		16
Version: TLS 1.2		03 03
Length		{L}
Protocol message	Message type: ClientKeyExchange	10
	Length	{L}
	PSK Identity length	{L}
	PSK Identity	#PSK_ID
Note 1: The TLS record length is coded with 2 bytes. *Note 2: The protocol message length is coded with 3 bytes.*		

Note 3: The PSK Identity length is coded with 2 bytes.

TLS_CHANGE_CIPHER_SPEC		
Content type: ChangeCipherSpec		14
Version: TLS 1.2		03 03
Length		00 01
Protocol message	Message type: ChangeCipherSpec	01

TLS_FINISHED		
Content type: Handshake		16
Version: TLS 1.2		03 03
Length		{L}
Protocol message	Message type: Finished	14
	Length	{L}
	Ciphered data	AA BB CC …
Note 1: The TLS record length is coded with 2 bytes. *Note 2: The protocol message length is coded with 3 bytes.* *Note 3: The ciphered data present in the table above is informative.*		

TLS_APPLICATION		
Content type: Application		17
Version: TLS 1.2		03 03
Length		{L}
Protocol message	Ciphered data	AA BB CC …
	MAC	AA BB CC …
	Padding	01
Note 1: The ciphered data contains the HTTP content. *Note 2: The TLS record length is coded with 2 bytes.* *Note 3: The ciphered data, the MAC and the padding present in the table above are informative.*		

TLS_ALERT_CLOSE_NOTIFY		
Content type: Handshake		15
Version: TLS 1.2		03 03
Length		{L}
Protocol message	Alert level : Warning	01
	Alert description: Close notify	00
	MAC	AA BB …
	Padding	01
Note 1: The TLS record length is coded with 2 bytes. *Note 2: The MAC and the padding present in the table above are informative.*		

TLS_ALERT_PROTOCOL_VERSION		
Content type: Handshake		15
Version: TLS 1.2		03 03
Length		{L}
Protocol message	Alert level : Fatal	02
	Alert description: Protocol version	46
	MAC	AA BB …
	Padding	01
Note 1: The TLS record length is coded with 2 bytes. *Note 2: The MAC and the padding present in the table above are informative.*		

Annex I Initial States

Here are all the initial states of the different entities under test. Each initial state is an extract of the GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2]. As consequence, each cross-reference present in the table below (i.e. column Initial state) does not refer to documents listed in the section 1.5 of this Test Plan. The column “Chapter” refers to the section where the initial state is defined in the document GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2].

Chapter	Initial state
2.2.1.1	There shall be only one ISD-R on an eUICC. The ISD-R shall be installed and first personalized by the EUM during eUICC manufacturing. The ISD-R shall be Associated with itself. After eUICC manufacturing, the ISD-R shall be in life-cycle state PERSONALIZED as defined in GlobalPlatform Card Specification [6], section 5.3. The ISD-R privileges shall be granted according to Annex C.
2.2.1.2	There shall be only one ECASD on an eUICC. The ECASD shall be installed and personalized by the EUM during the eUICC manufacturing. The ECASD shall be Associated with the ISD-R. After eUICC manufacturing, the ECASD shall be in life-cycle state PERSONALIZED as defined in GlobalPlatform Card Specification [6], section 5.3. The ECASD shall be personalized by the EUM during eUICC manufacturing with: • PK.CI.ECDSA • SK.ECASD.ECKA • CERT.ECASD.ECKA for eUICC Authentication and key establishment • EUM key set for key renewal • EID
2.2.1.3	At least one ISD-P with a Profile shall be installed and first personalized by the EUM during eUICC manufacturing to allow future eUICC connectivity.
2.2.3	The RID of the Executable Load File, the Executable Module and the Application of the ISD- R and the ECASD shall be set to 'A000000559' (as defined in ISO/IEC 7816-5:2004). The ISD- R Executable Load File AID and the ISD-R Executable Module AID can be freely selected by the EUM. The ISD-R application AID shall be ‘A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 01 00' as defined into Annex H. The ECASD Executable Load File AID and the ECASD Executable Module AID can be freely selected by the EUM.
2.2.5.1	To enable SCP80, the ISD-R shall be personalized before issuance by the EUM with at least one key set, with a Key Version Number between ‘01’ to ‘0F’ following GlobalPlatform Card Specification UICC Configuration [7].
2.2.5.1	To enable SCP81, the ISD-R shall be personalized with at least one key set, with a Key Version Number between ‘40’ to ‘4F’ following GlobalPlatform Secure Element Configuration[34].
2.3	• Every SM-SR and SM-DP shall be certified according to a GSMA agreed certification scheme. • The eUICC shall be certified according to the GSMA eUICC Protection Profile. • The eUICC Manufacturer shall be SAS certified.

Chapter	Initial state
2.3.1	The Certificate Issuer (CI) Role issues the certificates for the eUICC Remote Provisioning System and acts as a trusted third party for the purpose of mutual authentication of the entities of the system. The CI provides: • A self-signed Root Certificate used to verify certificates issued and signed by the CI. • A public key (PK.CI.ECDSA), part of that Root Certificate, used on the eUICC to verify certificates issued by the CI. • A certificate (CERT.DP.ECDSA, signed by the CI) to authenticate the SM-DP. This certificate is used in the “Load and Install Profile” procedure. • A certificate (CERT.SR.ECDSA, signed by the CI) to authenticate the SM-SR. This certificate is used in the “SM-SR change” procedure. • A certificate, signed by the CI, to authenticate the EUM. This certificate is used in the "Download and Install Profile" and in the “SM-SR change” procedures.
2.3.2	The following certificates shall be signed and issued by the CI: • Self-signed Root Certificate • EUM Certificates • SM-SR Certificates • SM-DP Certificates
2.3.2	The following certificates shall be signed and issued by the EUM: • eUICC Certificates
2.3.2	The following certificate and key shall be stored in the eUICC: • the eUICC Certificate • the Root public key
2.3.2	The eUICC Certificate is part of the EIS (eUICC Information Set) which is stored in the SM-SR and/or at EUM level. This certificate contains: • the PK.ECASD.ECKA used for ElGamal Elliptic Curves key agreement as defined in GlobalPlatform Card Specification Amendment E [11] • the EID • the technical reference of the product, which allows the Common Criteria (CC) certification report to be identified by Common Criteria certification body
Annex B	In case Web Services is used, the section "Binding to SOA environment" is normative and implementation shall comply with the requirements provided in this section.
Annex B / 2	This specification mandates usage of SOAP v1.2 as the minimal version and specified in [40].
Annex B / 2.1.2	WS-MakeConnection shall be used in asynchronous scenarios when the receiving party of a request cannot initiate a connection to the sending party (due to network security constraints for example).

Chapter	Initial state
Annex B / 2.2	To secure the messages being sent between Function requester and Function provider, one of the two following mechanisms shall be used: 1. Relying on mutual authenticated transport level security (Transport Layer Security, TLS) 2. Relying on transport level security (TLS) with only server side authentication and WS- Security standards This specification mandates usage of TLS v 1.2 defined in RFC 5246 [15] to allow appropriate algorithm and key length as defined in section 2.4.1
Annex B / 4	In case Web Services are used, the following WSDL files (provided within the SGP.02 WSDL package) shall be used: • ES1_SMSR.wsdl • ES2_MNO.wsdl • ES2_SMDP.wsdl • ES3_SMDP.wsdl • ES3_SMSR.wsdl • ES4_MNO.wsdl • ES4_SMSR.wsdl • ES7_SMSR_Provider.wsdl • ES7_SMSR_Requester.wsdl

Table 23: Initial States

Annex J Requirements

Each requirement in the tables below is an extract of either the GSMA Embedded SIM Remote Provisioning Architecture [1] or the GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2].

J.1 Format of the Requirements Table

The columns in Table 21 and 22 have the following meaning:

Column	Meaning
ID	Requirement identifier used in the test cases defined in this Test Plan. This identifier is unique and formatted as follow "XXX_REQYYY" with • XXX: a prefix related to the corresponding functional group • YYY: a number
Source	The cross-reference to the source document where the requirement is specified. All cross-references are described in the section 1.5 of this Test Plan.
Chapter	The chapter in the source document where the requirement is specified.
Support	The following common notations are used for the support column: M mandatory: shall be supported by the implementation C conditional: the support of the requirement depends of the support of other requirement(s) O optional: may be supported or not by the implementation
Description	An extract of the source document that describes the requirement. Some of these descriptions are adapted for readability reason. All cross-references present in this column do not refer to the ones present in this document (i.e. section 1.5) but refer to cross-references defined in the corresponding source document. The notes in *italic and underline* shall be considered as remarks or comments related to the requirement.
Functional group	Functional group of the corresponding requirement. A functional group may be: • Platform Management • eUICC Management • Profile Management • Procedure Flow • Security

Table 24 Format of the Tables of Requirements

J.2 Requirements in Scope

Here are all the requirements' descriptions that are covered by this Test Plan.

