SGP.11 eSIM Test Specification
Version 4.2.1
22 October 2020

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

Security Classification: Non-confidential

Copyright Notice

Disclaimer

Antitrust Notice

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

Table of Contents

1 Introduction 8
1.1 Overview 8
1.2 Scope 8
1.3 Definition of Terms 8
1.4 Abbreviations 11
1.5 Document Cross-references 15
1.6 Conventions 16
2 Testing Rules 17
2.1 Applicability 17
2.1.1 Format of the Optional Features Table 17
2.1.2 Format of the Applicability Table 17
2.1.3 Applicability and Notations 17
2.1.4 Optional Features Table 18
2.1.5 Applicability Table 18
2.2 General Consideration 24
2.2.1 Test Cases Definition 24
2.2.2 Test Cases Format 25
2.2.3 Using of Methods, Constants and Dynamic Content 27
2.2.4 Commands and Responses 27
2.2.5 Referenced Requirements 28
2.2.6 Pass Criterion 28
2.2.7 Future Study 28
3 Testing Architecture 29
3.1 Testing Scope 29
3.2 Testing Execution 30
3.2.1 Interfaces Compliancy 30
3.2.2 System Behaviour 33
3.3 Void 36
3.4 Testing Rules Exceptions 36
3.5 Integrated eUICC - Test Environment 36
4 Interface Compliancy Testing 38
4.1 General Overview 38
4.2 eUICC Interfaces 38
4.2.1 Generic Sub-sequences 38
4.2.2 OTA Transport Protocols 44
4.2.3 ES5 (SM-SR – eUICC): CreateISDP 49
4.2.4 ES5 (SM-SR – eUICC): EnableProfile 55
4.2.5 ES5 (SM-SR – eUICC): DisableProfile 62
4.2.6 ES5 (SM-SR – eUICC): SetFallbackAttribute 71
4.2.7 ES5 (SM-SR – eUICC): DeleteProfile 76
4.2.8 ES5 (SM-SR – eUICC): eUICCCapabilityAudit 85
4.2.9 ES5 (SM-SR – eUICC): MasterDelete 98

4.2.10 ES5 (SM-SR – eUICC): EstablishISDRKeySet 115
4.2.11 ES5 (SM-SR – eUICC): FinaliseISDRhandover 127
4.2.12 ES5 (SM-SR – eUICC): UpdateSMSRAddressingParameters 136
4.2.13 ES5 (SM-SR – eUICC): Notification on Profile Enabling 149
4.2.14 ES5 (SM-SR – eUICC): Notification on Profile Disabling 165
4.2.15 ES6 (MNO – eUICC): UpdatePOL1byMNO 176
4.2.16 ES6 (MNO – eUICC): UpdateConnectivityParametersByMNO 184
4.2.17 ES8 (SM-DP – eUICC): EstablishISDPKeySet 191
4.2.18 ES8 (SM-DP – eUICC): DownloadAndInstallation 207
4.2.19 ES8 (SM-DP – eUICC): UpdateConnectivityParameters 222
4.2.20 ES5 (SM-SR – eUICC): SetEmergencyProfileAttribute 228
4.2.21 ESX (SM-SR – eUICC): LocalEnableEmergencyProfile 234
4.2.22 ESX (SM-SR – eUICC): LocalDisableEmergencyProfile 238
4.3 Off-card Interfaces 243
4.3.1 ES1 (EUM – SM-SR): RegisterEIS 243
4.3.2 ES2 (MNO – SM-DP): GetEIS 246
4.3.3 ES2 (MNO – SM-DP): DownloadProfile 249
4.3.4 ES2 (MNO – SM-DP): UpdatePolicyRules 257
4.3.5 ES2 (MNO – SM-DP): UpdateSubscriptionAddress 259
4.3.6 ES2 (MNO – SM-DP): EnableProfile 261
4.3.7 ES2 (MNO – SM-DP): DisableProfile 267
4.3.8 ES2 (MNO – SM-DP): DeleteProfile 272
4.3.9 ES3 (SM-DP – SM-SR): GetEIS 277
4.3.10 ES3 (SM-DP – SM-SR): AuditEIS 279
4.3.11 ES3 (SM-DP – SM-SR): CreateISDP 280
4.3.12 ES3 (SM-DP – SM-SR): SendData 282
4.3.13 ES3 (SM-DP – SM-SR): UpdatePolicyRules 284
4.3.14 ES3 (SM-DP – SM-SR): UpdateSubscriptionAddress 286
4.3.15 ES3 (SM-DP – SM-SR): UpdateConnectivityParameters 288
4.3.16 ES3 (SM-DP – SM-SR): EnableProfile 290
4.3.17 ES3 (SM-DP – SM-SR): DisableProfile 294
4.3.18 ES3 (SM-DP – SM-SR): DeleteISDP 297
4.3.19 ES4 (MNO – SM-SR): GetEIS 301
4.3.20 ES4 (MNO – SM-SR): UpdatePolicyRules 303
4.3.21 ES4 (MNO – SM-SR): UpdateSubscriptionAddress 304
4.3.22 ES4 (MNO – SM-SR): AuditEIS 306
4.3.23 ES4 (MNO – SM-SR): EnableProfile 308
4.3.24 ES4 (MNO – SM-SR): DisableProfile 311
4.3.25 ES4 (MNO – SM-SR): DeleteProfile 315
4.3.26 ES4 (MNO – SM-SR): PrepareSMSRChange 318
4.3.27 ES4 (MNO – SM-SR): SMSRchange 321
4.3.28 ES7 (SM-SR – SM-SR): HandoverEUICC 326
4.3.29 ES7 (SM-SR – SM-SR): AuthenticateSMSR 331
4.3.30 ES7 (SM-SR – SM-SR): CreateAdditionalKeySet 335
4.3.31 ES2 (MNO – SM-DP): Usage of WSA fields 338

4.3.32 ES4 (M2MSP – SM-SR): SetEmergencyProfileAttribute not authorised 342
4.3.33 ES4 (M2M SP – SM-SR): Enable Profile by M2M SP with errors 345
4.3.34 ES4 (M2M SP– SM-SR): GetPLMA 347
4.3.35 ES2 (MNO - SM-DP): AuditEIS 351
4.3.36 ES4 (MNO – SM-SR and M2MSP – SM-SR): SetFallBackAttribute not authorised 353
4.4 OTA Layer Testing 358
4.4.1 Generic Sub-Sequences 358
4.4.2 ES3 (SM-DP – SM-SR): AuditEIS 361
4.4.3 ES3 (SM-DP – SM-SR) and ES4 (MNO - SM-SR): usage of WSA fields 361
4.4.4 ES3 (SM-DP - SM-SR): DisableProfile by M2M SP (via the SM-DP of a MNO) 367
4.4.5 ES4 (MNO – SM-SR and M2MSP – SM-SR): SetFallBackAttribute authorized 371
4.4.6 ES4 (MNO – SM-SR and M2MSP – SM-SR): SetEmergencyProfileAttribute authorized 378
4.4.7 ES4 (M2M SP - SM-SR): EnableProfile by M2M SP 384
4.4.8 ES4 (M2M SP - SM-SR): EnableProfile by M2M SP with ONC set 387
4.4.9 ES4 (MNO – SM-SR): SMSRChange 390
4.4.10 ES5 (SM-SR – eUICC): CreateISDP 399
4.4.11 ES5 (SM-SR – eUICC): Profile Download Procedure 410
4.4.12 ES7 (SM-SR – SM-SR): CreateAdditionalKeyset 415
5 System Behaviour Testing 420
5.1 General Overview 420
5.2 eUICC Behaviour 420
5.2.1 Device – eUICC 420
5.2.2 LOCKED State Unsupported by ISD-R and ISD-P 421
5.2.3 Components and Visibility 424
5.2.4 Security and Responsibility 446
5.2.5 Confidential Setup of MNO Secure Channel Keys 461
5.2.6 Full Profile Installation Process 464
5.3 Platform Behaviour 468
5.3.1 eUICC Identity Check 468
5.3.2 Profile Download and Installation Process 471
5.3.3 Profile Enabling Process 483
5.3.4 Profile Disabling Process 503
5.3.5 Profile Deletion Process 522
5.3.6 Master Delete Process 530
5.3.7 SM-SR Change Process 530
5.3.8 Update Connectivity Parameters Process 548
6 Test Specifications 551
6.1 TCA eUICC Profile Package Test Specification 551
Annex A Reference Applications 552

A.1 Applet1 552
A.1.1 Description 552
A.1.2 AID 552
A.1.3 Source Code (Java Card) 552
A.2 Applet2 554
A.2.1 Description 554
A.2.2 AID 554
A.2.3 Source Code (Java Card) 554
A.3 Applet3 554
A.3.1 Description 554
A.3.2 AID 554
A.3.3 Source Code (Java Card) 554
Annex B Constants 556
B.1 Hexadecimal Constants 556
B.2 ASCII Constants 558
B.3 eUICC Settings 560
B.4 Platforms Settings 562
B.5 RPS Elements 566
B.6 Profiles Information 581
B.7 Profile Package Description 584
B.7.1 Profile Package Content 585
B.7.2 Access Rules 598
B.7.3 Additional Profile Elements 599
Annex C Dynamic Content 608
Annex D Methods 611
Annex E Commands and Responses 629
E.1 Commands 629
E.2 Responses 654
Annex F Bearer Independent Protocol 667
Annex G CAT_TP PDUs 669
Annex H TLS Records 672
Annex I Initial States 675
Annex J Requirements 679
J.1 Format of the Requirements Table 679
J.2 Requirements in Scope 679
J.3 Out of Scope Requirements 774
Annex K Integrated eUICC Testing (Normative) 797
K.1 Overview (Informative) 797
K.2 Integrated eUICC test requirements 798
K.3 USB CCID test interface 799
7 Document History 800
7.1 Document Owner 804

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 (for example SMS-C address) required by the eUICC to open a communication channel (for example 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 (for example NAA) present in the Profile are not selectable over the eUICC - Terminal interface.
Domain Name System	A internet protocol for translating domain names (or hostnames) into IP addresses.
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.

Term	Description
Enabled (Profile)	The state of a Profile when its files and/or applications (e.g. NAA) are selectable over the UICC-Terminal interface.
eUICC Certificate	A certificate issued by the EUM for a specific, individual eUICC. This certificate can be verified using the EUM Certificate.
eUICC Manufacturer	Supplier of the eUICCs and resident software (for example 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 GSMA CI Certificate.
Executable Load File	An on-card container of one or more application's executable code as defined in GlobalPlatform Card Specification [3].
Executable Module	The on-card executable code of a single application present within an Executable Load File as defined in GlobalPlatform Card Specification [3].
Fall-back Attribute	This is an attribute of a Profile which, when set, identifies the Profile to be enabled by the Fall-back Mechanism or by the execution of the Disable Profile function on another Profile. Only one Profile on the eUICC can have the Fall-back Attribute set at a time.
Fall-back Mechanism	eUICC based mechanism which enables the Profile with Fall-back Attribute set when the Enabled Profile loses network connectivity.
Integrated Circuit Card ID	Unique number to identify a Profile in an eUICC. The ICCID is coded as defined by ITU-T E.118[20].
Integrated eUICC	An eUICC implemented on a Tamper Resistant Element (TRE) that is integrated into a System-on-Chip (SoC), optionally making use of remote volatile/non-volatile memory.
Integrated eUICC Test Interface	An external interface for the purpose of testing eUICC functionality.
International Mobile Subscriber Identity	Unique identifier owned and issued by Mobile Network Operators as defined in ETSI TS 123 003 [21].
Issuer Security Domain	A security domain on the UICC as defined by GlobalPlatform Card Specification [3].
MNO-SD	Security domain part of the Profile, owned by the Operator, 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.
Mobile Network Operator	An entity providing access capability and communication services to its Customers through a mobile network infrastructure.
Network Access Application	An application residing on a UICC which provides authorization to access a network for example a USIM application.
Operator	A Mobile Network Operator or Mobile Virtual Network Operator; a company providing wireless cellular network services.

Term	Description
OTA Keys	The credentials included in the Profile, used in conjunction with OTA Platforms.
OTA Platform	An Operator platform for remote management of UICCs and the content of Enabled Operator 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.
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 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 • Connectivity Parameters
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 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.

Term	Description
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]).
RID	Registered Application Provider Identifier, the value of which is part of the AID.
Roles	Roles are representing a logical grouping of functions.
GSMA CI 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 Manager Data Preparation	Role that prepares the Profiles to be securely provisioned on the eUICC and manages the secure download and installation of these Profiles onto the eUICC.
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, for example SMS, to the eUICC.
Subscription Manager Secure Routing	Role that securely performs functions of Platform Management commands and the transport of Profile Management commands.
Telecommunication Service Provider	An entity that provides Subscriptions to Subscribers either as part of an Operator or as a party with a wholesale agreement with an Operator. The Telecommunication Service Provider could also be the Operator.
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

Abbreviation	Description
ATS	Answer To Select
BIP	Bearer Independent Protocol
C-APDU	Command APDU
CASD	Controlling Authority Security Domain
CAT_TP	Card Application Toolkit Transport Protocol
CCID	(USB) Chip Card Interface Device
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
DNS	Domain Name System
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
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

Abbreviation	Description
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
NAA	Network Access Application
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
PK.SR.ECDSA	Public Key of the SM-SR part of the CERT.SR.ECDSA, for verifying his signatures
PLMA	Profile Lifecycle Management Authorisation
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
PPK-ENC	Profile Protection Key for message encryption/decryption
PPK-MAC	Profile Protection Key for command MAC generation/verification
PPK-RMAC	Profile Protection Key for response MAC generation/verification

Abbreviation	Description
PSK	Pre-Shared Key
PUK	PIN Unblocking Key
R-APDU	Response APDU
REQ	Requirement
RFM	Remote File Management
R-MAC	Response MAC
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
SoC	System on a Chip
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
TLV	Tag, Length, Value
TON	Type Of Number
URI	Uniform Resource Identifier
USB	Universal Serial Bus
USIM	Universal Subscriber Identity Module
W3C	World Wide Web Consortium

Abbreviation	Description
XML	Extensible Markup Language

1.5 Document Cross-references

Ref	Title
[1]	GSMA SGP.01 - Embedded SIM Remote Provisioning Architecture v4.2
[2]	GSMA SGP.02 - Remote Provisioning Architecture for Embedded UICC - Technical Specification v4.2
[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 2.3.1
[17]	Trusted Connectivity Alliance (TCA) eUICC Profile Package: Interoperable Format Test Specification Version 2.3.1
[18]	GlobalPlatform Card Specification v.2.2 Amendment B: Remote Application Management over HTTP v1.1.3
[19]	RFC 2119 - Key words for use in RFCs to Indicate Requirement Levels, S. Bradner http://www.ietf.org/rfc/rfc2119.txt
[20]	ITU-T E.118 The international telecommunication charge card
[21]	ETSI TS 123 003 - Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification; Release 9
[22]	SMPP Developers Forum - SMPP Protocol Specification v3.4
[23]	Pcsc1-10 V2 01 14 .- Interoperability Specification for ICCs and Personal Computer Systems
[24]	CCID Rev 1.1 - CCID Specification for Integrated Circuit(s) Cards Interface Devices
[25]	GSMA PRD AA.35 Procedures for Industry Specifications

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", and "MAY" in this document SHALL be interpreted as described in RFC 2119 [19].

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 a unique conditional status expression which is defined in Table 5. 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.

Option	Support	Mnemonic	Entity(ies) responsible to declare the support of the feature
Support of HTTPS		O_HTTPS	eUICC-UT
Support of CAT_TP		O_CAT_TP	eUICC-UT
HTTPS enabled on the default MNO-SD		O_MNO_HTTPS	eUICC-UT
Confidential setup of default Profile keys using scenario #2.B supported		O_MNO_SC2B	eUICC-UT
Confidential setup of default Profile keys using scenario #3 supported		O_MNO_SC3	eUICC-UT
Support of DNS resolution		O_DNS	eUICC-UT
Support of SOAP on Off-Card interfaces		O_SOAP	SM-SR-UT, SM-DP-UT
Emergency Profile Management		O_EMERGENCY	eUICC-UT , SM-SR-UT, SM-DP-UT
ONC management		O_ONC	SM-SR-UT
Support of JavaCard		O_JAVACARD	eUICC-UT, SM-DP-UT

Table 4: Options

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 implies that the O_HTTPS is also supported.

The support of the optional feature O_DNS implies 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

Test case	Name	Roles	Applicability
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
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
4.2.9.2.1.7	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	C1
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	eUICC	M
4.2.12.2.2	TC.ES5.USAP.2:UpdateSMSRAddrParam_CAT_TP	eUICC	C2
4.2.12.2.3	TC.ES5.USAP.3:UpdateSMSRAddrParam_HTTPS	eUICC	C1
4.2.12.2.4	TC.ES5.USAP.4:UpdateSMSRAddrParam_DNS	eUICC	C11
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

Test case	Name	Roles	Applicability
4.2.13.2.3	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
4.2.14.2.3	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
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
4.2.20.2.1	TC.ES5.SetEmergencyProfileAttribute_SMS	eUICC	C13
4.2.20.2.2	TC.ES5.SetEmergencyProfileAttribute_HTTPS	eUICC	C14
4.2.21.2.1	TC.ESX.LocalEnableEmergencyProfile	eUICC	C13
4.2.22.2.1	TC.ESX.LocalDisableEmergencyProfile	eUICC	C13
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. DOWNP.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
4.3.6.2.1.4	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

Test case	Name	Roles	Applicability
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
4.3.29.2.1	TC.ES7.CAK.1:CreateAdditionalKeyset	SM-SR	M
4.3.31.2.1	TC.ES2.WSA.1	SM-DP	C12
4.3.32.2.1	TC.ES4.SEPA.1: SetEmergencyProfileAttribute not authorised	SM-SR	C13
4.3.33.2.1	TC.ES4. EPM2MSP.1: Enable Profile by M2M SP with errors	SM-SR	M
4.3.34.2.1	TC.ES4.GPLMA.1: Retrieve PLMA	SM-SR	M
4.3.35.2.1	TC.ES2.AEIS.1: AuditEIS via ES2	SM-DP	M
4.3.36.2.1	TC.ES4.SFBA.1: SetFallBackAttribute not authorized	SM-SR	M
OTA Layer Testing			
4.4.3.2.1	TC.ES3ES4.WSA.1	SM-SR	C12
4.4.4.2.1	TC.ES3.EPM2MSP.1: DisableProfile by M2M SP	SM-SR	M
4.4.5.2.1	TC.ES4.SFBA.2: SetFallBackAttribute authorised	SM-SR	M
4.4.6.2.1	TC.ES4.SEPA.2: SetEmergencyProfileAttribute authorised	SM-SR	C13
4.4.7.2.1	TC.ES4.EPM2MSP.2: EnableProfile by M2M SP	SM-SR	M

Test case	Name	Roles	Applicability
4.4.8.2.1	TC.ES4.EPM2MSP.3: EnableProfile by M2M SP with ONC	SM-SR	C15
4.4.9.2.1	TC.ES4.SMSRC.2: SMSRChange fails in case Handover fails or expires after authenticate SM-SR success	SM-SR	M
4.4.9.2.2	TC.ES4.SMSRC.3: SMSRChange fails in case Handover fails after CreateAdditionalKeyset success	SM-SR	M
4.4.9.2.3	TC.ES4.SMSRC.4: SMSRChange expires in case Handover doesn't complete after CreateAdditionalKeyset success	SM-SR	M
4.4.10.2.1	TC.ES5.CreateISDP.1: ISDP_Auto_Deletion	eUICC	M
4.4.10.2.2	TC.ES5.CreateISDP.2: Memory_Allocation	eUICC	M
4.4.10.2.3	TC.ES5.CreateISDP.3: Targeted_SD	eUICC	M
4.4.11.2.1	TC.ES5.ProfileDownload.1: Targeted Security Domains	eUICC	M
4.4.12.2.1	TC.ES7.CAK.1: CreateAdditionalKeyset with proper SIN/SDIN	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
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	C18
5.2.3.2.3	TC.CV.3:ISDPNotEnabled Test Sequence N°2, Test Sequence N°4	eUICC	C17
5.2.3.2.3	TC.CV.3:ISDPNotEnabled Test Sequence N°5	eUICC	C3
5.2.3.2.3	TC.CV.3:ISDPNotEnabled Test Sequence N°6	eUICC	C4
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°1	eUICC	C18
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°2	eUICC	C17
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°3	eUICC	C16
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°4	eUICC	C3
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°5	eUICC	C4
5.2.3.2.3.4	TC.CV.4:TarAllocation Test Sequence N°6	eUICC	M
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°1	eUICC	C18
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°2	eUICC	C17
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°3	eUICC	C16
5.2.3.2.5	TC.CV.5:AIDAllocation	eUICC	C3

Test case	Name	Roles	Applicability
	Test Sequence N°4		
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°5	eUICC	C4
5.2.3.2.5	TC.CV.5:AIDAllocation Test Sequence N°6	eUICC	M
5.2.3.2.5.4	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
5.2.4.2.3	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
5.3.5.2.1.3	TC.PROC.DEL.2:ProfileDeletionBySMDP	SM-DP, SM-SR	M
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

Test case	Name	Roles	Applicability
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
Test Specifications			
6.1	TCA eUICC Profile Package Test Specification	eUICC	M

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	VOID
C11	IF (O_DNS) THEN M ELSE N/A
C12	IF (O_SOAP) THEN M ELSE N/A
C13	IF (O_EMERGENCY) THEN M ELSE N/A
C14	IF (O_EMERGENCY AND O_HTTPS) THEN M ELSE N/A
C15	IF (O_ONC) THEN M ELSE N/A
C16	IF (O_JAVACARD) THEN M ELSE N/A
C17	IF (O_HTTPS AND O_JAVACARD) THEN M ELSE N/A
C18	IF (O_CAT_TP AND O_JAVACARD) THEN M ELSE N/A

Table 6: Conditional Items Referenced by Table 5

2.2 General Consideration

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

2.2.1 Test Cases Definition

Test descriptions are independent.

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

It is implicitly assumed that all entities under test SHALL be compliant with the initial states described in Annex 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.

2.2.2 Test Cases Format

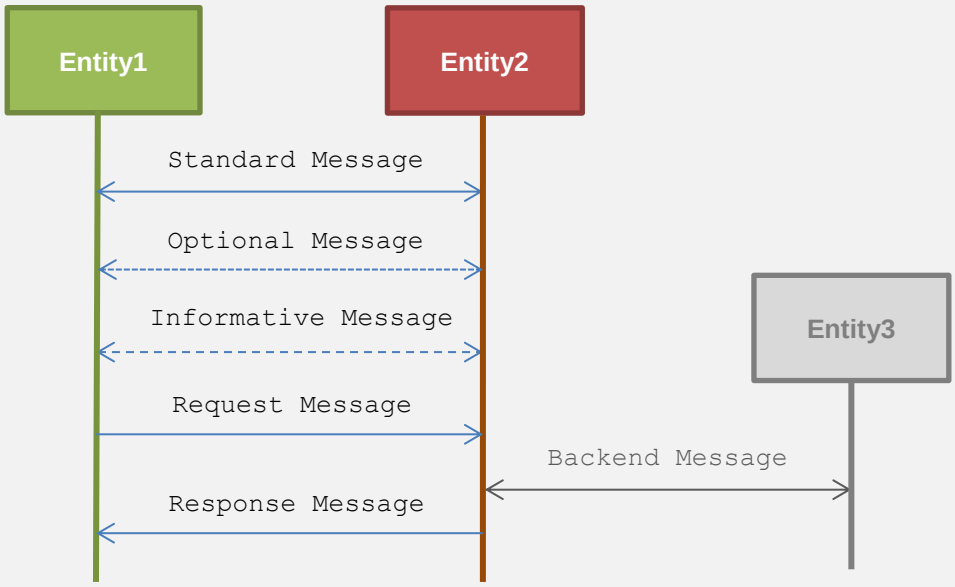
Here is an explanation of the way to define the test cases in chapters 4 and 5.

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.

2.2.3 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.

2.2.4 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.

2.2.5 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]

2.2.6 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.

2.2.7 Future Study

Some of the test cases or test sequences described in this Test Plan are FFS (For Future Study). The reason for not specifying the test case or test sequence is provided; when no reason is provided, it means the test or test sequence was assumed to be too complex in regard of the added clarification. In all cases, test and test sequences marked “FFS” SHALL NOT be executed.