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ1	[2]	2.2.1.1	M	The LOCKED state shall not be supported by the ISD-R.	eUICC Management
PF_REQ1	[2]	2.2.1.1	M	The ISD-R shall only be able to perform Platform Management functions on ISD-Ps.	Platform Management
PM_REQ1	[2]	2.2.1.3	M	No component outside the ISD-P shall have visibility or access to any Profile Component with the exception of the ISD-R, which shall have read access to POL1	Profile Management
PM_REQ2	[2]	2.2.1.3	M	A Profile Component shall not have any visibility of, or access to, components outside its ISD-P. An ISD-P shall not have any visibility of, or access to, any other ISD-P.	Profile Management
EUICC_REQ2	[2]	2.2.1.3	M	It shall be possible to allocate the same AID within different Profiles. A Profile Component shall not use the reserved ISD-R, ISD-P and ECASD AIDs.	eUICC Management
EUICC_REQ3	[2]	2.2.1.3	M	It shall be possible to allocate the same TAR within distinct Profiles. A Profile Component shall not use the reserved ISD-R, ISD-P and ECASD TARs.	eUICC Management
EUICC_REQ4	[2]	2.2.1.3	M	After execution of the procedure described in section 3.1.1 (ISD-P creation), the ISD-P shall be in SELECTABLE state	eUICC Management
EUICC_REQ5	[2]	2.2.1.3	M	After execution of the procedure described in section 3.1.2 (Key Establishment with Scenario#3-Mutual Authentication), the ISD-P shall be in PERSONALIZED state	eUICC Management
PM_REQ3	[2]	2.2.1.3	M	After execution of the procedure described in section 3.1.3 (Download and Installation of the Profile) or 3.4 (Profile Disabling), the ISD-P shall be in the DISABLED state. The ISD-P can also transition to the DISABLED state as the result of the enabling of another ISD-P as described in section 3.2, or the activation of the fall-back mechanism.	Profile Management
PM_REQ4	[2]	2.2.1.3	M	After execution of the procedure described in section 3.2 (Profile Enabling), the ISD-P shall be in the ENABLED state. The ISD-P can also transition to the ENABLED state as the result of the activation of the fall-back mechanism.	Profile Management
EUICC_REQ6	[2]	2.2.1.3	M	The LOCKED state shall not be supported by an ISD-P.	eUICC Management
EUICC_REQ7	[2]	2.2.1.3	M	When an ISD-P is not in Enabled state, the eUICC shall ensure that Remote management of any Profile Component is not possible via the ES6 interface	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ8	[2]	2.2.1.3	M	When an ISD-P is not in Enabled state, the eUICC shall ensure that the file system within the Profile cannot be selected by the Device or any application on the eUICC	eUICC Management
EUICC_REQ9	[2]	2.2.1.3	M	When an ISD-P is not in Enabled state, the eUICC shall ensure that the applications (including NAAs and Security Domains) within the Profile cannot be selected, triggered or deleted.	eUICC Management
EUICC_REQ10	[2]	2.2.2	M	The EID shall be stored within the ECASD and can be retrieved by the Device at any time using the standard GlobalPlatform GET DATA command by targeting the ECASD as specified in GlobalPlatform Card Specification [6] as follows: > Select the ECASD using the SELECT command with the AID value defined in section 2.2.3, > Send a ‘GET DATA’ command to the ECASD with the data object tag '5A' to get the EID. The EID shall have the format described in section 2.2.2.	eUICC Management
EUICC_REQ11	[2]	2.2.3	M	The ECASD application AID shall be ‘A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 02 00' as defined into Annex H.	eUICC Management
EUICC_REQ12	[2]	2.2.3	M	The ISD-P application shall be installed by SM-SR during the “Profile Download and Installation” procedure. The ISD-P Executable Load File AID shall be ‘A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 0D 00' as defined into Annex H. The ISD-P Executable Module AID shall be ‘A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 0E 00' as defined into Annex H. The ISD-P application AID shall be coded according to Annex 8. The SM-SR shall allocate the ISD-P application AID in the range defined in Annex H.	eUICC Management
PM_REQ5	[2]	2.2.3	M	The MNO-SD application AID and TAR(s) can be freely allocated by the MNO during Profile definition.	Profile Management
EUICC_REQ13	[2]	2.2.5.1	M	The eUICC shall support SCP80 (defined in ETSI 102 225 [4] and ETSI 102 226 [5]).	eUICC Management
EUICC_REQ14	[2]	2.2.5.1	C	The eUICC may support SCP81 (as defined in ETSI TS 102 226) *Note: If EUICC_REQ18 is not supported, this requirement shall be supported*	eUICC Management
EUICC_REQ15	[2]	2.2.5.2	M	To enable SCP03 and SCP03t, the ISD-P shall be personalized with at least one key set, with a Key Version number between ‘30’ to ‘3F’ (see GlobalPlatform Secure Element Configuration [34]).	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ16	[2]	2.3	M	For the eUICC interfaces, the Platform Management commands (ES5) and the OTA Platform commands (ES6) shall be protected by either a SCP80 or SCP81 secure channel with security level defined in section 2.4.	eUICC Management
EUICC_REQ17	[2]	2.3	M	The Profile Management commands (ES8) shall be at least protected by a SCP03 security level as detailed in section 2.5.	eUICC Management
EUICC_REQ18	[2]	2.4.1	C	The eUICC may support CAT_TP *Note: If EUICC_REQ14 is not supported, this requirement shall be supported*	eUICC Management
PF_REQ2	[2]	2.4.1	M	The SM-SR shall support SMS, HTTPS and CAT_TP.	Platform Management
EUICC_REQ19	[2]	2.4.3	M	The eUICC shall support the sending of secure packet over SMS as defined in 3GPP TS 31.115 [13]. The eUICC shall support RAM over SMS as defined in ETSI TS 102 226 [5]. The eUICC shall comply with 3GPP TS 31.111 [27] and 3GPP TS 31.116 [28]. Except for the notification described in section 3.15.1, concerning the security level, the SMS (MT or MO) shall make use of a CC with a length of 64 bits using AES CMAC mode, ciphering using AES in CBC mode and counter value higher (SPI1='16')..	eUICC Management
EUICC_REQ20	[2]	2.4.3	M	Procedures for the PoR shall follow ETSI TS 102 225 **Error! Reference source not found.** and 3GPP TS 31.115 **Error! Reference source not found.** with the following precisions: • In the case that an incoming SMS for the ISD-R does not meet the security level described in "EUICC_REQ19", it must be rejected by the eUICC and no PoR shall be sent back • When the eUICC cannot authenticate the SM-SR, it shall not send any PoR and discard the command packet with no further action being taken	eUICC Management
EUICC_REQ54	[2]	2.4.3	M	SPI2 shall be set to: • '00': no PoR (this value shall only be used for the notification described in section 3.15.1), • or to '39': PoR with CC and encryption.	eUICC Management
EUICC_REQ21	[2]	2.4.3	M	When a PoR is returned, the SMS shall make use of a CC with a length of 64 bits using AES CMAC mode , ciphering using AES in CBC mode and shall be sent using SMS-SUBMIT mode.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ22	[2]	2.4.3.3	M	The commands sent to the eUICC within a secure script in SMS shall be formatted as an expanded remote command structure as defined in ETSI TS 102 226 [5]. As a consequence, the eUICC shall provide the answer as an expanded remote response structure.	eUICC Management
EUICC_REQ23	[2]	2.5	M	The eUICC shall support the Secure Channel Protocol 03 (SCP03) as defined in GlobalPlatform Card Specification Amendment D [10], as well as the variant SCP03t defined in this specification (see section **Error! Reference source not found.**), with: • AES in CBC mode with key length of 128 bits, referred as AES-128 • Use of C-MAC, C-DECRYPTION R-MAC and R-ENCRYPTION for SCP03 (set in reference control parameter P1 of the EXTERNAL AUTHENTICATE command) and for SCP03t • Use of mode i='70', meaning use of pseudo-random card challenge, R-MAC and R-ENCRYPTION support As a result the SM-DP and its ISD-P are mutually authenticated, all commands sent from the SM-DP to the ISD-P are signed and encrypted, and all responses sent by the ISD-P to the SM-DP are also signed and encrypted.	eUICC Management
PROC_REQ1	[2]	3.1.1	M	The ISD-P creation process must be compliant with the Figure 10 and with the procedure described in this section.	Procedure Flow
PROC_REQ2	[2]	3.1.2	M	The Key Establishment with Scenario#3-Mutual Authentication process must be compliant with the Figure 11 and with the procedure described in this section.	Procedure Flow
PROC_REQ3	[2]	3.1.3	M	The Download and Installation of the Profile process must be compliant with the Figure 12 and with the procedure described in this section.	Procedure Flow
PROC_REQ4	[2]	3.1.4	M	The Error Management Sub-Routine described in Figure 13 must be called when an error occurs during the key-establishment or the Profile Download and Installation procedures. This process shall be compliant with the procedure described in this section.	Procedure Flow
PROC_REQ5	[2]	3.2.1	M	The profile enabling process must be compliant with the Figure 14 and with the procedure described in this section.	Procedure Flow
PROC_REQ6	[2]	3.2.2	M	The Connectivity failure case described in Figure 15 must be called when an error occurs during the profile enabling procedure. This process shall be compliant with the procedure described in this section.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
PROC_REQ7	[2]	3.3.1	M	The Profile Enabling via SM-DP must be compliant with the Figure 16 and with the procedure described in this section.	Procedure Flow
PROC_REQ8	[2]	3.3.2	M	The connectivity failure case described in Figure 17 must be called when an error occurs during the profile enabling via SM-DP procedure. This process shall be compliant with the procedure described in this section.	Procedure Flow
PROC_REQ9	[2]	3.4	M	The Profile Disabling process must be compliant with the Figure 18 and with the procedure described in this section.	Procedure Flow
PROC_REQ10	[2]	3.5	M	The Profile Disabling via SM-DP process must be compliant with the Figure 19 and with the procedure described in this section.	Procedure Flow
PROC_REQ11	[2]	3.6	M	The Profile and ISD-P deletion process must be compliant with the Figure 20 and with the procedure described in this section.	Procedure Flow
PROC_REQ12	[2]	3.7	M	The Profile and ISD-P Deletion via SM-DP must be compliant with the Figure 21 and with the procedure described in this section.	Procedure Flow
PROC_REQ13	[2]	3.8	M	The SM-SR Change process must be compliant with the Figure 22 and with the procedure described in this section.	Procedure Flow
PROC_REQ14	[2]	3.9	M	The eUICC registration process must be compliant with the Figure 23 and with the procedure described in this section.	Procedure Flow
PROC_REQ16	[2]	3.11	M	The POL2 Update via SM-DP process must be compliant with the Figure 25 and with the procedure described in this section.	Procedure Flow
PROC_REQ17	[2]	3.12	M	The POL1Update by MNO process must be compliant with the Figure 26 and with the procedure described in this section.	Procedure Flow
PROC_REQ18	[2]	3.13	M	The Connectivity Parameters Update by MNO must be compliant with the Figure 27 and with the procedure described in this section.	Procedure Flow
PROC_REQ19	[2]	3.14	M	The Connectivity Parameters Update using SCP03 must be compliant with the Figure 28 and with the procedure described in this section.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
PROC_REQ20	[2]	3.15.1	M	The Default Notification Procedure using SMS must be compliant with the Figure 29 and with the procedure described in this section.	Procedure Flow
PROC_REQ21	[2]	3.15.2	M	The Default Notification Procedure using HTTPS must be compliant with the Figure 30 and with the procedure described in this section.	Procedure Flow
PROC_REQ22	[2]	3.16	M	The Fall-back Activation Procedure must be compliant with the Figure 31 and with the procedure described in this section.	Procedure Flow