3 Testing Architecture

3.1 Testing Scope

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

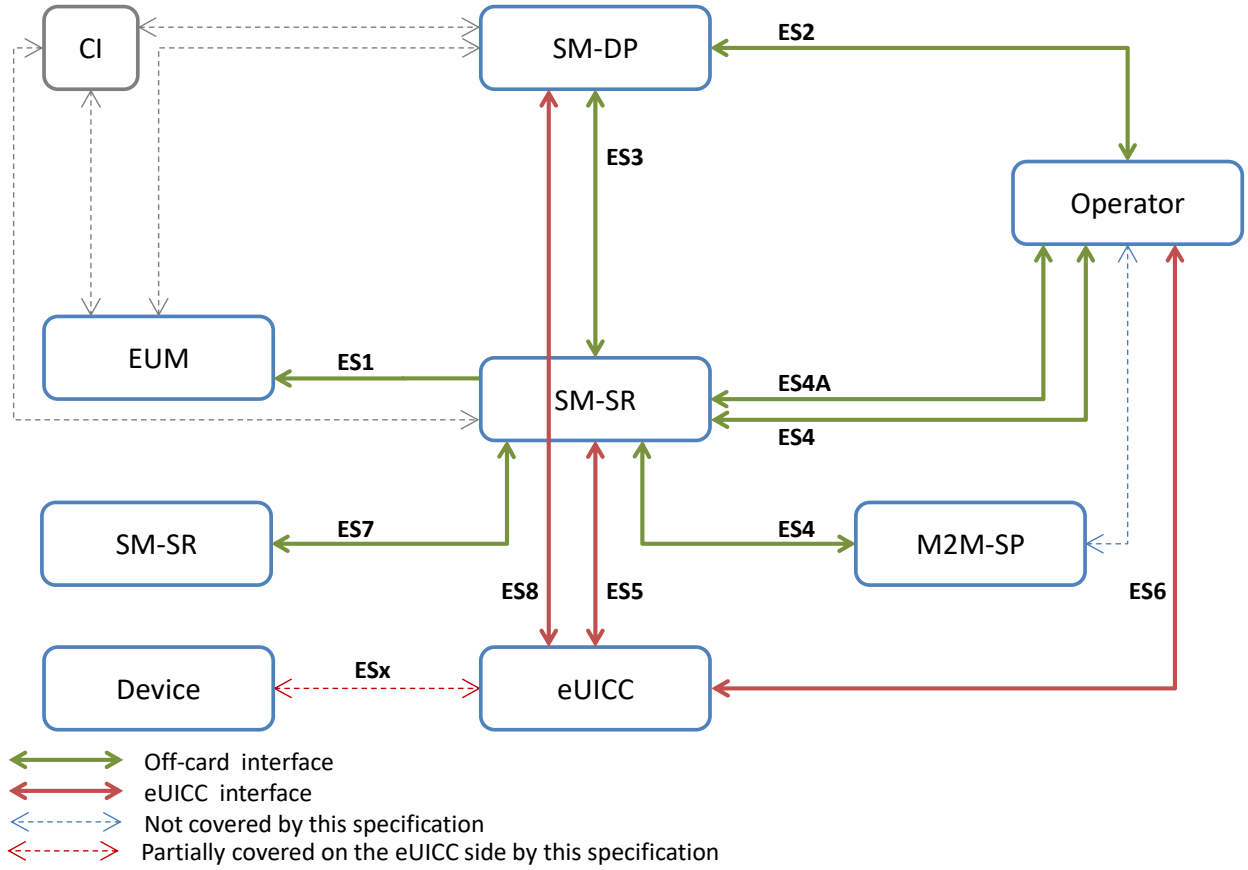


Figure 1: Scope of the Tests

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

The DNS resolution defined in SGP.02 [2], section 2.4.5, is an optional feature. Some specific tests in section 4.2.12.2.4 cover DNS resolution by an eUICC that supports it. All other eUICC test cases defined in this document are designed to be independent of this optional feature. For those other eUICC test cases, DNS resolution will be deliberately by-passed by ensuring that the ISD-R has always an IP address either configured in the Connection Parameters of the Security Domain Administration Session Parameters or supplied in the Administration Session Triggering Parameters (as defined by GlobalPlatform Amendment B [18]). As a consequence, the eUICC SHALL NOT perform any DNS resolution during the execution of the HTTPs test cases defined in sections 4.2 and 5.2 except in the specific tests in section 4.2.12.2.4.

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
- Device-Network-S: the Device and Network simulator used to simulate mobile equipment and network connectivity allowing the delivery of short messages (SCP80 over SMS) as defined in ETSI 102 225 [4] and ETSI 102 226 [6] as well as packet data transfer using SCP81 secure channel protocol as defined in ETSI 102 226 [6] and GP CS v2.2 Amd B.[18]
- M2MSP-S: the M2M SP simulator used to simulate the M2M SP and to test an 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).

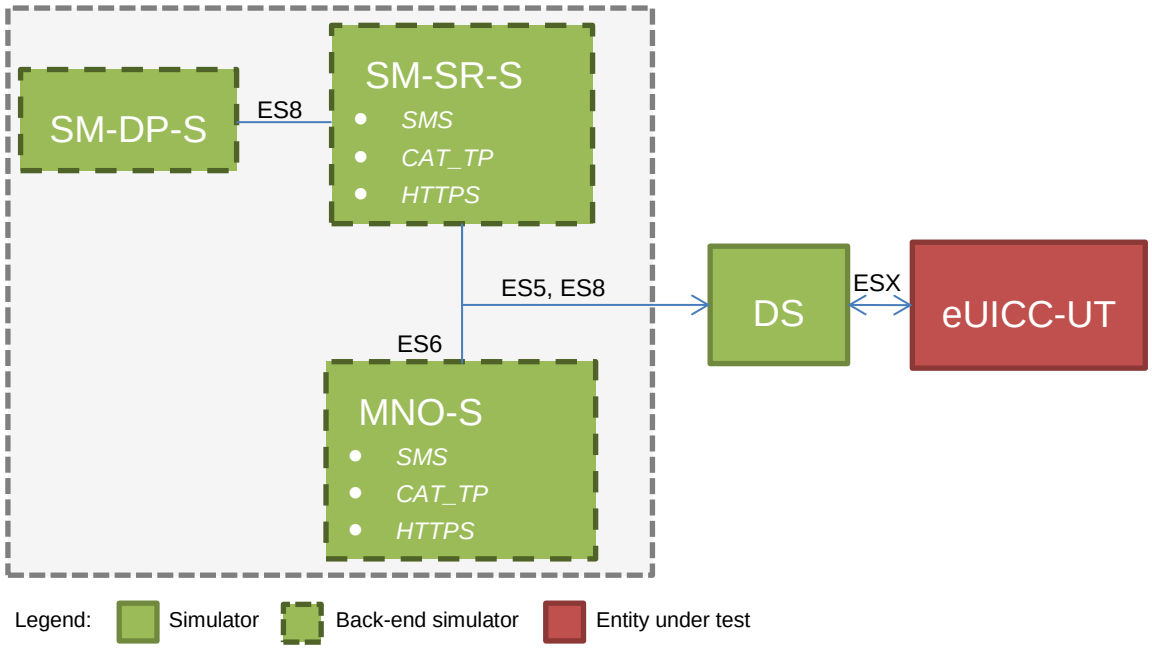


Figure 2: eUICC Interfaces Test Environment

The aim of the eUICC 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.

The Device Simulator SHALL honor any POLL INTERVAL proactive commands issued by the eUICC, and accordingly send STATUS commands at the interval requested.

The Device Simulator SHALL honor any TIMER MANAGEMENT proactive commands issued by the eUICC, and accordingly send an ENVELOPE (TIMER EXPIRATION) command after the specified time, if a timer has been activated.

3.2.1.2 Off-card Interfaces

The aim of Off-card Interfaces test cases is to verify the compliance of the server platforms for scenarios that do not require interaction with the eUICC.

The off-card test cases assume that all simulated platforms (i.e. EUM-S, MNO1-S, MNO2-S, SM-DP-S, SM-SR-S, M2M-SP-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 4.3).

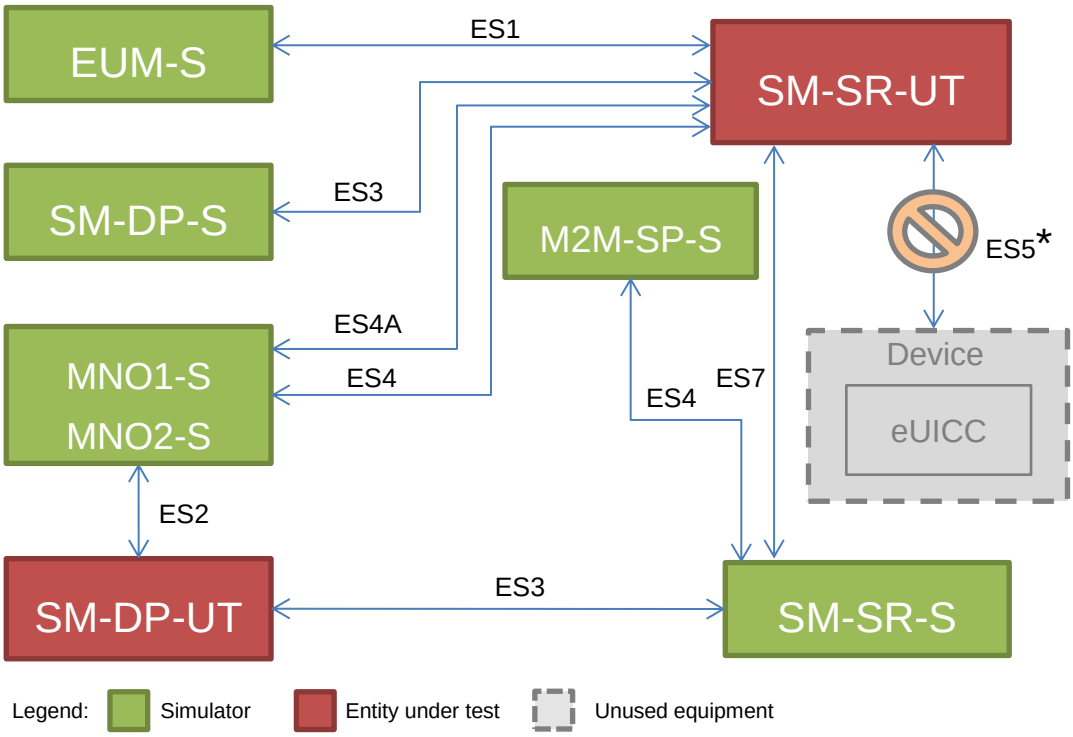


Figure 3: Off-card Interfaces Test Environment

** 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 sections "4.4 OTA layer testing" and "5* - System Behaviour Testing*", to be performed using environments defined respectively in sections "3.2.1.3 Off-card Entities Tested via eUICC Interfaces (OTA Interfaces)" and "3.2.2 System Behavior".*

3.2.1.3 Off-card Entities Tested via eUICC Interfaces (OTA Interfaces)

The aim of OTA Interface test cases is to verify that the SM-SR server platform properly supports the OTA communication with the eUICC when its off-card interfaces are triggered. The off-card test cases assume that all simulated platforms (MNO-S, SM-DP-S, M2M-SP-S, Device-Network-S) 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 4 shows the different entities used during the execution of the test cases related to the testing of the off-card entities through the on-card interfaces (see section 4.4).

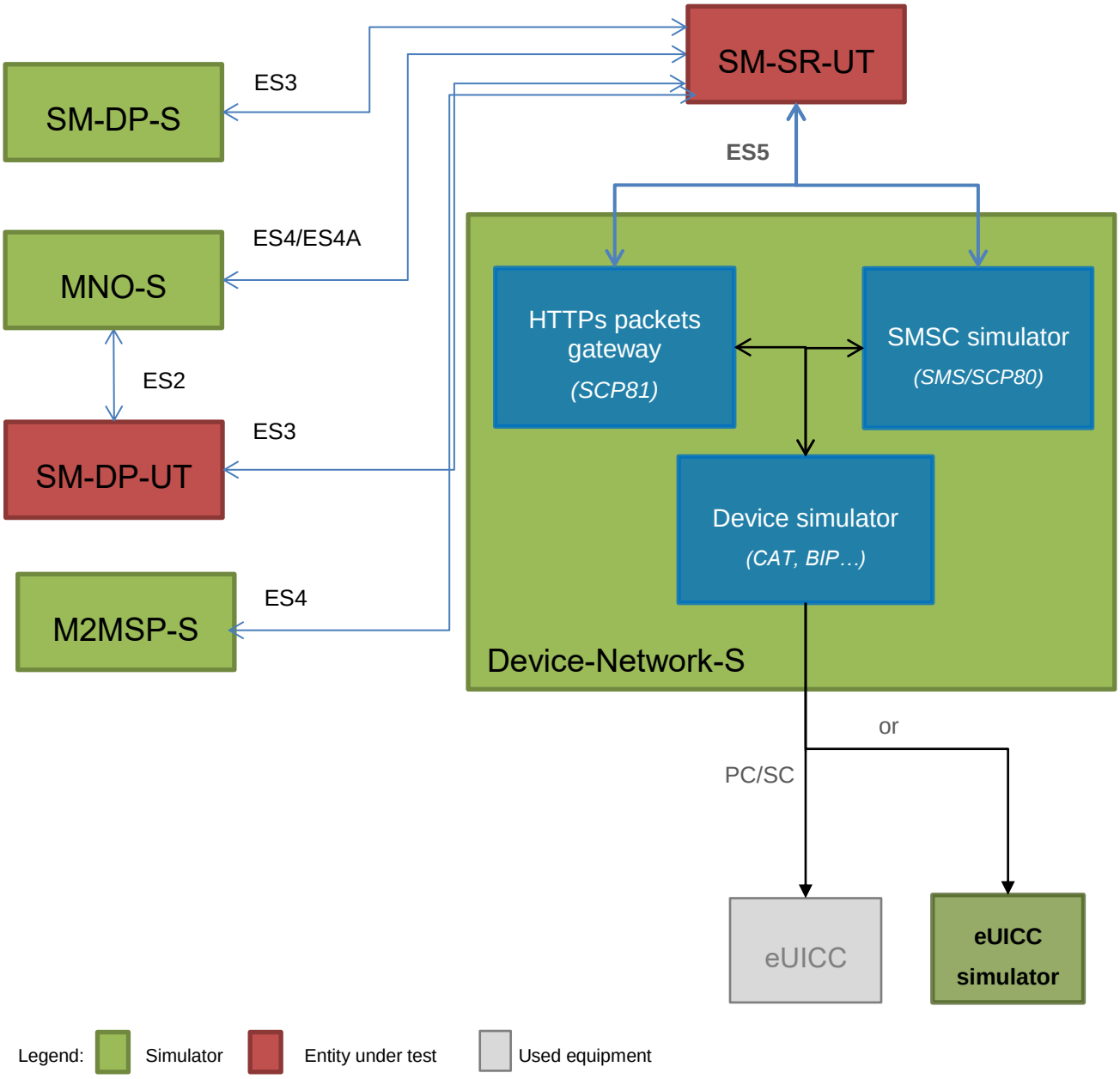


Figure 4: On-card Interfaces for Off-card Entities Test Environment

The entities inside the Device-network-S are logical grouping of functions, but the test tool provider MAY choose to not expose separate executable or interfaces for these entities.

The SMSC simulator entity SHALL support at least SMPP release 3.4 [22].

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).