PF_REQ3	[2]	4.1.1.1	M	ES5: CreateISDP Description: This function creates an ISD-P on the eUICC. Parameters: • ISD-P-AID • Memory quota for the ISD-P (optional) Command Description: INSTALL COMMAND The command is an Install command as defined in GlobalPlatform Card Specification [6] and must be compliant with the Tables defined in section 4.1.1.1. Privileges granted to the ISD-P, as specified in Annex C, shall be at least: • Security Domain • Trusted Path • Authorized Management Data Field Returned in the Response Message: A single byte of '00' shall be returned indicating that no additional data is present, as defined in the GlobalPlatform Card Specification [6].	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ4	[2]	4.1.1.2	M	ES5: EnableProfile Description: This function is used to enable a Profile on the eUICC. The function makes the target Profile Enabled, and disables implicitly the currently Enabled Profile. Parameters: • ISD-P-AID Function flow Upon reception of the Profile Enabling command, the eUICC shall: • Verify that the target Profile is in the Disabled state • Verify that POL1 of the currently Enabled Profile allows its disabling • If any of these verifications fail, terminate the command with an error status word • Disable the currently Enabled Profile and Enable the target Profile • Send the REFRESH command in “UICC Reset” mode to the Device according to ETSI TS 102 223 [3] • Send notification Command Description: STORE DATA COMMAND This command is a STORE DATA command, as described in GlobalPlatform Card Specification [6] and in Tables defined in section 4.1.1.2. Data Field Returned in the Response Message: The data field of the response message shall not be present. Specific Processing State returned in response Message: ’69 85’: Profile is not in the Disabled state. ’69 E1’: POL1 of the currently Enabled Profile prevents this action.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ5	[2]	4.1.1.3	M	ES5: DisableProfile Description: This function is used to disable a Profile on the eUICC. This function makes the target Profile Disabled, and implicitly enables the Profile which has the Fall-back Attribute set. Parameters: • ISD-P-AID of the currently Enabled Profile Function flow Upon reception of the Profile Disabling command, the eUICC shall: • Verify that the target Profile is in Enabled state • Verify that POL1 of the currently Enabled Profile allows its disabling • Verify that the target Profile is not the Profile with Fall-back Attribute set • If any of these verifications fail, terminate the command with an error status word • Disable the target Profile and enable the Profile with the Fall-back Attribute set • Send the REFRESH command in “UICC Reset” mode to the Device according to ETSI TS 102 223 [3]. Command Description: STORE DATA COMMAND This command is a STORE DATA command, as described in GlobalPlatform Card Specification [6] and in Tables defined in section 4.1.1.3. Data Field Returned in the Response Message: The data field of the response message shall not be present. Specific Processing State returned in response Message: ’69 85’: Profile is not in the Enabled state or Profile has the Fall-back Attribute. ’69 E1’: POL1 of the Profile prevents disabling.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ6	[2]	4.1.1.4	M	ES5: DeleteProfile Description: This function is used to delete a Profile from the eUICC. This function deletes the ISD-P and its associated Profile. Parameters: • ISD-P-AID Function flow Upon reception of the DELETE command, the eUICC shall: • Verify that POL1 of the target Profile allows its deletion • Verify that the target Profile is not the Profile with Fall-back Attribute set • Verify that the target Profile is not in the Enabled state • If any of these verifications fail, terminate the command with an error status word • Delete the ISD-P with its Profile Command Description: DELETE COMMAND This function is realized through the GlobalPlatform DELETE command as defined in GlobalPlatform Card Specification Amendment C [9] and in Tables defined in section 4.1.1.4. Data Field Returned in the Response Message: A single byte of '00' shall be returned indicating that no additional data is present. Specific Processing State returned in response Message: ’69 85’: Profile is in Enabled state or Profile has the Fall-back Attribute. ’69 E1’: POL1 of the Profile prevents deletion.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ7	[2]	4.1.1.5	M	ES5: eUICCCapabilityAudit Description: This function is used to query the status of the eUICC. Parameters: It may be used to ensure the data within the SM-SR's EIS database is up to date. This function uses two commands which shall be implemented as an extension of the GlobalPlatform functions GET DATA and GET STATUS. Commands Description: GET DATA The GET DATA command is coded according to the Tables defined in section 4.1.1.5. This function can return: • Number of installed ISD-P and available not allocated memory • ECASD Certificate Data Field Returned in the Response Message: The coding of the response message is defined in Tables defined in section 4.1.1.5. GET STATUS The GET STATUS command is coded according to Tables defined in section 4.1.1.5. This function can return: • Each ISD-P-AID • State of the ISD-Ps / Profiles Data Field Returned in the Response Message: The coding of the response message is defined in Tables defined in section 4.1.1.5.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ8	[2]	4.1.1.6	M	ES5: MasterDelete Description: This function deletes a target Profile on the target eUICC regardless of POL1 Rules. This function shall use the ISD-P token verification key in order to authenticate the source of the command. Parameter: • ISD-P-AID • Delete Token as defined by GlobalPlatform Card Specification [6] , provided by the SM-DP Function flow Upon reception of the Master Delete command, the eUICC shall: • Verify that the target Profile is in the Disabled state • Verify that the target Profile is not the Profile with Fall-back Attribute set • Verify the Token (actually performed by the ISD-P) • If any of these verifications fail, terminate the command with an error status word • Delete the ISD-P with its Profile, regardless of POL1 Command Description: This function is realized through the GlobalPlatform DELETE command as defined in GlobalPlatform Card Specification Amendment C [9] and in Tables defined in section 4.1.1.6. Data Field Returned in the Response Message: A single byte of ‘00’ shall be returned indicating that no additional data is present. Specific Processing State returned in response Message: ’69 85’: Profile is not in the Disabled state or Profile has the Fall-back Attribute.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ9	[2]	4.1.1.7	M	ES5: SetFallbackAttribute Description: This function sets the Fall-back Attribute for one Profile on the target eUICC. Parameters: • ISD-P-AID Function flow Upon reception of the STORE DATA command, the eUICC shall: • Set the Fall-back Attribute for the target Profile • Remove the Fall-back Attribute from the Profile that has the attribute currently assigned Setting of the Fall-back Attribute is done via ISD-R. Command Description: STORE DATA Command This function is realized through the GlobalPlatform STORE DATA command as defined in GlobalPlatform Card Specification [6] and in Tables defined in section 4.1.1.7. Data Field Returned in the Response Message: The data field of the response message shall not be present.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ24	[2]	4.1.1.8	M	ES5: establishISDRKeySet Description: This function is used to perform mutual authentication between the new SM- SR and the eUICC and to establish a shared secret key set between the new SM-SR and the ISD-R. Parameters: • Ephemeral public key of the new SM-SR • Certificate for the new SM-SR Command Description: This function is realized through GlobalPlatform STORE DATA commands as defined in GlobalPlatform Card Specification [6]. First STORE DATA command Command Message The STORE DATA command message shall be coded according to Tables defined in section 4.1.1.8. Data Field Returned in the Response Message: The STORE DATA response shall contain the data described in Tables defined in section 4.1.1.8. Second STORE DATA command Command Message The STORE DATA command message shall be coded according to Tables defined in section 4.1.1.8. Data Field Returned in the Response Message: The STORE DATA response shall contain the data described in Tables defined in section 4.1.1.8.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ25	[2]	4.1.1.9	M	ES5: FinaliseISDRhandover Description: This function deletes all keys in the ISD-R except for the key ranges indicated by the command parameter(s). It is intended as a simple clean-up mechanism for the new SM-SR after takeover to get RID of all keys of the previous SM-SR in the ISD-R. Parameters: • Key Ranges of keys not to be deleted. Command Description: DELETE COMMAND This function is realized through a GlobalPlatform DELETE command as defined in GlobalPlatform Card Specification [6] with proprietary parameters (see Tables defined in section 4.1.1.9). Function flow Upon reception of the DELETE command, the eUICC shall: • Check that all keys of the key set(s) used for setting up the current secure channel are among the keys not to be deleted. For SCP81, this also includes the key set used for the push SM. If that check fails, the command is terminated without deleting any key. • Delete all keys except those in the key ranges indicated in the command parameters. Data Field Returned in the Response Message: The data field of the response message shall contain a single byte of ‘00’. Specific Processing State returned in response Message: ’69 85’: Key(s) of key set used for the current secure channel is/are among the keys to be deleted.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ26	[2]	4.1.1.10	M	ES5: UpdateSMSRAddressingParameters Description: This function is used to update SM-SR addressing Parameters on the eUICC. Parameters: • ISD-R AID • SM-SR addressing Parameters Function flow Upon reception of the SM-SR addressing Parameters update command, the eUICC shall: Update the SM-SR addressing Parameters of the targeted ISD-R. Commands This command is a STORE DATA command, as described in GlobalPlatform Card Specification [6] and in Tables defined in section 4.1.1.10. Data Field Returned in the Response Message: The data field of the response message shall not be present.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ27	[2]	4.1.1.11	M	ES5: HandleDefaultNotification Description: This function provides a default notification from the eUICC to the SM-SR. Parameters: • EID • ISD-P AID • Mobile Equipment Identification (e.g. MEID, IMEI) • Notification Sequence number • Notification type The eUICC notification is composed of a single BER-TLV tag including several COMPREHENSION-TLV data objects; the COMPREHENSION-TLV format is defined in ETSI TS 102 223 [3]. See Tables defined in section 4.1.1.11. Secured data structure for eUICC notification over SMS The data shall be sent using definite length coding, and shall contain one Command TLV encapsulated in the Command Scripting Template.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ29	[2]	4.1.1.12	M	ES5: HandleNotificationConfirmation Description: This function confirms the notification and triggers potential follow-up activities required by POL1. Parameters: • Notification Sequence number Function flow Upon reception of the STORE DATA command, the eUICC shall: • Disable the retry mechanism for the notification • Perform the follow-up activities required by POL1 upon the activity that triggered the original notification • Return the result of any such activity in the response data Command Description: This function is realized through the GlobalPlatform STORE DATA command as defined in GlobalPlatform Card Specification [6] and in Tables defined in section 4.1.1.12. Data Field Returned in the Response Message: The data field of the response message shall either • not be present, if no follow-up activities had to be performed, or • contain the data structure defined in section 4.1.1.12 if follow-up activities were performed	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ6	[2]	4.1.2.1	M	ES6: UpdatePOL1byMNO Description: This function is used to update POL1 on the eUICC. Parameters: • POL1 Function flow Upon reception of the POL1 update command, the eUICC shall: • Update POL1 of the ISD-P containing the targeted MNO-SD. Commands This function consists of an INSTALL [for personalization] command followed by a STORE DATA command, as described in GlobalPlatform Card Specification [6] and in Tables defined in section 4.1.2.1. Data Field Returned in the Response Message: A single byte of '00' shall be returned indicating that no additional data is present, as defined in the GlobalPlatform Card Specification [6].	