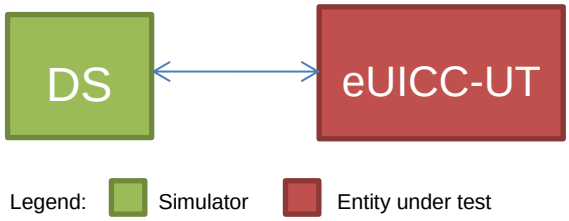


Figure 4: eUICC Behaviour Test Environment

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 5.3).

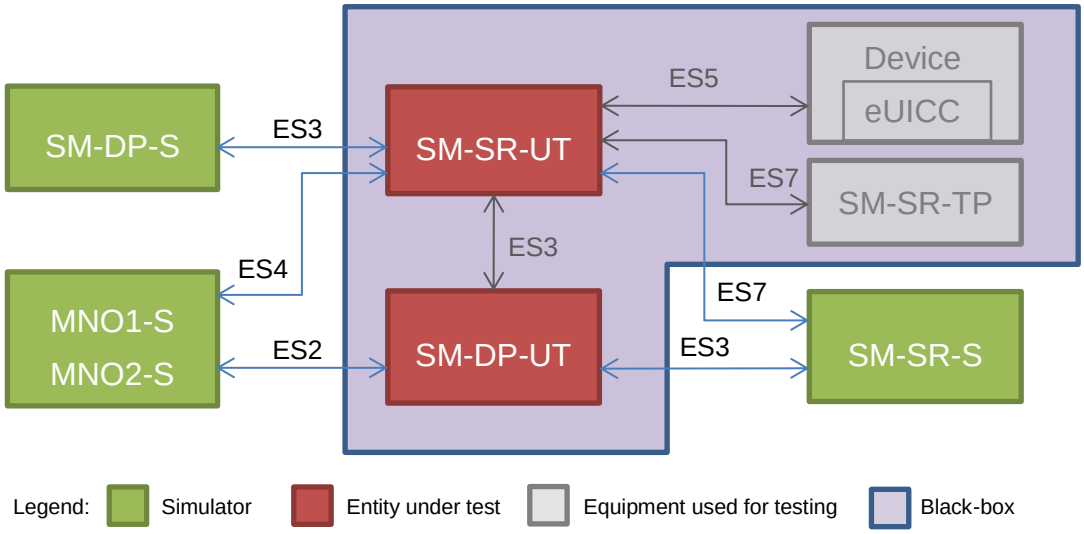


Figure 5: Platform Behaviour Test Environment

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 4.3.

- 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:

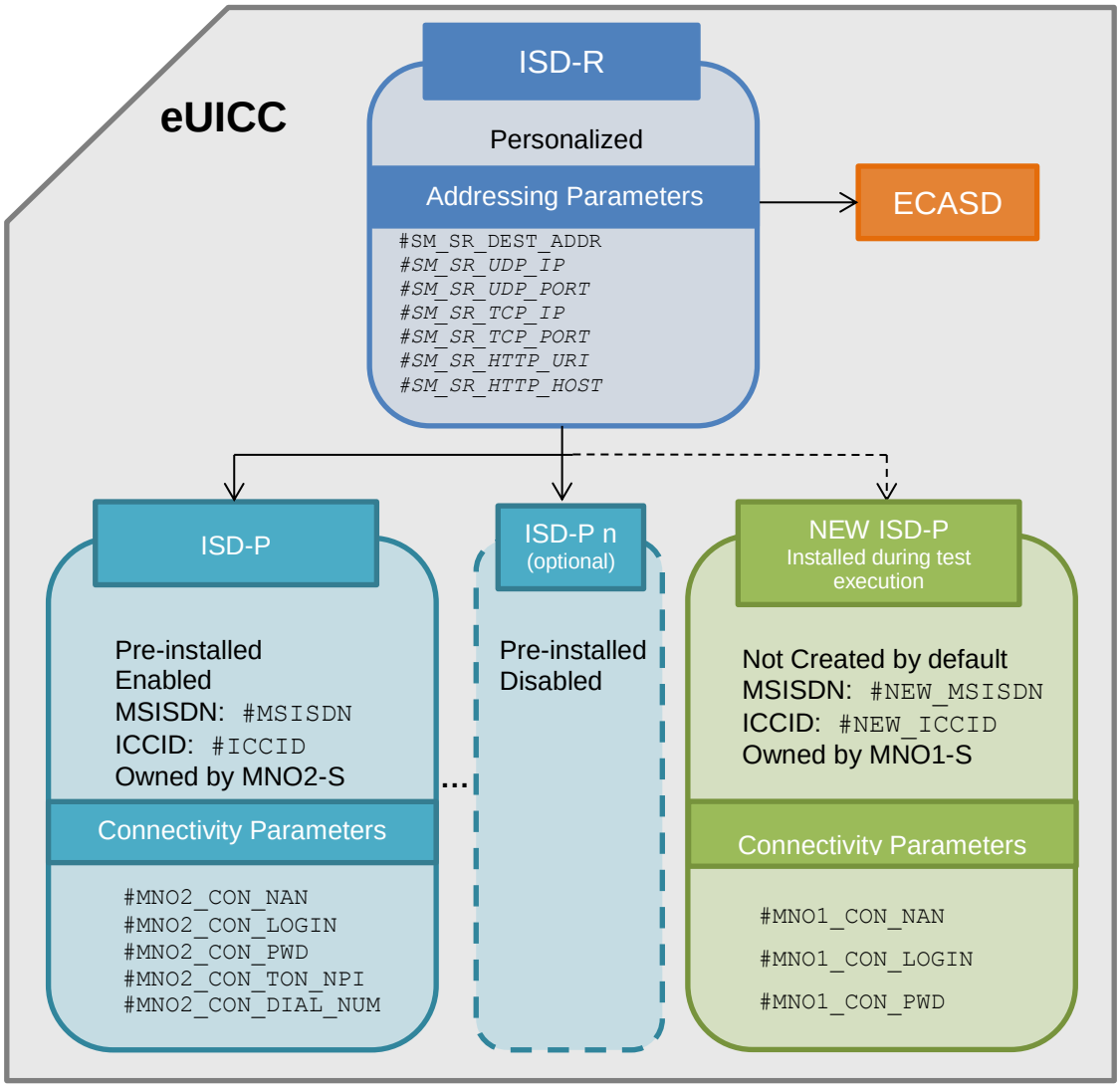


Figure 6: eUICC Configuration

The eUICC, used to execute the test cases defined in the section 5.3, 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 5.3).

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.

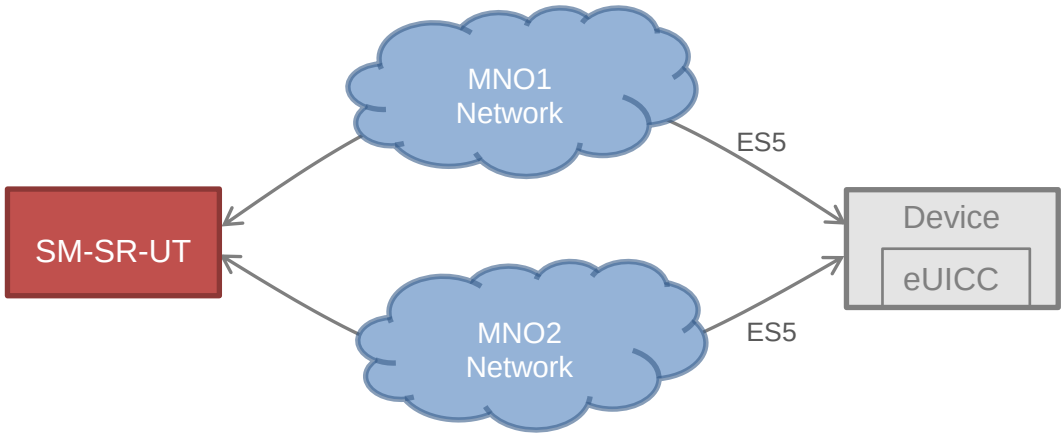


Figure 7: Required Network Access for SM-SR-UT

3.3 Void

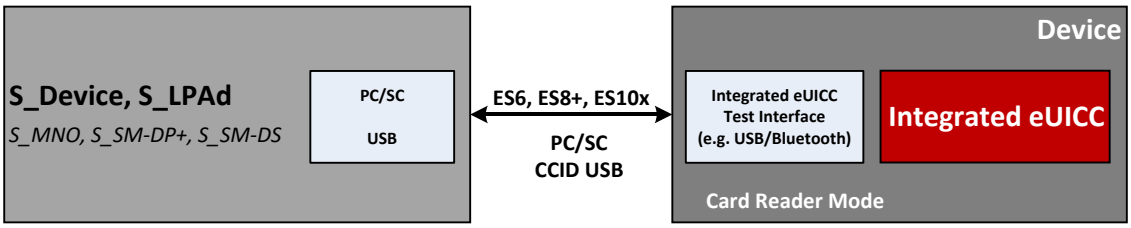
3.4 Testing Rules Exceptions

In version 4.0 of SGP.02, it is indicated that the ISD-R information present in the EIS structure SHOULD NOT be returned by the SM-SR through the ES2 and ES4 interfaces. Nevertheless, some SM-SR providers MAY decide to still return this element in order to remain backward compatible with implementations based on former versions of SGP.02/WSDL. As a consequence, even if this version of the Test Plan does not expect the `"Isd-r"` field to be part of the EIS returned over the ES2 and ES4 interfaces, the Test Tool SHALL NOT take into account this specific rule during the tests execution.

3.5 Integrated eUICC - Test Environment

The following test environment is used for eUICC test cases as defined in chapter 4 and 5 (unless it is specified differently in the specific test case). Following conditions apply:

- MNO / SM-DP / SM-SR / Device Simulators SHALL be implemented by the test tools
- Integrated eUICC shall provide a test interface, which includes one of the following:
  - USB CCID [24]
  - ISO/IEC 7816-4 [10]
- For Integrated eUICC providing a USB CCID [24] test interface, the provisions on Annex H SHALL apply.
- For Integrated eUICC providing ISO/IEC 7816-4 [10], the requirements of 3.2. shall apply



The reference of this Test Environment is TE_Integrated eUICC.

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 Annex H.*				
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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#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 Annex H.*				
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 #HOST #X_ADMIN_PROTOCOL #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

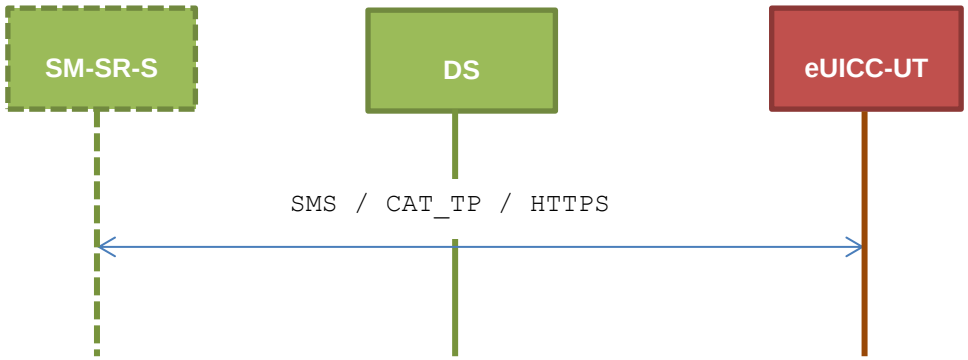
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ21_1, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ53, EUICC_REQ54

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

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 `#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

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- Verify the cryptographic checksum using #SCP80_AUTH_KEY 4- The response data is in expanded format with definite length	EUICC_REQ13, EUICC_REQ18
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_REQ21_1, 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			

Step	Direction	Sequence / Description	Expected result	REQ
3	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_DEFAULT_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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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			

4.2.2.2.3.2 Test Sequence N°2 – Nominal Case: No POR required in the SMS for HTTPS session triggering

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_NO_POR, #ISD_R_TAR, [OPEN_SCP81_SESSION])	No POR sent by the eUICC	EUICC_REQ22, EUICC_REQ42, EUICC_REQ54, EUICC_REQ21_1
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 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

Step	Direction	Sequence / Description	Expected result	REQ
6	DS → eUICC-UT	TERMINAL RESPONSE		
7	Execute the generic sub-sequence “Open HTTPS Session on ISD-R” from step 10 to step 14 (as described in section 4.2.1.5)			

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

9	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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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

10	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

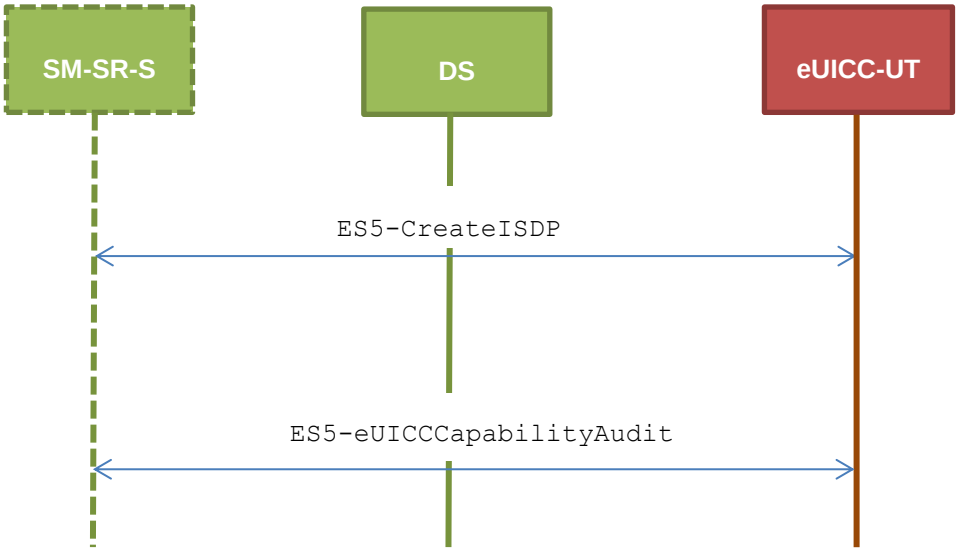
- PF_REQ3, PF_REQ7
- EUICC_REQ4, EUICC_REQ12, EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

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

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 #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

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 #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 #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 #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

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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

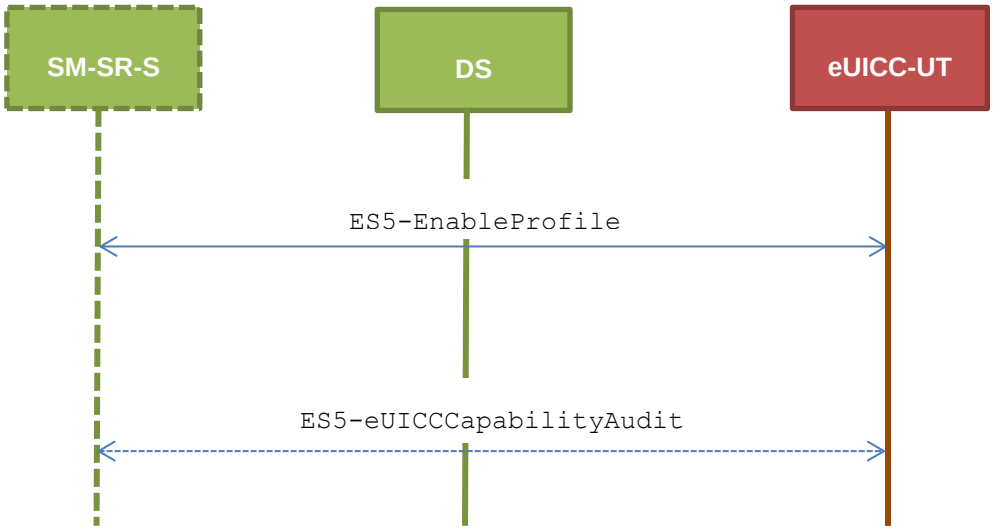
- PF_REQ4, PF_REQ7
- SEC_REQ14
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

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

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_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		
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_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		
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_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 `#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	

Step	Direction	Sequence / Description	Expected result	REQ
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

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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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
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

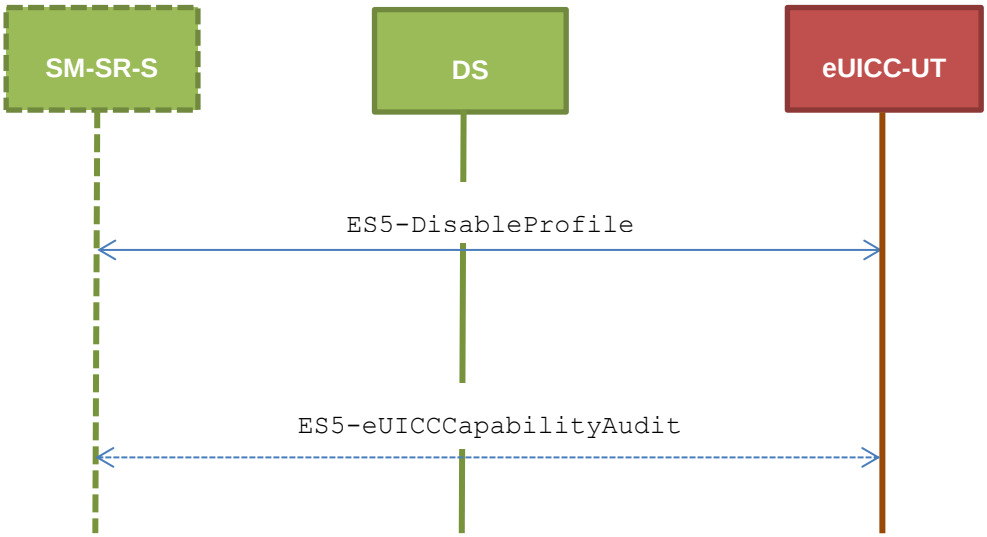
- PF_REQ5, PF_REQ7
- SEC_REQ14
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

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

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		
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_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

Step	Direction	Sequence / Description	Expected result	REQ
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

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		
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_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		

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_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

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		
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_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		

Step	Direction	Sequence / Description	Expected result	REQ
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

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		
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_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		

Step	Direction	Sequence / Description	Expected result	REQ
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

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, [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 #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

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

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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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	

Step	Direction	Sequence / Description	Expected result	REQ
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

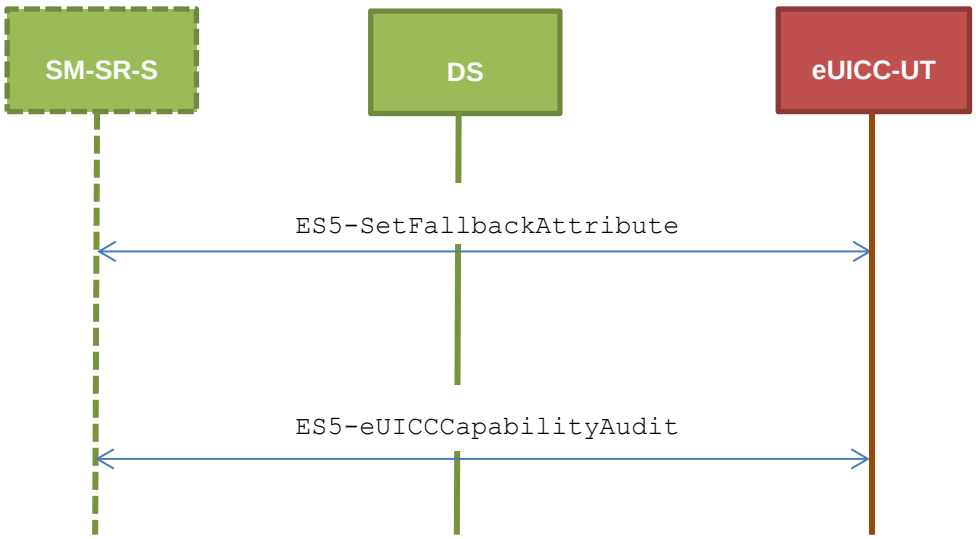
- PF_REQ7, PF_REQ9
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

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

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			
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [SET_FALLBACK])		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 `#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 #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 #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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

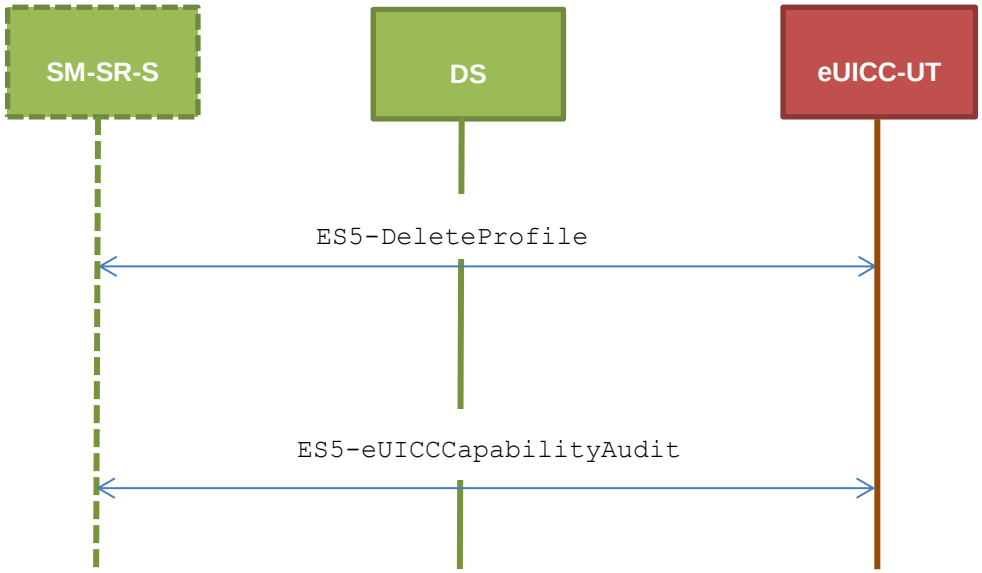
- GSMA Embedded SIM Remote Provisioning Architecture [1]

- 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

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

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

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

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’	

4.2.7.2.1.5 Test Sequence N°5 – Error Case: ISD-P not present on the eUICC

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
- The Profile identified by the ISD-P AID `#ISD_P_AID_UNKNOWN` is not 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,`
		`[DELETE_ISDP_UNKNOWN])`
				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_6A88]` Note: Status code 6A82 MAY also be returned.	PF_REQ6, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
6	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

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 `#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

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 `#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

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(`
		`[DELETE_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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

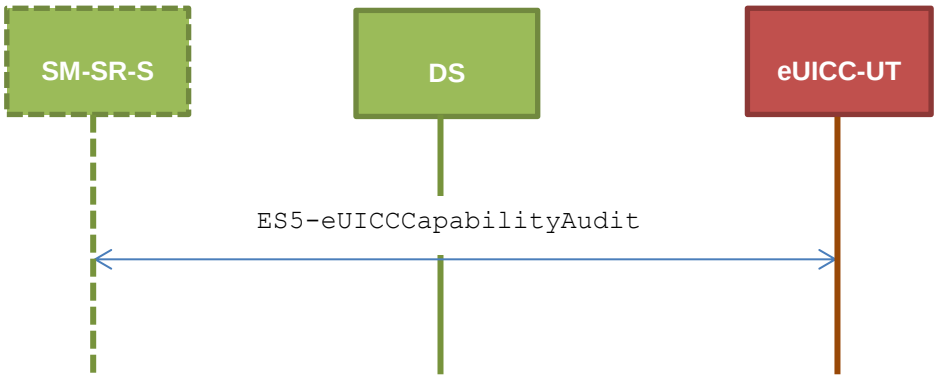
- PF_REQ7
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

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] (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] (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 #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 '42' is equal to #EUM_OID 7- TAG '95' is equal to #KEY_USAGE 8- TAG '73' contains the TLV 'C0', 'C1', 'C2' and 'C9' 9- TAG 'C9' is equal to #EUM_SUBJECT_KEY_ID 10- 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			
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			
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 #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E3_ISDP_LIST3] (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 #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

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 #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E3_ISDP1_1F] (see Note 2)	PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ18
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 #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 #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

Step	Direction	Sequence / Description	Expected result	REQ
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 #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			
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
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP_LIST3] (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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

Step	Direction	Sequence / Description	Expected result	REQ
7	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_ISDP_DISABLED])		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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP1_1F] (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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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, PF_REQ8_1, PF_REQ8_2
- 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*`

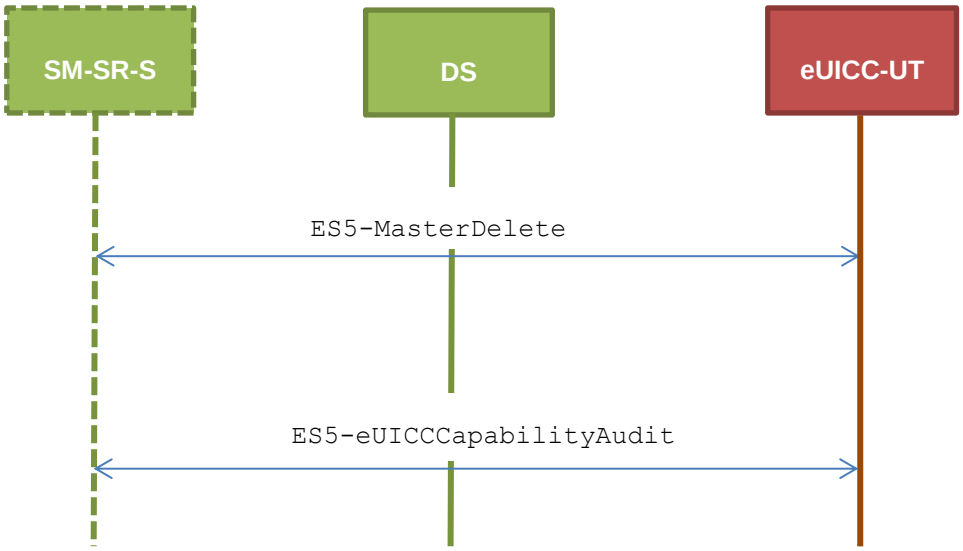
** To set the SDIN, SIN and the Application Provider Identifier, the sequence below SHALL be executed just after the establishment of the ISD-P keysets:*

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}`
				PF_REQ8_1

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'	PF_REQ8_1
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, PF_REQ8_1, PF_REQ8_2
- 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
- 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: With default Application Provider identifier (5F20)

Initial Conditions

- #ISD_P_AID1 in Disabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute
- No POL1 defined on #ISD_P_AID1
- #ISD_P_AID1 contains the SDIN value #ISD_P_SDIN*
- #ISD_P_AID1 contains the SDN value #ISD_P_SIN*
- #ISD_P_AID1 does not contain any Application Provider Identifier value *

** To set the SDIN and the SIN, the sequence below SHALL be executed just after the establishment of the ISD-P keysets (this overrides the related general initial condition defined in this section):*