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ7	[2]	4.1.2.2	M	ES6: UpdateConnectivityParametersByMNO Description: This function is used to update Connectivity Parameters on the eUICC. Parameters: • Connectivity Parameters Function flow Upon reception of the Connectivity Parameters update command, the eUICC shall: • Update the Connectivity Parameters of the ISD-P containing the targeted MNO-SD. Commands This function consists of an INSTALL [for personalization] command followed by a STORE DATA command, as described in GlobalPlatform Card Specification [6]. According to GlobalPlatform Card Specification [6], INSTALL [for personalization] command can only be used on applications Associated with a Security Domain. As an exception from this rule, the eUICC shall allow the MNO-SD to receive this command sequence with data destined to the ISD-P. INSTALL [for personalization] command: As defined in section 4.1.2.1. STORE DATA command: As defined in section 4.1.3.4.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ8	[2]	4.1.3.1	M	ES8: EstablishISDPKeySet Description: This function is used to perform mutual authentication between the SM-DP and the eUICC and to establish a shared secret key set between the SM-DP and the ISD-P. Parameters: • ISD-P AID • Ephemeral public key of the SM-DP • Certificate for the SM-DP Command Description: This function is realized through GlobalPlatform INSTALL [for personalization] and STORE DATA commands as defined in GlobalPlatform Card Specification [6]. INSTALL [for personalization] command: see Tables defined in section 4.1.3.1. Data Field Returned in the Response Message: A single byte of '00' shall be returned indicating that no additional data is present as defined in the GlobalPlatform **Error! Reference source not found.**. First STORE DATA command The STORE DATA command message shall be coded according to Tables defined in section 4.1.3.1. Data Field Returned in the Response Message: The STORE DATA response shall contain the data described in Tables defined in section 4.1.3.1. Second STORE DATA command The STORE DATA command message shall be coded according to Tables defined in section 4.1.3.1. Data Field Returned in the Response Message: The STORE DATA response shall contain the data described in Tables defined section 4.1.3.1.	Profile Management
EUICC_REQ30	[2]	4.1.3.2	M	All ES8 functions in subsequent sections require securing the commands by SCP03. *(Replaced by the EUICC_REQ17)*	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ9	[2]	4.1.3.3	M	ES8: DownloadAndInstallation Description: This function is used to load a Profile into an ISD-P on the eUICC. The ISD-P must be already created and also already personalized. The Profile created by the SM-DP must be compatible with the targeted eUICC. The Profile shall include in particular: • the setting of POL1, if defined by MNO • the setting of Connectivity Parameters (see section 4.1.3.4) • the setting of ISD-P state from 'CREATED' to 'DISABLED' when installation is finished Parameters: • Profile	Profile Management
EUICC_REQ57	[2]	4.1.3.3	M	During the downloading process, the Profile shall be protected by SCP03t. Description of SCP03t: This is a secure channel protocol based on GlobalPlatform's SCP03 usable for TLV structures. The data transported in the command TLVs shall consist of the Profile Package specified in the SIMalliance eUICC Profile Package - Interoperable Format Technical Specification [53]; the response TLVs shall transport PE responses as provided by the Profile Package processing specified in [53]. The Profile Package consists of a sequence of Profile Element (PE) TLVs. As the security mechanisms are exactly the same as SCP03, the SCP03 key sets are used for SCP03t.	eUICC Management
EUICC_REQ58	[2]	4.1.3.3	M	SCP03t does not take that PE structure into account, but treats the whole Profile Package as one block of transparent data. That block of data is split into segments of a maximum size of 1024 bytes (including the tag and length field). The eUICC shall support profile command data segments of at least up to this size.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ59	[2]	4.1.3.3	M	SCP03t initiation uses a TLV equivalent to the INITIALIZE UPDATE APDU. Secure Channel Initiation: INITIALIZE UPDATE command TLV: The data used in the command and response TLVs are described in the section 4.1.3.3 and shall be encapsulated with the tag '84'. In case of an error, tag '9F84' is used. The following values are defined: • '01': error in length or structure of command data • '03': referenced data not found	eUICC Management
EUICC_REQ60	[2]	4.1.3.3	M	SCP03t initiation uses a TLV equivalent to the EXTERNAL AUTHENTICATE APDU. Secure Channel Initiation: EXTERNAL AUTHENTICATE command TLV: The data used in the command and response TLVs are described in the section 4.1.3.3 and shall be encapsulated with the tag '85'. The security level shall be set to '33': "C DECRYPTION, R ENCRYPTION, C MAC, and R MAC". If the message is accepted, a TLV with tag '85' and length zero shall be returned. In case of an error, tag '9F85' is used. The following values are defined: • '01': error in length or structure of command data • '02': security error	eUICC Management
EUICC_REQ61	[2]	4.1.3.3	M	SCP03t Command TLV C-MAC and C-DECRYPTION Generation and Verification: For encapsulating encrypted profile command data in a SCP03t TLV, tag '86' is used. SCP03t Response R-MAC and R-ENCRYPTION Generation and Verification: For encapsulating encrypted profile response data in a SCP03t TLV, tag '86' is used. In case of an error, tag '9F86' is used. The following values are defined: • '01': error in length or structure of command data • '02': security error	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ31	[2]	4.1.3.4	M	ES8: UpdateConnectivityParameters SCP03 Description: This function is used to update Connectivity Parameters on the eUICC. This function has the following parameter: • ISD-P AID • Connectivity Parameters Function flow Upon reception of the Connectivity Parameters update command, the eUICC shall: • Update the Connectivity Parameters of the targeted ISD-P Commands STORE DATA Command This command is a STORE DATA command, as described in GlobalPlatform Card Specification [6] section 11.11.3.2 and in Tables described in section 4.1.3.4. Data Field Returned in the Response Message: The data field of the response message shall not be present.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ32	[2]	5.2.1	M	ES1: RegisterEIS Description: This function allows an eUICC Manufacturer (EUM) to register an eUICC represented by its eUICC Information Set (EIS) within an identified SM-SR information database. The EIS contains the complete set of data that is applicable for the SM-SR to manage the lifecycle of this eUICC. This data set is split in two different parts: • A fixed signed part containing the identification of the eUICC • A variable part containing the keys for the Platform Management plus the list of the different Profile loaded with the identified eUICC This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the registration function has been successfully executed on the SM-SR as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.2.1.	eUICC Management
PM_REQ10	[2]	5.3.1	M	ES2: GetEIS Description: This function allows the MNO to retrieve up to date the EIS information. The SM-DP shall forward the function request to the SM-SR “ES3.GetEIS” as defined in section 5.4.1. Input/Output data described in Tables present in section 5.3.1.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ11	[2]	5.3.2	M	ES2: DownloadProfile Description: This function allows the MNO to request that the SM-DP downloads a Profile, identified by its ICCID, via the SM-SR identified by the MNO on the target eUICC, the eUICC being identified by its EID. Function flow Upon reception of the function request, the SM-DP shall perform the following minimum set of verifications: • The SM-DP shall verify it is responsible for downloading and installation of the Profile SM-DP may provide additional verifications In case one of these conditions is not satisfied, the SM-DP shall refuse the function request and return a ‘Function execution status’ indicating ‘Failed’ with the relevant status code (see table below). The SM-DP shall perform/execute the function according to the Profile Download and Installation procedure described in section 3.1. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the function has been successfully executed by the function provider as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.3.2.	Profile Management
PM_REQ12	[2]	5.3.3	M	ES2: UpdatePolicyRules Description: This function allows the MNO to update POL2 of a Profile, identified by its ICCID, and installed on an eUICC identified by its EID. The SM-DP shall forward this function request to the identified SM-SR by calling the ES3.UpdatePolicyRules function as defined in section 5.4.6. Input/Ouput data described in Tables present in section 5.3.3.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ13	[2]	5.3.4	M	ES2: UpdateSubscriptionAddress Description: This function enables the caller to update the Subscription Address for a Profile in the eUICC Information Set (EIS) of a particular eUICC identified by the EID and ICCID. The Subscription Address is the identifier, such as MSISDN and/or IMSI, through which the eUICC is accessible from the SM-SR via the mobile network when the Profile is in Enabled state. The function replaces the content of the Subscription Address. The SM- DP shall forward the function request to the SM-SR “ES3.UpdateSubscriptionAddress” as defined in section 5.4.7. Input/Output data described in Tables present in section 5.3.4.	Profile Management
PF_REQ12	[2]	5.3.5	M	ES2: EnableProfile Description: This function allows the MNO owner of the Profile to request a SM-DP to enable a Profile in a specified eUICC, eUICC being identified by its EID. The SM-DP receiving this request shall process it according to the “Profile Enabling via SM- DP” procedure described in the section 3.3 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been Enabled on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.3.5.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ13	[2]	5.3.6	M	ES2: DisableProfile Description: This function allows the MNO to request a Profile Disabling to the SM-DP in charge of the management of the targeted eUICC; eUICC being identified by its EID. The target Profile is owned by the requesting MNO. The SM-DP receiving this request shall process it according to Profile Disabling via SM-DP procedure described in section 3.5 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been Disabled on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.3.6.	Platform Management
PF_REQ14	[2]	5.3.7	M	ES2: DeleteProfile Description: This function allows the MNO to request deletion of the target ISD-P with the Profile to the SM-DP; eUICC being identified by its EID. The SM-DP shall forward the function request to the SM-SR “ES3.DeleteISDP” as defined in section 5.4.10. Input/Output data described in Tables present in section 5.3.7.	Platform Management
PF_REQ15	[2]	5.3.8	M	ES2: HandleProfileDisabledNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been Disabled on the eUICC identified by its EID. It is assumed that the ICCID is enough for the SM-DP to retrieve the MNO to notify. This notification also conveys the date and time specifying when the operation has done. Input data described in Tables present in section 5.3.8.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ16	[2]	5.3.9	M	ES2: HandleProfileEnabledNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been Enabled on the eUICC identified by its EID. It is assumed that the ICCID is sufficient for the SM-DP to retrieve the MNO to notify. This notification also conveys the date and time specifying when the operation has been done. Input data described in Table present in section 5.3.9.	Platform Management