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])) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		PF_REQ8_1 PF_REQ8_2
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'	PF_REQ8_1 PF_REQ8_2
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

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_RID])		EUICC_REQ22, EUICC_REQ54, PF_REQ8_2

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 #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_009000]	PF_REQ8, PF_REQ8_2, 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'	

4.2.9.2.1.3 Test Sequence N°3 – 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
- #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			

Step	Direction	Sequence / Description	Expected result	REQ
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'	

4.2.9.2.1.4 Test Sequence N°4 – 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

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, [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'	

4.2.9.2.1.5 Test Sequence N°5 – 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

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, [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'	

4.2.9.2.1.6 Test Sequence N°6 – 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

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] (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'				

4.2.9.2.1.7 Test Sequence N°7 – Error Case: With empty Application Provider identifier (5F20)

Initial Conditions

- #ISD_P_AID1 in Disabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute
- No POL1 defined on #ISD_P_AID1
- #ISD_P_AID1 contains the SDIN value #ISD_P_SDIN*
- #ISD_P_AID1 contains the SIN value #ISD_P_SIN*

- #ISD_P_AID1 does not contain any Application Provider Identifier value *

** To set the SDIN and the SIN, the sequence below SHALL be executed just after the establishment of the ISD-P keysets (this overrides the related general initial condition defined in this section):*

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])) Use the SCP03 keys {SCP_KENC}, {SCP_KMAC} and {SCP_KDEK}		PF_REQ8_1 PF_REQ8_2
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'	PF_REQ8_1 PF_REQ8_2
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

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_NO_PROV_ID])		EUICC_REQ22, EUICC_REQ54, PF_REQ8_2
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_6985] (see Note 1)	PF_REQ8, PF_REQ8_2, 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'				

4.2.9.2.1.8 Test Sequence N°8 – Error Case: With incorrect SDIN

Initial Conditions

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

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [MASTER_DEL_ISDP1_INV_SDIN])		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] (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'				

4.2.9.2.1.9 Test Sequence N°9 – Error Case: With incorrect SIN

Initial Conditions

- #ISD_P_AID1 in Disabled state
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute
- 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_INV_SIN])		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] (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'	

Step	Direction	Sequence / Description	Expected result	REQ
Note 1: The SW MAY be also '6A80' or '6982'				

4.2.9.2.1.10 Test Sequence N°10 – Error Case: With incorrect Application Provider ID

Initial Conditions

- `#ISD_P_AID1` in Disabled state
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute
- 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_RID])`
				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]` (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		

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_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, PF_REQ8_1
- 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

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, [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 #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 #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, PF_REQ8_1
- 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

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

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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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

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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

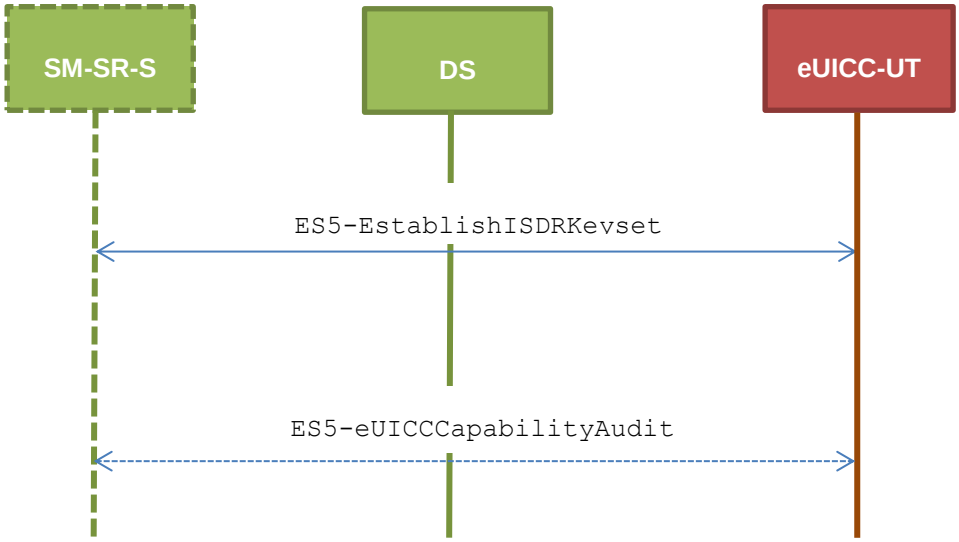
- PF_REQ7
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54
- PROC_REQ13_1

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

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 #SCP80_AUTH_KEY 3- The response data is equal to [R_AB_02RC] 4- Retrieve the {RC} 5- The {RC} length is either 16 or 32 bytes	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24, PROC_REQ13_1
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		
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'	

Step	Direction	Sequence / Description	Expected result	REQ
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED]) Use #SCP80_NEW_KVN, {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			
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		

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_02RC] 4- Retrieve the {RC} 5- The {RC} length is either 16 or 32 bytes	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24, PROC_REQ13_1
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		
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'	

Step	Direction	Sequence / Description	Expected result	REQ
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED]) Use #SCP80_NEW_KVN, {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			
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		

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_02RC] 4- Retrieve the {RC} 5- The {RC} length is either 16 or 32 bytes	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ24, PROC_REQ13_1
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		
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'	

Step	Direction	Sequence / Description	Expected result	REQ
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED]) Use #SCP80_NEW_KVN, {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

Step	Direction	Sequence / Description	Expected result	REQ
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			
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 `#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

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_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 #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
7	DS → eUICC-UT	ACK_DATA containing the result of SCP80_PACKET(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED]) Use #SCP80_NEW_KVN, {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			
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

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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

Step	Direction	Sequence / Description	Expected result	REQ
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 #SCP80_NEW_KVN, {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

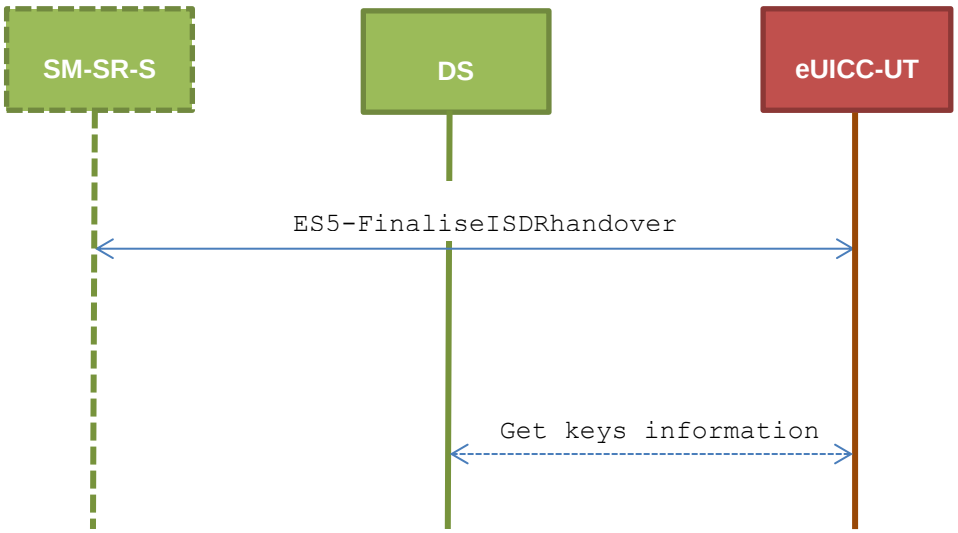
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

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

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 #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'	
12	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [PUTKEY_SCP81])		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_PUTKEY]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ25

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		

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_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			

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 `#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

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 #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_E0_SCP80] (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			

4.2.11.2.2.2 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			
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 #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

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 #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			

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([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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

Step	Direction	Sequence / Description	Expected result	REQ
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

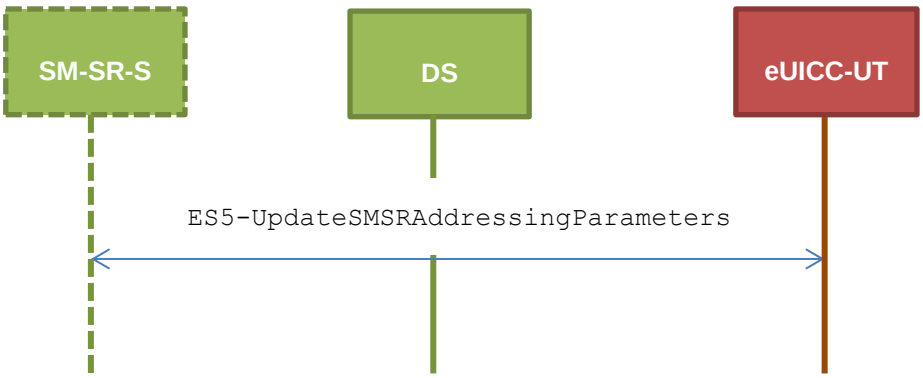
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26, EUICC_REQ26_1, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54

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, and that the eUICC deletes all previously stored information related to each concerned protocol subtag and just store the new set of parameters.

N.B.: Each of the subtags 'A3', 'A4', 'A5' is related to a different protocol, and can be updated without altering the configuration for the other protocols.

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_REQ26_1, EUICC_REQ54

Initial Conditions

- None

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

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (SHALL be the initial state of the eUICC)
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute
- `#ISD_P_AID1` in Disabled state
- `#ISD_P_AID1` has been personalized with the following SCP03 keys:
  o {SCP_KENC}, {SCP_KMAC}, {SCP_KDEK}
- No POL1 is defined on the `#DEFAULT_ISD_P_AID` and on the `#ISD_P_AID1`
- The SMS mode is the default way (priority order1) to send the notification
- TP-Destination-Address has been set on `#ISD_R_AID` with `#DEST_ADDR2`
- SMS-C parameters have been set on `#DEFAULT_ISD_P_AID` and `#ISD_P_AID1` with `#TON_NPI` and `#DIALING_NUMBER`
- For both `#DEFAULT_ISD_P_AID` and `#ISD_P_AID1`, TP-PID and TP-DCS are set to default values (no specific values have been set)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2		Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 10 in order to enable the `#ISD_P_AID1`	All steps successfully executed	
3		Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 2 to step 11 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 The TP-Destination-Address present in the eUICC Notification (SMS) is equal to `#DEST_ADDR2` (see step 5 of the test sequence defined in section 4.2.13.2.1.1) Check that TP-PID and TP-DCS are set to default value: • TP-PID = '00' • TP-DCS = '04'	

Step	Direction	Sequence / Description	Expected result	REQ
4	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [STORE_SMS_PARAM])		EUICC_REQ22, EUICC_REQ54
5	eUICC-UT → DS	*PROACTIVE COMMANDPENDING:* SEND SHORT MESSAGE		
6	DS → eUICC-UT	FETCH		
7	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
8	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
9	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_P_TAR1, SCP03_SCRIPT(#SCP03_KVN, [STORE_SMS_PARAM_MNO1]))		EUICC_REQ17, EUICC_REQ22, EUICC_REQ54
10	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
11	DS → eUICC-UT	FETCH		
12	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
13	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
14	Execute the test sequence defined in section 4.2.5.2.1.1 (TC.ES5.DISP.1:DisableProfile_SMS) from step 2 to step 10 in order to disable the #ISD_P_AID1		All steps executed successfully	

Step	Direction	Sequence / Description	Expected result	REQ
15		Execute the test sequence defined in section 4.2.14.2.1.1 (TC.ES5.NOTIFPD.1:Notification_SMS) from step 2 to step 11 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 Disabled	All steps successfully executed The TP-Destination-Address present in the eUICC Notification (SMS) is equal to #DEST_ADDR (see step 5 of the test sequence defined in section 4.2.14.2.1.1) Check that TP-PID and TP-DCS are set to default value: • TP-PID = '00' • TP-DCS = '04'	EUICC_REQ26
16		Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step2 to step 10 in order to enable the #ISD_P_AID1	All steps successfully executed	
17		Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 2 to step 11 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 The TP-Destination-Address present in the eUICC Notification (SMS) is equal to #DEST_ADDR (see step 5 of the test sequence defined in section 4.2.13.2.1.1) Check that TP-PID and TP-DCS are the values set in Step 9 : • TP-PID is set to #PID • TP-DCS is set to #DCS	

4.2.12.2.1.2 Test Sequence N°2 – Nominal Case: Update SMS Parameters with Profiles-Specific SM-SR Destination Addresses

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (SHALL be the initial state of the eUICC)
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute
- #ISD_P_AID1 in Disabled state
- No POL1 is defined on the #DEFAULT_ISD_P_AID and on the #ISD_P_AID1
- The SMS mode is the default way (priority order1) 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 and #ISD_P_AID1 with #TON_NPI and #DIALING_NUMBER

Step	Direction	Sequence / Description	Expected result	REQ
1		Initialization sequence as described in section 4.2.1.1		
Set a specific SM-SR destination address on both Profiles (#DEFAULT_ISD_P_AID and on the #ISD_P_AID1)				