EUICC_REQ33	[2]	5.3.10	M	ES2: HandleSMSRChangeNotification Description: This function shall be called for notifying each MNO owning a Profile hosted in the eUICC, identified by its EID that the SM-SR has changed. The notification is sent by the new SM-SR to the SM-DP, which route this notification to the MNO. This notification also conveys the date and time specifying when the operation has been done. Input data described in Tables present in section 5.3.10.	eUICC Management
PF_REQ17	[2]	5.3.11	M	ES2: HandleProfileDeletedNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been deleted on the eUICC identified by its EID. This notification also conveys the date and time specifying when the operation has been done. Input data described in Tables present in section 5.3.11.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ14	[2]	5.4.1	M	ES3: GetEIS Description: This function allows retrieving the eUICC Information Set (EIS) of a particular eUICC from the SM-SR information database based on the EID. The retrieved EIS contains only the data that is applicable for that particular SM-DP. The SM-DP utilises the retrieved EIS, for instance, to verify the eligibility of the eUICC (e.g. type, certificate and memory). This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the download function has been successfully executed on the SM-SR as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.1.	Profile Management
PM_REQ15	[2]	5.4.2	M	ES3: AuditEIS Description: This function allows the SM-DP to retrieve up to date the EIS information. The SM-SR shall use the relevant functions of the ES5 interface to retrieve the information from the eUICC. At the end of the successful execution of this function, the SM-SR shall update its EIS database upon the basis of this information. Input/Output data described in Tables present in section 5.4.2.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ16	[2]	5.4.3	M	ES3: CreateISDP Description: This function allows the SM-DP to request the creation of an ISD-P to the SM- SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. Function flow Upon reception of the function request, the SM-SR shall perform the following minimum set of verifications: • The SM-SR is responsible for the management of the targeted eUICC • The Profile identified by its ICCID is not already present within its EIS database (meaning allocated to another ISD-P) • The requested amount of memory can be satisfied SM-SR may provide additional verifications The SM-SR receiving this request shall process it according to the “Profile Download and Installation” procedure described in the section 3.1 of this specification. When the SM-SR ends successfully this function it shall update the eUICC EIS by adding a new Profile entry in the EIS. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the ISD-P has been successfully created on the eUICC as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.3.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ17	[2]	5.4.4	M	ES3: SendData Description: This function allows the SM-DP to send securely commands defined in ES8 interface (i.e.: Profile download or establish a key set) to an ISD-P or the ISD-R thru the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. Function flow Upon reception of the function request, the SM-SR shall perform the following minimum set of verifications: • The SM-SR is responsible for the management of the targeted eUICC • The targeted ISD-P is created on the eUICC. SM-SR may provide additional verifications The data provided by the SM-DP shall be a list of C-APDU as defined in ETSI TS 102 226 [5] section 5.2.1. The SM-SR has the responsibility to build the final Command script, depending on eUICC capabilities and selected protocol: • by adding the Command scripting template for definite or indefinite length • and, if necessary, by segmenting the provided command script into several pieces and adding the relevant Script chaining TLVs This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the function has been successfully executed by the function provider as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.4.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ18	[2]	5.4.5	M	ES3: ProfileDownloadCompleted Description: This function allows the SM-DP to indicate to the SM-SR that the Profile download (identified by its ICCID) has been completed on the eUICC; eUICC being identified by its EID. The Subscription Address is the identifier, such as MSISDN and/or IMSI, through which the eUICC is accessible from the SM-SR via the mobile network when the Profile is in Enabled state. On reception of this function request the SM-SR shall immediately update the EIS to set the identified Profile: • (Conditional) the new Subscription Address. If the Profile is to be Enabled after it is loaded then the Subscription Address becomes mandatory. • (Optional) the provided POL2 At the end of this function call, the Profile state is “Disabled”. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the function has been correctly executed • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.5.	Profile Management
PM_REQ19	[2]	5.4.6	M	ES3: UpdatePolicyRules Description: This function allows the SM-DP authorized by the MNO to update POL2 of a Profile, identified by its ICCID, and installed on an eUICC identified by its EID. The function can update a Profile in “Disabled” or “Enabled” state and shall return an error for any other Profile state. The function completely replaces the definition of existing POL2. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the update Policy Rules function has been successfully executed by the SM-SR as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.6.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ20	[2]	5.4.7	M	ES3: UpdateSubscriptionAddress Description: This function enables the caller to update the Subscription Address for a Profile in the eUICC Information Set (EIS) of a particular eUICC identified by the EID and ICCID. The Subscription Address is the identifier, such as MSISDN and/or IMSI, through which the eUICC is accessible from the SM-SR via the mobile network when the Profile is in Enabled state. The function replaces the content of the Subscription Address. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the UpdateSubscriptionAddress function has been successfully executed by the SM-SR as requested by the function caller • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.7.	Profile Management
PF_REQ18	[2]	5.4.8	M	ES3: EnableProfile Description: This function allows the SM-DP to request a Profile Enabling to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. The target Profile is managed by the SM-DP authorized by the MNO owner of the Profile. The SM-SR receiving this request shall process it according to “Profile Enabling via SM-DP” procedure described in the section 3.3 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been Enabled on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.8.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ19	[2]	5.4.9	M	ES3: DisableProfile Description: This function allows the SM-DP authorized by the MNO to request a Profile Disabling to the SM-SR in charge of the management of the targeted eUICC, eUICC being identified by its EID. The target Profile shall be owned by the requesting MNO. The SM-SR receiving this request shall process it according to Profile Disabling procedure described in section 3.5 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been Disabled on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.9.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ20	[2]	5.4.10	M	ES3: DeleteISDP Description: This function allows the SM-DP to request deletion of the target ISD-P with the Profile to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. The target Profile can only be a Profile that can be managed by the SM- DP authorized by the MNO. On reception of the function request, the SM-SR shall perform the following minimum set of verifications: • The SM-SR is responsible for the management of the targeted eUICC • The ISD-P identified by its AID exits on the targeted eUICC • The SM-DP is authorized to delete the target Profile by the MNO owning the target Profile • The POL2 of the target Profile allows the deletion • The target Profile is not the Profile having the Fall-back Attribute The SM-SR receiving this request shall process it according to “Profile and ISD-P deletion via SM-DP” procedure described in section 3.7 of this specification. In case the target Profile is “Enabled”, the SM-SR shall automatically disable it before executing the deletion. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been deleted on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.10.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ21	[2]	5.4.11	M	ES3: UpdateConnectivityParameters Description: This function allows the MNO, or the SM-DP authorized by the MNO to update the Connectivity Parameters store in the ISD-P, identified by its ICCID, and installed on an eUICC identified by its EID. The function can update a Profile in “Disabled” or “Enabled” state and shall return an error for any other Profile state. The function updates the definition of existing Connectivity Parameters. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the update of the Connectivity Parameters function has been successfully executed by the SM-SR as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.4.11.	Profile Management
PF_REQ21	[2]	5.4.12	M	ES3: HandleProfileDisabledNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been Disabled on the eUICC identified by its EID. ICCID may be not enough to identify right address of recipient; SM-SR should map it internally to MNO notification endpoint. This notification also conveys the date and time specifying when the operation has done. In case of multiply handlers are served SM-SR should ensure completionTimestamp to be equal for every message. Input data described in Tables present in section 5.4.12.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ22	[2]	5.4.13	M	ES3: HandleProfileEnabledNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been Enabled on the eUICC identified by its EID. ICCID may be not enough to identify right address of recipient; SM-SR should map it internally to MNO notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case of multiply handlers are served SM-SR should ensure completionTimestamp to be equal for every message. Input data described in Tables present in section 5.4.13.	Platform Management
EUICC_REQ34	[2]	5.4.14	M	ES3: HandleSMSRChangeNotification Description: This function shall be called for notifying each SM-DP authorized by the MNO owning a Profile hosted in the eUICC, identified by its EID that the SM-SR has changed. The notification is sent by the new SM-SR to the SM-DP, which shall route this notification to the MNO. This notification also conveys the date and time specifying when the operation has been done. Input data described in Tables present in section 5.4.14.	eUICC Management
PF_REQ23	[2]	5.4.15	M	ES3: HandleProfileDeletedNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been deleted on the eUICC identified by its EID. ICCID may be not enough to identify right address of recipient; SM-SR should map it internally to SM-DP notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case of multiply handlers are served, SM-SR should ensure 'completionTimestamp' to be equal for every message. Input data described in Tables present in section 5.4.15.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ22	[2]	5.5.1	M	ES4: GetEIS Description: This function allows retrieving the eUICC Information Set (EIS) of a particular eUICC from the SM-SR information database based on the EID. The retrieved EIS contains only the data that is applicable for that particular MNO. This function may return: • A 'Function execution status' with 'Executed-success' indicating that the download function has been successfully executed on the SM-SR as requested by the function caller • A 'Function execution status' with 'Expired' with a status code as defined in section 5.1.6.4 • A 'Function execution status' indicating 'Failed' with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.5.1.	Profile Management
PM_REQ23	[2]	5.5.2	M	ES4: UpdatePolicyRules Description: This function allows the MNO to update POL2 of a Profile, identified by its ICCID, and installed on an eUICC identified by its EID. Input/Output data described in section 5.4.6.	Profile Management
PM_REQ24	[2]	5.5.3	M	ES4: UpdateSubscriptionAddress Description: This function enables the caller to update the Subscription Address for a Profile in the eUICC Information Set (EIS) of a particular eUICC identified by the EID and ICCID. The function replaces the content of the Subscription Address. This function may return: • A 'Function execution status' with 'Executed-success' indicating that the UpdateSubscriptionAddress function has been successfully executed by the SM-SR as requested by the function caller • A 'Function execution status' indicating 'Failed' with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.5.13.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ25	[2]	5.5.4	M	ES4: AuditEIS Description: This function allows the MNO to retrieve the up to date information for the MNO's Profiles. The SM-SR shall only provide information for the Profiles owned by the requesting MNO. The SM-SR shall use the relevant functions of the ES5 interface to retrieve the information from the eUICC. The SM-SR shall update its EIS database upon the basis of this information. Input/Output data described in Tables present in section 5.5.4.	Profile Management
PM_REQ26	[2]	5.5.4	M	ES4: AuditEIS If no list of ICCIDs is provided, it is implied that all the EIS data for the Profiles owned by the requesting MNO is required.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ24	[2]	5.5.5	M	ES4: EnableProfile Description: This function allows the MNO to request a Profile Enabling to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. The target Profile is managed by the MNO. On reception of the function request, the SM-SR shall perform the following minimum set of verifications: • The SM-SR is responsible for the management of the targeted eUICC • The Profile identified by its ICCID is loaded on the targeted eUICC • The target Profile is owned by the requesting MNO • The target Profile is in Disabled state • The POL2 of the target Profile and the POL2 of the currently Enabled Profile allow the enabling The SM-SR receiving this request shall process it according to “Profile enabling” procedure described in the section 3.2 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been Enabled on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code indicating a Unknown eUICC or with a status code indicating a Unknown ICCID with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.5.5.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ25	[2]	5.5.6	M	ES4: DisableProfile Description: This function allows the MNO to request a Profile Disabling to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. The targeted is owned by the requesting MNO. The SM-SR receiving this request shall process it according to “Profile disabling” procedure described in section 3.4 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been Disabled on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.5.6.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ26	[2]	5.5.7	M	ES4: DeleteProfile Description: This function allows the MNO to request deletion of the target ISD-P with the Profile to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. The target Profile can only be a Profile owned by the requesting MNO. On reception of the function request, the SM-SR shall perform the following minimum set of verifications: • The SM-SR is responsible for the management of the targeted eUICC • The ISD-P identified by its AID exists on the targeted eUICC • The POL2 of the target Profile allows the deletion • The target Profile is not the Profile having the Fall-back Attribute • The target Profile is owned by the requesting MNO and the function request is authorized by the MNO owning the target Profile The SM-SR receiving this request shall process it according to “ISD-P Deletion” procedure described in the section 3.6 of this specification. In case the target Profile is “Enabled”, the SM-SR shall automatically disable it before executing the deletion. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the Profile has been deleted on the eUICC • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.5.7.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ35	[2]	5.5.8	M	ES4: PrepareSMSRChange Description: This function allows the Initiator to request to a new SM-SR to prepare for a change for an eUICC identified by its EID. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the PrepareSMSRChange function has been successfully executed on the SM-SR as requested by the function caller • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.5.8.	eUICC Management
EUICC_REQ36	[2]	5.5.9	M	ES4: SMSRChange Description: This function allows the initiator to request to the current SM-SR to change for a specific eUICC identified by its EID. The SM-SR receiving this request shall process it according to the “SM-SR Change” procedure described in GSMA Remote Provisioning Architecture for Embedded UICC [1]. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the function has been successfully executed by the function provider as requested by the function caller • A ‘Function execution status’ indicating ‘Expired’ with the status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.5.9.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ27	[2]	5.5.10	M	ES4: HandleProfileDisabledNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been Disabled on the eUICC identified by its EID. ICCID may be not enough to identify right address of recipient; SM-SR should map it internally to MNO notification endpoint. This notification also conveys the date and time specifying when the operation has done. In case of multiply handlers are served SM-SR should ensure completionTimestamp to be equal for every message. Input data described in Tables present in section 5.5.10.	Platform Management
PF_REQ28	[2]	5.5.11	M	ES4: HandleProfileEnabledNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been Enabled on the eUICC identified by its EID. ICCID may be not enough to identify right address of recipient; SM-SR should map it internally to MNO notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case of multiply handlers are served SM-SR should ensure completionTimestamp to be equal for every message. Input data described in Tables present in section 5.5.11.	Platform Management
EUICC_REQ37	[2]	5.5.12	M	ES4: HandleSMSRChangeNotification Description: This function shall be called for notifying each MNO owning a Profile hosted in the eUICC, identified by its EID, that the SM-SR has changed. The notification is sent by the new SM-SR. This notification also conveys the date and time specifying when the operation has been done. Input data described in Tables present in section 5.5.12.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ29	[2]	5.5.13	M	ES4: HandleProfileDeletedNotification Description: This function shall be called to notify that the Profile identified by its ICCID has been deleted on the eUICC identified by its EID. ICCID may be not enough to identify right address of recipient; SM-SR should map it internally to MNO notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case of multiply handlers are served SM-SR should ensure ‘completionTimestamp’ to be equal for every message. Input data described in Tables present in section 5.5.13.	Platform Management
EUICC_REQ38	[2]	5.6.1	M	ES7: CreateAdditionalKeySet Description: This function enables a new SM-SR to request for a new key set to be created in the ISD-R for the eUICC identified by the EID. The new keyset belongs the new SM-SR and is unknown to the current SM-SR. The current SM-SR shall map this function onto the second STORE DATA command in the ES5.establishISDRKeySet, see section 4.1.1.8. The following parameters used within this command are not provided by the new SM-SR and it is the current SM-SR’s responsibility to set these parameters as defined below. • Key Usage Qualifier shall be set to '10' (3 secure channel keys) • Key Access shall be set to ‘00’ (The key may be used by the Security Domain and any associated Application) • Key Type shall be set to ‘88’ (AES) This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the function has been successfully executed by the function provider as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.6.1.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ39	[2]	5.6.2	M	ES7: HandoverEUICC Description: This function enables to request for the handover management of an eUICC represented by its eUICC Information Set (EIS). The EIS contains the complete set of data including information about Profiles, audit trail, which is applicable for the SM-SR to manage the lifecycle of this eUICC The function provider shall execute the function accordingly to the procedure detailed in section 3.8. The handover is only committed at the end of the successfully procedure execution. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the register eUICC function has been successfully executed on the SM-SR as requested by the function caller. • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.6.2.	eUICC Management
EUICC_REQ40	[2]	5.6.3	M	ES7: AuthenticateSMSR Description: This function is used to authenticate the new SM-SR to the eUICC identified by the EID. The function will return the random challenge generated by the eUICC to be used to create the signature for the second step in the SM-SR key establishment procedure. This function may return: • A ‘Function execution status’ with ‘Executed-success’ indicating that the AuthenticateSMSR function has been successfully executed by the SM-SR as requested by the function caller • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section 5.1.6.4 • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 Input/Output data described in Tables present in section 5.6.3.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ41	[2]	5.6.4	M	ES7: HandleSMSRChangeNotification Description: This function shall be called for notifying the new SM-SR owning the eUICC, identified by its EID, that the old SM-SR has deleted the EIS of the eUICC. The notification is sent by the old SM-SR. This notification also conveys the date and time specifying when the operation has been done. Input data described in Tables present in section 5.6.4.	eUICC Management
SEC_REQ23	[1]	2.4	M	The eUICC shall implement the Milenage network authentication algorithm.	Security
SEC_REQ1	[1]	4.4.1	M	Past or future communications associated with Profile download and installation, between the SM-DP and the eUICC, whenever trappable by third party shall not be recoverable based upon the compromise of a single long-term key used for message encryption. *Note: Related to Secure Channel Protocols: this requirement is considered as superseded*	Security
SEC_REQ6	[1]	4.4.2	M	Communication between the SM-SR and the eUICC shall be protected against replay attacks.	Security
SEC_REQ9	[1]	4.4.2	M	When two security realms are exchanging data, they shall at first engage a security negotiation (e.g. EAP, IPSEC, TLS handshake…) resulting in the application of an agreed security level between them. *Note: Related to TLS: initial states already defined, so this requirement is considered as superseded*	Security
SEC_REQ11	[1]	4.4.2	M	When negotiating a communication, at least the lowest acceptable common cryptographic suite shall apply. *Note: Related to TLS: initial states already defined, so this requirement is considered as superseded*	Security
SEC_REQ12	[1]	4.4.3	M	Upon Profile deletion, the eUICC shall ensure of the complete wipe of the Profile.	Security