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [STORE_SMS_PARAM_ISDPS])		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_9000]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26, EUICC_REQ26_1
6	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 10 in order to enable the #ISD_P_AID1		All steps successfully executed	
7	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 2 to step 11 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 The TP-Destination-Address present in the eUICC Notification (SMS) is equal to #DEST_ADDR3 (see step 5 of the test sequence defined in section 4.2.13.2.1.1)	EUICC_REQ26, EUICC_REQ26_1
8	Execute the test sequence defined in section 4.2.5.2.1.1 (TC.ES5.DISP.1:DisableProfile_SMS) from step 2 to step 10 in order to disable the #ISD_P_AID1		All steps executed successfully	
9	Execute the test sequence defined in section 4.2.14.2.1.1 (TC.ES5.NOTIFPD.1:Notification_SMS) from step 2 to step 11 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 Disabled		All steps successfully executed The TP-Destination-Address present in the eUICC Notification (SMS) is equal to #DEST_ADDR2 (see step 5 of the test sequence defined in section 4.2.14.2.1.1)	EUICC_REQ26, EUICC_REQ26_1
Set a specific SM-SR destination address only on the Default Profile (#DEFAULT_ISD_P_AID)				
10	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [STORE_SMS_PARAM_ISDP])		EUICC_REQ22, EUICC_REQ54

Step	Direction	Sequence / Description	Expected result	REQ
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_9000]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26, EUICC_REQ26_1
14	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 10 in order to enable the #ISD_P_AID1		All steps successfully executed	
15	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 2 to step 11 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 The TP-Destination-Address present in the eUICC Notification (SMS) is equal to #DEST_ADDR (see step 5 of the test sequence defined in section 4.2.13.2.1.1)	EUICC_REQ26, EUICC_REQ26_1

4.2.12.2.1.3 VOID

4.2.12.2.1.4 VOID

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			

Step	Direction	Sequence / Description	Expected result	REQ
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 #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

- #DEFAUT_ISD_P_AID in Enabled state (SHALL be the initial state of the eUICC)
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute
- #ISD_P_AID1 in Disabled state
- No POL1 is defined on the #DEFAULT_ISD_P_AID and on the #ISD_P_AID1

- The HTTP mode is the default way (priority order 1) to send the notification in both #DEFAULT_ISD_P_AID and #ISD_P_AID1
- HTTPS Connectivity Parameters have been set on #ISD_R_AID with #TCP_PORT, #IP_VALUE2, #ADMIN_HOST, #AGENT_ID, #PSK_ID, #SCP81_KVN, #SCP81_KEY_ID and #ADMIN_URI
- HTTPS Connectivity Parameters have been set on #ISD_P_AID1 and on the #DEFAULT_ISD_P_AID with #BEARER_DESCRIPTION, #NAN_VALUE, #LOGIN and #PWD

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	Execute the test sequence defined in section 4.2.4.2.3 (TC.ES5.EP.3:EnableProfile_HTTPS) from step 2 to step 9 in order to enable the #ISD_P_AID1		All steps successfully executed	
3	Execute the test sequence defined in section 4.2.13.2.3.1 (TC.ES5.NOTIFPE.3:Notification_HTTPS) from step 2 to step 14 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 The Data Destination-Address present in the OPEN CHANNEL is equal to #IP_VALUE2 (see step 5 of the test sequence defined in section 4.2.13.2.3.1)	
4	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, STORE_HTTPS_PARAM)		EUICC_REQ22, EUICC_REQ54
5	DS → eUICC-UT	FETCH		
6	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
7	Execute the test sequence defined in section 4.2.5.2.1.1 (TC.ES5.DISP.1:DisableProfile_SMS) from step 2 to step 10 in order to disable the #ISD_P_AID1		All steps successfully executed	
8	Execute the test sequence defined in section 4.2.13.2.3.1 (TC.ES5.NOTIFPE.3:Notification_HTTPS) from step 2 to step 14 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 Disabled		All steps successfully executed The Data Destination-Address present in the OPEN CHANNEL is equal to #IP_VALUE (see step 5 of the test sequence defined in section 4.2.13.2.3.1)	EUICC_REQ26
9	Close HTTPS session as described in section 4.2.1.7			

4.2.12.2.4 TC.ES5.USAP.4: UpdateSMSRAddrParam_DNS

Test Purpose

To ensure that the eUICC accepts the configuration of DNS parameters in the ISD-R by the SM-SR, and that the eUICC uses the DNS configuration appropriately.

Referenced Requirements

- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26, EUICC_REQ54, PROC_REQ23, PROC_REQ24, PROC_REQ25, PROC_REQ26, EUICC_REQ28, EUICC_REQ29, EUICC_REQ62, EUICC_REQ63,EUICC_REQ64, EUICC_REQ65, EUICC_REQ66

Initial Conditions

- `#DEFAULT_ISD_P_AID` in Enabled state (SHALL be the initial state of the eUICC)
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-back Attribute
- HTTPS Connectivity Parameters have been set on the `#DEFAULT_ISD_P_AID` with `#BEARER_DESCRIPTION, #NAN_VALUE, #LOGIN` and `#PWD`

4.2.12.2.4.1 Test Sequence N°1 – Nominal Case: Update DNS Parameters when OTA IP present in ISD-R

Test sequence Purpose

To ensure that the eUICC accepts the DNS configuration, but does not try to resolve the address of the SM-SR when the IP address is of the SM-SR statically know in the ISD-R configuration.

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`

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
Set a specific DNS configuration				

2	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_R_TAR,`
		`[STORE_DNS_PARAM])`
				EUICC_REQ22, EUICC_REQ64 EUICC_REQ65

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

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC → 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]	
6	DS → eUICC-UT	*TERMINAL RESPONSE*	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE_NO_POR, #ISD_R_TAR, [OPEN_SCP81_SESSION_WITH_NO_IP_ADDRESS])		
8	eUICC → DS	*PROACTIVE COMMAND PENDING:* OPEN_CHANNEL		
9	DS → eUICC-UT	FETCH		
10	eUICC → 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_REQ62
11	DS → eUICC-UT	TERMINAL RESPONSE		
12	Execute steps 10 to 14 of sub-sequence 4.2.1.5 to open the HTTPS session			
13	Execute sub-sequence 4.2.1.7 to close the HTTPS session			

4.2.12.2.4.2 Test Sequence N°2 – Nominal Case: Update DNS Parameters and no OTA IP present in ISD-R

Test sequence Purpose

To ensure that the eUICC accepts the DNS configuration, and uses it to start a DNS query to resolve the address of the SM-SR when the IP address of the SM-SR is not known.

The full DNS conversation is not tested and is FFS. Not completing the DNS resolution allows to avoid caching of the resolved address and will execute sequentially several DNS-related tests.

The eUICC may implement a retry mechanism, so the test sequence has to exhaust the number of retries to avoid impacting other tests.

Initial Conditions

- The ISD-R is configured with a TCP port but no IP address in the Connection Parameters of the Security Domain Administration Session Parameters (as defined by [STORE_HTTPS_PARAM_NO_IP_ADDRESS])

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
Set a specific DNS configuration				
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [STORE_DNS_PARAM])		EUICC_REQ22, EUICC_REQ64 EUICC_REQ65
3	eUICC → DS	PROACTIVE COMMAND PENDING: SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH		
5	eUICC → 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]	
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE_NO_POR, #ISD_R_TAR, [OPEN_SCP81_SESSION_WITH_NO_IP_ADDRESS])		
Loop in order to exhaust retries *(The maximum number of retries SHALL be given by the EUM to the Test Tool Provider).*				
8	eUICC → DS	*PROACTIVE COMMAND PENDING:* OPEN_CHANNEL (See note 1)		
9	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
10	eUICC → DS	PROACTIVE COMMAND: OPEN_CHANNEL	1- The UICC/terminal interface transport level field indicates UDP, and the port value #DNS_PORT specified in the DNS configuration at step 2 2- The Data destination address field contains the IP address #DNS_IP specified in the DNS configuration at step 2	EUICC_REQ62
11	*DS → eUICC*	TERMINAL RESPONSE with Result='21' (Network currently unable to process command)	SW='9000' Or SW indicate proactive command pending.	
End loop when at step 11 SW= '9000', or after the maximum number of retries is reached.				
Note 1: It is assumed that some proactive commands TIMER MANAGEMENT or POLL INTERVALL MAY be sent by the eUICC between iterations of the loop. The Device Simulator SHALL honor these commands as per section 3.2.1.1				

4.2.12.2.4.3 Test Sequence N°3 – Nominal Case: Update DNS Parameters when OTA IP present in the administration session triggering message

Test sequence Purpose

To ensure that the eUICC accepts the DNS configuration, but does not try to resolve the address of the SM-SR when the IP address of the SM-SR is provided in the administration session triggering message.

Initial Conditions

- The ISD-R is configured with a TCP port but no IP address in the Connection Parameters of the Security Domain Administration Session Parameters (as defined by [STORE_HTTPS_PARAM_NO_IP_ADDRESS])

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
Set a specific DNS configuration				
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [STORE_DNS_PARAM])		EUICC_REQ22, EUICC_REQ64, EUICC_REQ65
3	eUICC → DS	PROACTIVE COMMAND PENDING: SEND SHORT MESSAGE		

Step	Direction	Sequence / Description	Expected result	REQ
4	DS → eUICC-UT	FETCH		
5	eUICC → 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]	
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE_NO_POR, #ISD_R_TAR, [OPEN_SCP81_SESSION])		
8	eUICC → DS	PROACTIVE COMMAND PENDING: OPEN_CHANNEL		
9	DS → eUICC-UT	FETCH		
10	eUICC → 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_REQ62
11	Execute steps 10 to 14 of sub-sequence 4.2.1.5 to open the HTTPS session			
12	Execute sub-sequence 4.2.1.7 to close the HTTPS session			

4.2.12.2.4.4 VOID

4.2.12.2.4.5 Test Sequence N°5 – Error Case: Remove DNS Parameters with no OTA IP

Test sequence Purpose

To ensure that the eUICC erases the DNS configuration.

NOTE Since all cases where the IP address of the SM-SR is statically known do not lead to a DNS resolution, the only way to check that DNS configuration is erased is to verify that a DNS resolution is not started in a case where the IP address is not known.

Initial Conditions

- The ISD-R is configured with a TCP port but no IP address in the Connection Parameters of the Security Domain Administration Session Parameters (as defined by [STORE_HTTPS_PARAM_NO_IP_ADDRESS])
- The ISD-R is configured with DNS parameters (e.g. like after execution of Test Sequence N°2)

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
Set a specific DNS configuration				
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [STORE_DNS_PARAM_ERASE])		EUICC_REQ22, EUICC_REQ64 EUICC_REQ65
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_9000]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ26
6	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE_NO_POR, #ISD_R_TAR, [OPEN_SCP81_SESSION_WITH_NO_IP_ADDRESS])	1- No POR sent by the eUICC 2- Check that the eUICC does not send OPEN CHANNEL	EUICC_REQ22, EUICC_REQ42, EUICC_REQ54, EUICC_REQ21_1

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

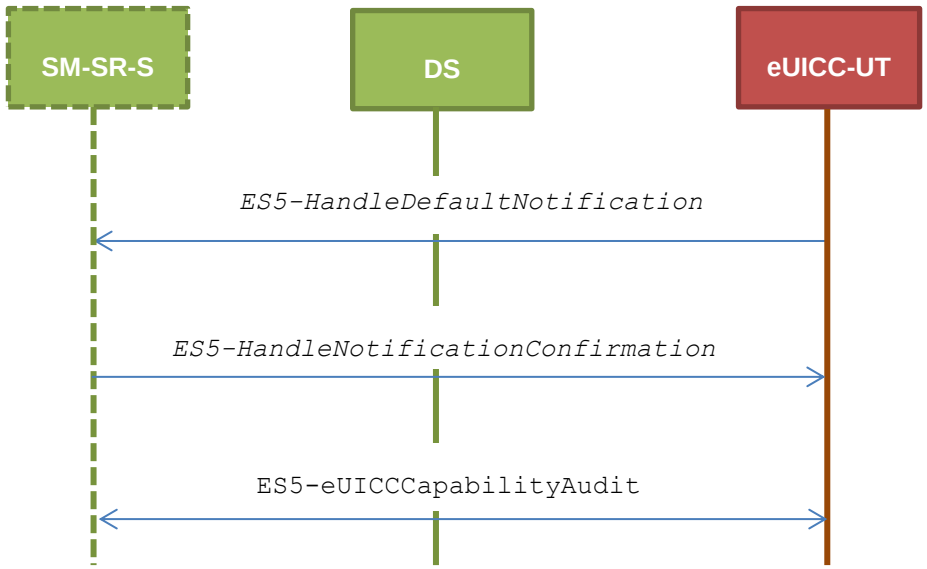
- PROC_REQ6, PROC_REQ8, PROC_REQ20, PROC_REQ2, PROC_REQ5_1
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ54

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



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, PROC_REQ5_1

- 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,
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		

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_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'	

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
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_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

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_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 – Nominal Case: No Follow-up Activities when the Profile is set with the Fall-Back Attribute and POL1 "Profile deletion is mandatory when its state is changed to disabled"

Initial Conditions

- POL1 "Profile deletion is mandatory when its state is changed to disabled" is 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		

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_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
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_NOTIF]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ20, PROC_REQ5_1
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		

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_ISDP_LIST1]	PM_REQ3, PM_REQ4, PF_REQ4, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, PROC_REQ5_1
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).