ID	Source	Chapter	Support	Description	Functional group
SEC_REQ13	[1]	4.4.3	M	eUICC shall only accept Platform and Profile Management commands sent from an authorized SM-SR or SM-DP. *Note: In the context of this specification, an authorized SM-SR or SM-DP is a platform that knows the keys that allow communicating with the eUICC. As consequence, initial states and requirements are already defined, so this requirement is considered as superseded*	Security
SEC_REQ14	[1]	4.4.3	M	eUICC shall reject any Platform and Profile Management commands that are in conflict with the Policy Rules of any Profile on the eUICC the only exception being for the master delete command.	Security
SEC_REQ15	[1]	4.4.3	M	The eUICC shall provide a secure way for the SM-DP and SM-SR to check its identity and status in such a way that the entity has a proof of identity and origin. This capability is offered through the Eligibility Verification function.	Security
SEC_REQ19	[1]	4.4.4	M	The donor SM-SR shall not be able to access the eUICC once the SM-SR switch procedure has been completed.	Security
SEC_REQ20	[1]	4.4.4	M	The MNO shall be able to update the OTA Keys in its Profile on the eUICC in a secure and confidential way reusing existing OTA Platform mechanisms.	Security
SEC_REQ22	[1]	4.4.6	M	Policy Rule transport shall be treated as per SR2 (SR2=Communication between the SM-SR and the eUICC shall be protected against replay attacks). *Note: Related to Secure Channel Protocols: this requirement is considered as superseded*	Security
Requirements related to the conditional requirement EUICC_REQ14 - HTTPS supported on eUICC					
EUICC_REQ42	[2]	2.4.3.1	C	The SM-SR shall make use of a special SMS for triggering the opening of an HTTPS session to the eUICC. This SMS shall be addressed to the ISD-R. The necessary TAR information shall be included in the EIS. The SMS shall comply with the format described in: GlobalPlatform Card Specification Amendment B [8], section “Administration session triggering parameters”.	eUICC Management
EUICC_REQ43	[2]	2.4.4.1.1	C	The eUICC shall support the Transport Layer Security (TLS) protocol v1.2 [15] with at least one of the following Pre-Shared Key Cipher suites as defined in RFC 5487 [17]: • TLS_PSK_WITH_AES_128_GCM_SHA256 • TLS_PSK_WITH_AES_128_CBC_SHA256	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ55	[2]	2.4.4.1.1	C	The eUICC ISD-R shall be configured with 'i' = '04' to indicate only TLS 1.2 supported as defined in GlobalPlatform Amd B [8].	eUICC Management
EUICC_REQ56	[2]	2.4.4.1.1	C	Session resumption and several parallel sessions shall not be supported.	eUICC Management
EUICC_REQ44	[2]	2.4.4.1.1	C	The eUICC shall support the Transport Layer Security (TLS) protocol v1.2 [15] with the following Pre-Shared Key Cipher suites as defined in RFC 5487 [17]: TLS_PSK_WITH_AES_128_CBC_SHA256 *Note: Replaced by EUICC_REQ43*	eUICC Management
EUICC_REQ45	[2]	2.4.4.1.2	C	As specified in RFC 4279 [16], the PSK Identity shall be first converted to a character string, and then sent encoded in octets using UTF-8 [18] by the eUICC. In the context of this specification, the PSK Identity before conversion is a sequence of Tag/Length/Value (TLV) objects in hexadecimal string representation.	eUICC Management
EUICC_REQ46	[2]	2.4.4.2	C	The ISD-R shall strictly follow GlobalPlatform Card Specification Amendment B [8] for the format of the POST request	eUICC Management
EUICC_REQ47	[2]	2.4.4.2	C	The content of the HTTP POST header field X-Admin-From shall be filled with the "Agent Id" information standardized in GlobalPlatform Card Specification Amendment B [8], section "Administration Session Triggering Parameters" (the format of this field is not standardized). "Agent Id" information shall include two parts: • the eUICC identifier (EID) • the identifier of the Security Domain representing the Admin Agent function	eUICC Management
EUICC_REQ48	[2]	2.4.4.2	C	The eUICC shall use the Chunked mode [Transfer-Encoding: chunked CRLF] for the POST request message.	eUICC Management
EUICC_REQ49	[2]	2.4.4.2	C	The SM-SR shall use Chunked mode [Transfer-Encoding: chunked CRLF] for the POST response.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ50	[2]	2.4.4.3	C	POST response sent by the SM-SR containing commands that shall be executed by the ISD-R: HTTP/1.1 200 CRLF X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF Content-Type : application/vnd.globalplatform.card-content-mgt;version=1.0 CRLF X-Admin-Next-URI: <uri of the next POST> CRLF CRLF [Command script]	eUICC Management
EUICC_REQ51	[2]	2.4.4.3	C	POST response sent by the SM-SR containing commands that shall be executed by the ISD-P: HTTP/1.1 200 CRLF X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF Content-Type : application/vnd.globalplatform.card-content-mgt;version=1.0 CRLF X-Admin-Next-URI: <uri of the next POST> CRLF X-Admin-Targeted-Application://aid/<rid>/<pix> (of the ISD-P-AID) CRLF CRLF [Command script]	eUICC Management
EUICC_REQ52	[2]	2.4.4.4	C	The commands sent to the eUICC within a secure script in HTTP messages shall be formatted in an expanded remote command structure with indefinite length coding as defined in ETSI TS 102 226 [5]. As a consequence, the eUICC will provide the answer as an expanded remote response structure with indefinite length coding.	eUICC Management
Requirements related to the conditional requirement EUICC_REQ18 - CAT_TP supported on eUICC					
EUICC_REQ53	[2]	2.4.3.2	C	The SM-SR shall make use of a special SMS for triggering the opening of a CAT_TP session to the eUICC. This SMS shall be addressed to the ISD-R. The necessary TAR information shall be included in the EIS. The SMS shall comply with the format described in: ETSI TS 102 226 [5], using the parameter “Request for BIP channel opening” and “Request for CAT_TP link establish”.	eUICC Management

Table 25: Requirements in scope

J.3 Out of Scope Requirements

Here are all the requirements' descriptions that are not covered by this Test Plan. Note that these requirements may be implemented in a future version of this Test Plan.

ID	Source	Chapter	Support	Description	Functional group
PROC_REQ15	[2]	3.10	M	The Master Delete Process must be compliant with the Figure 24 and with the procedure described in this section.	Procedure Flow
EUICC_REQ28	[2]	4.1.1.11	M	ES5: HandleDefaultNotification Default Notification Protocol Priority A protocol priority order for default notification may be defined for every Profile, using SMS, HTTPS and CAT_TP. If not defined for a Profile, the default priority order is set as SMS, HTTPS, CAT_TP.	eUICC Management
PF_REQ10	[2]	5.1.2.1	M	By providing a validity period, the function caller indicates a specific amount of time to the function provider to process the function. As a consequence, during this validity period, the function caller shall not issue the same request again as it might generate duplicate execution steps within the function provider system.	Platform Management
PF_REQ11	[2]	5.1.2.1	M	After the end of the validity period, the function provider shall no longer continue with new execution steps. It is only mandated to tell the function caller that the function processing has expired. It is then the caller responsibility to either: • Request the same function again • Or simply abandon the overall process into which the function was called	Platform Management
SEC_REQ2	[1]	4.4.1	M	All cryptographic keys shall be kept in secure environment (e.g. HSM, eUICC).	Security
SEC_REQ3	[1]	4.4.1	M	The keys used by the EUM for eUICC Certificate generation shall be stored in a secure environment (i.e. in a Hardware Security Module).	Security
SEC_REQ4	[1]	4.4.1	M	The MNO shall be able to reject to use a non-trusted system for the Embedded UICC management.	Security
SEC_REQ5	[1]	4.4.2	M	Security realms shall be identifiable and mutually authenticated for the purpose of any communication.	Security

ID	Source	Chapter	Support	Description	Functional group
SEC_REQ7	[1]	4.4.2	M	Any end to end data communication between two security realms of the eUICC ecosystem shall be origin authenticated, integrity and confidentiality protected, protected against replay attacks and non-repudiable. Non-repudiation may not apply to communication with the eUICC.	Security
SEC_REQ8	[1]	4.4.2	M	Network communication links used inside a security realm shall be dedicated – i.e. neither public network, neither mutualised. E.g. solutions such as MPLS or GRE are not considered as dedicated links; a solution such as an authenticated and secured VPN is considered as dedicated.	Security
SEC_REQ10	[1]	4.4.2	M	Security realms shall enforce filtering rules, so, that only authorized entities are granted access to allowed services.	Security
SEC_REQ16	[1]	4.4.4	M	SM-SR shall implement an access control mechanism on the request for execution of the SMSR functions only to authorized security realms.	Security
SEC_REQ17	[1]	4.4.4	M	SM-DP shall implement an access control mechanism on the request for execution of the SMDP functions only to authorized security realms.	Security
SEC_REQ18	[1]	4.4.4	M	Security realm of SM-SR and SM-DP, and eUICC interfaces shall have proper counter measures against denial of services attacks.	Security
SEC_REQ21	[1]	4.4.5	M	The machine to machine Device shall not be able to access nor modify sensitive Profile data, i.e. credentials, management commands, Policy Rules, authentication algorithm parameters.	Security

Table 26: Out of Scope Requirem

6 Document History

Version	Date	Brief description of change	Editor / Company
1.0	13 October 2014	PSMC approved, first release	Sébastien Kuras, FIME
2.0	October 2015	15ESIMWI311_01, 15ESIMWI311_02r1, 15ESIMWI311_03, 15ESIMWI311_04, 15ESIMWI311_05, 15ESIMWI311_06, 15ESIMWI311_07, 15ESIMWI311_08, 15ESIMWI311_09, 15ESIMWI311_11, 15ESIMWI311_12, 15ESIMWI311_13, 15ESIMWI312_03r1, 15ESIMWI312_07r1, 15ESIMWI312_08r1, 15ESIMWI312_09r1, 15ESIMWI312_11r1, 15ESIMWI312_12r1, 15ESIMWI312_15r1, 15ESIMWI312_16r1, 15ESIMWI312_17r1, 15ESIMWI312_18r1, 15ESIMWI312_19, 15ESIMWI312r1_20, 15ESIMWI312_21r1, 15ESIMWI313_01, 15ESIMWI313_02, 15ESIMWI313_04, 15ESIMWI313_05, 15ESIMWI313_11, 15ESIMWI313_12, 15ESIMWI313_13, 15ESIMWI313_14, 15ESIMWI313_15, 15ESIMWI313_16, 15ESIMWI313_21, 15ESIMWI313_22r3, 15ESIMWI314_01, 15ESIMWI314_02r1, 15ESIMWI314_03,	Sébastien Kuras, FIME

Version	Date	Brief description of change	Editor / Company
		15ESIMWI314_04, 15ESIMWI314_05, 15ESIMWI315_01r1, 15ESIMWI315_02, 15ESIMWI315_03, 15ESIMWI315_04, 15ESIMWI315_05r1, 15ESIMWI315_06r1, 15ESIMWI316_01, 15ESIMWI316_02, 15ESIMWI317_01, 15ESIMWI317_02, 15ESIMWI317_03, 15ESIMWI317_04, 15ESIMWI317_05, 15ESIMWI317_06, 15ESIMWI317_07, 15ESIMWI317_08, 15ESIMWI317_09, 15ESIMWI317_10, 15ESIMWI317_11, 15ESIMWI317_12	
3.0	October 2015	Third release	Sébastien Kuras, FIME
3.1	May 2016	15ESIMWI318_01, 15ESIMWI318_02, 15ESIMWI319_01, 15ESIMWI319_02, 15ESIMWI319_03, 15ESIMWI319_04, 15ESIMWI319_05r1, 15ESIMWI319_06, 15ESIMWI319_07, 15ESIMWI319_08, 15ESIMWI319_09r1, 15ESIMWI319_10, 15ESIMWI319_11, 15ESIMWI319_12r1, 15ESIMWI319_13, 15ESIMWI320_01r1, 15ESIMWI320_02r4, 16ESIMWI320_03r2, 16ESIMWI320_04, 16ESIMWI320_05, 16ESIMWI320_06	Sébastien Kuras, FIME

6.1 Document Owner

Type	Description
Document Owner	SIM Group
Editor / Company	Sébastien Kuras, FIME

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.