4.2.13.2.1.4 Test Sequence N°4 – 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		

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_CHANGE (see Note 1) 6- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ20
6	DS → eUICC-UT	TERMINAL RESPONSE		
Start loop while maximum retries number is not reached *(The maximum number of retries to wait for a Notification SHALL be given by the EUM to the Test Tool Provider)*				
7	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE See Note 4	This proactive command MAY be triggered by either an ENVELOPE(TIMER MANAGEMENT) or a STATUS command (maximum number of STATUS commands SHALL be given by the EUM to the Test Tool Provider)	EUICC_REQ27, PROC_REQ6, PROC_REQ8, PROC_REQ20
8	DS → eUICC-UT	FETCH		
9	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}: it SHALL be the same as the previous one	EUICC_REQ16, EUICC_REQ27, EUICC_REQ54, PROC_REQ6, PROC_REQ8, PROC_REQ20
10	DS → eUICC-UT	TERMINAL RESPONSE		
End loop				

Step	Direction	Sequence / Description	Expected result	REQ
11	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH See note 4	This proactive command MAY be triggered by either an ENVELOPE(TIMER MANAGEMENT) or a STATUS command (maximum number of STATUS commands SHALL be given by the EUM to the Test Tool Provider)	EUICC_REQ27, PROC_REQ6, PROC_REQ8, PROC_REQ20
12	DS → eUICC-UT	FETCH		
13	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PM_REQ3, PROC_REQ6, PROC_REQ8
14	DS → eUICC-UT	RESET	ATR returned by eUICC	
15	Initialization sequence as described in section 4.2.1.1			
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- 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` (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
19	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

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

21	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
22	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
23	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_NOTIF]	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ6, PROC_REQ8
24	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
25	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP_ENABLED])		EUICC_REQ54
26	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
27	DS → eUICC-UT	FETCH		
28	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]	PM_REQ3, PM_REQ4, PF_REQ7, PROC_REQ6, PROC_REQ8, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29
29	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).

Note 4: It is assumed that some proactive commands TIMER MANAGEMENT or POLL INTERVALL MAY be sent by the eUICC between iterations of the loop. The Device Simulator SHALL honor these commands as per section 3.2.1.1

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		
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.*				

Step	Direction	Sequence / Description	Expected result	REQ
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 (see Note 1) 5- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ18, EUICC_REQ27, EUICC_REQ54
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 #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])		EUICC_REQ22, EUICC_REQ54
15	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
16	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
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		
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 #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 Annex H.*				
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

Step	Direction	Sequence / Description	Expected result	REQ
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 3- The POST URI is equal to #POST_URI_NOTIF (see Note 1) 4- The headers are equal to #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R	EUICC_REQ14, EUICC_REQ27, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, PROC_REQ21
12	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([NOTIF_CONFIRMATION])		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 #HOST#X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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			

Step	Direction	Sequence / Description	Expected result	REQ
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_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'	

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
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ27, EUICC_REQ29, EUICC_REQ43,

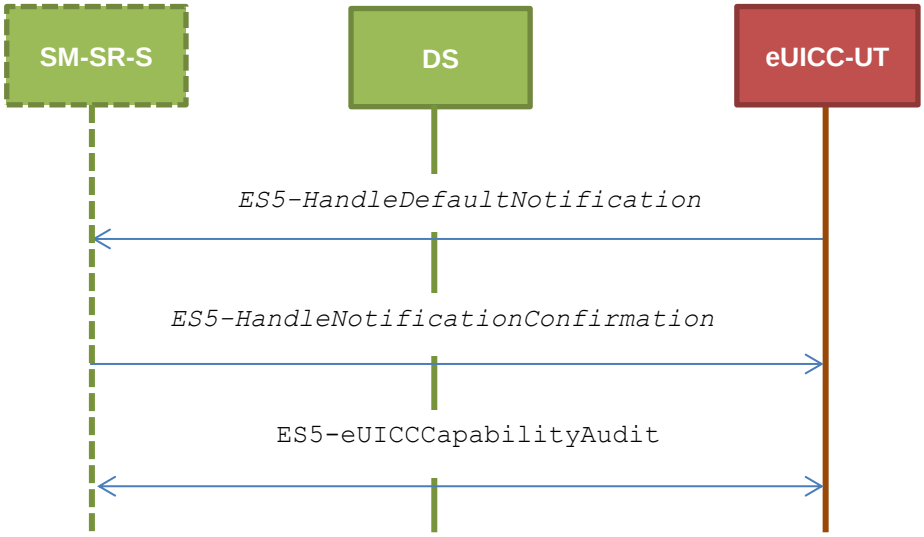
EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ50, EUICC_REQ54

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

Test Environment



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 SHALL be Enabled).

Referenced Requirements

- PF_REQ5, PF_REQ7
- PM_REQ3, PM_REQ4
- 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 `#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	
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_DEFAULT` (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

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_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_LIST2]	PM_REQ3, PM_REQ4, PF_REQ5, PF_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_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		
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_DEFAULT` (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

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_NOTIF2]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ20,
11	DS → eUICC-UT	TERMINAL RESPONSE	SW=’9000’	

Step	Direction	Sequence / Description	Expected result	REQ
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		
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_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).

4.2.14.2.2 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 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

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
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

Step	Direction	Sequence / Description	Expected result	REQ
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_DEFAULT (see Note 1) 5- Extract the {NOTIF_NUMBER}	EUICC_REQ16, EUICC_REQ18, EUICC_REQ27, EUICC_REQ54
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 #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])		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
18	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).*			

4.2.14.2.3 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 SHALL be Enabled).

Referenced Requirements

- PF_REQ5, 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`

4.2.14.2.3.1 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		
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 Annex H.*				
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

Step	Direction	Sequence / Description	Expected result	REQ
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
			3- The POST URI is equal to `#POST_URI_NOTIF_DEFAULT` (see Note 1)
			4- The headers are equal to
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
				EUICC_REQ14, EUICC_REQ27, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, PROC_REQ21

12	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT(`
		`[NOTIF_CONFIRMATION])`
				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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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])`
				EUICC_REQ22, EUICC_REQ54

16	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
17	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
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
19	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).

4.2.15 ES6 (MNO – eUICC): UpdatePOL1byMNO

4.2.15.1 Conformance Requirements

References

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

Requirements

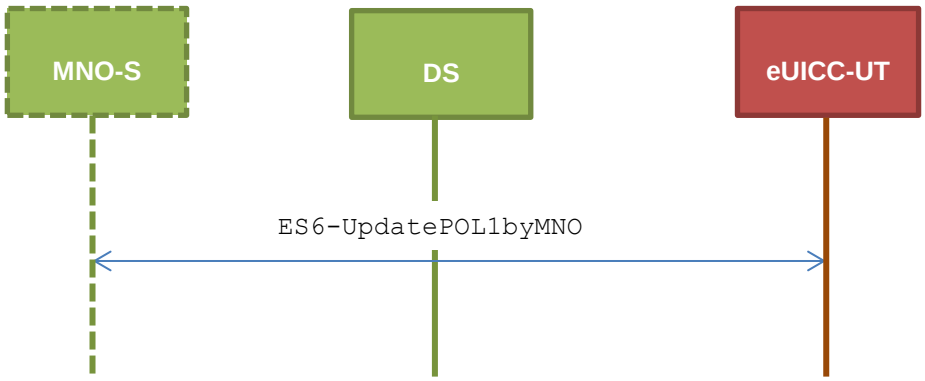
- PM_REQ6
- PROC_REQ17
- EUICC_REQ7, EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ43, EUICC_REQ48, EUICC_REQ52

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

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 #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		
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

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 #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			
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

Initial Conditions

- #DEFAULT_ISD_P_AID is in Enabled state (SHALL be the initial state of the eUICC)
- #ISD_P_AID1 in Disabled state
- For this test sequence, #MNO_TAR (MNO-SD TAR of the Profile linked to #DEFAULT_ISD_P_AID) is set to '010203' and SHALL not be equal to 'B20100'
- MNO-SD TAR of the Profile linked to the #ISD_P_AID1 is set to 'B20100' (as defined in section B.7.1)
- #DEFAULT_ISD_P_AID contains the POL1 "Disabling of the Profile not allowed"
- MNO-SD SCP80 keys of the Profile linked to the #ISD_P_AID1 are the same as the ones configured in the Profile #DEFAULT_ISD_P_AID (i.e. #MNO_SCP80_ENC_KEY, #MNO_SCP80_AUTH_KEY and #MNO_SCP80_DATA_ENC_KEY)
- 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 and #ISD_P_AID1 with #TON_NPI and #DIALING_NUMBER

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		
3	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
4	DS → eUICC-UT	FETCH	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]	
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
6	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 10 in order to enable the #ISD_P_AID1		All steps successfully executed	
7	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 2 to step 11 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	

Step	Direction	Sequence / Description	Expected result	REQ
8	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
9	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE	See Note 1	
10	DS → eUICC-UT	FETCH	The SCP80 status code is '09' – TAR unknown	PM_REQ6, PROC_REQ17, EUICC_REQ7, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22
11	DS → eUICC-UT	TERMINAL RESPONSE		
Note 1: 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 9, 10 and 11 SHALL be considered as optional.				

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			

Step	Direction	Sequence / Description	Expected result	REQ
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_MNO #CONTENT_TYPE #TRANSFER_ENCODING #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

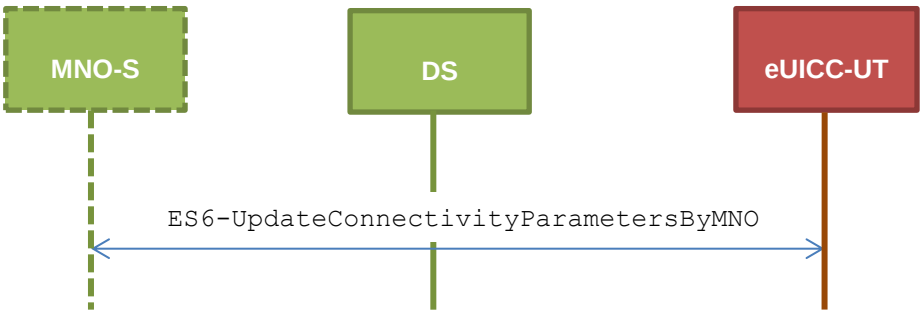
- PM_REQ7
- PROC_REQ18
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22

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, and configure the order of protocols used for the notitications.

Referenced Requirements

- PM_REQ7
- PROC_REQ18
- EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ28

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			
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'	

4.2.16.2.1.4 Test Sequence N°4 – Nominal Case: Update HTTPS + SMS Parameters

Initial Conditions

- #DEFAULT_ISD_P_AID in Enabled state (SHALL be the initial state of the eUICC)
- #DEFAULT_ISD_P_AID is the Profile with the Fall-back Attribute
- #ISD_P_AID1 present on the eUICC, in Disabled state
- No POL1 is defined on the #DEFAULT_ISD_P_AID and on the #ISD_P_AID1
- The SMS mode is the only way (priority order n°1, and no other protocol set) to send the notification on both ISD-P
- SMS-C parameters has been set on #ISD_P_AID1 with #TON_NPI and #DIALING_NUMBER
- SMS-C parameters has been set on #DEFAULT_ISD_P_AID with #TON_NPI and #DIALING_NUMBER_INITIAL
- TP-Destination-Address has been set on #ISD_R_AID with #DEST_ADDR
- 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

Specific conditions during execution of the test

The test sequence changes the Connectivity Parameters in the #DEFAULT_ISD_P_AID, and also verifies that the following notification sequence obeys the new Connectivity Parameters.
In order to trigger usage of both notification protocols, the DS SHALL be configured to reject HTTPS session opening, but allow SMS notification to succeed.

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
Update Connectivity Parameters via ES6				
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_PERSO_RES_ISDP]; [STORE_HTTPSSMS_PARAM]) 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'	
Enable #ISD_P_AID1				
7	Execute the test sequence defined in section 4.2.4.2.1.1 (TC.ES5.EP.1:EnableProfile_SMS) from step 2 to step 10 in order to enable the #ISD_P_AID1		All steps successfully executed	
8	Execute the test sequence defined in section 4.2.13.2.1.1 (TC.ES5.NOTIFPE.1:Notification_SMS) from step 2 to step 11 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	
Disable #ISD_P_AID1				

Step	Direction	Sequence / Description	Expected result	REQ
9	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [DISABLE_ISDP1])		EUICC_REQ22, EUICC_REQ54
10	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
11	DS → eUICC-UT	FETCH		
12	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]	PF_REQ5, EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22,
13	DS → eUICC-UT	TERMINAL RESPONSE		
14	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 1	
15	DS → eUICC-UT	FETCH		
16	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		PF_REQ5
17	DS → eUICC-UT	RESET	ATR returned by eUICC	
Handle notification sequence such that HTTP notification fails				
18	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
19	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL		
20	DS → eUICC-UT	FETCH		
21	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, EUICC_REQ28

Step	Direction	Sequence / Description	Expected result	REQ
22	DS → eUICC-UT	TERMINAL RESPONSE with Result field = '21' (Network currently unable to process command)	See Note 4	
Loop on steps 19 to 22 (see Note 4) while maximum retries number is not reached *(The maximum number of retries for HTTP session establishment SHALL be given by the EUM to the Test Tool Provider)*				
Handle notification in SMS sequence such that SMS notification succeeds				
23	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
24	DS → eUICC-UT	FETCH		
25	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, EUICC_REQ28
26	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

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

28	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
29	DS → eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
30	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_NOTIF]`	EUICC_REQ13, EUICC_REQ16, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ29, PROC_REQ20
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. *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 between the notifications (within a maximum time interval of 10 STATUS events).* *Note 4: It is assumed that some proactive commands TIMER MANAGEMENT or POLL INTERVALL MAY be sent by the eUICC between iterations of the loop. The Device Simulator SHALL honor these commands as per section 3.2.1.1*			

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

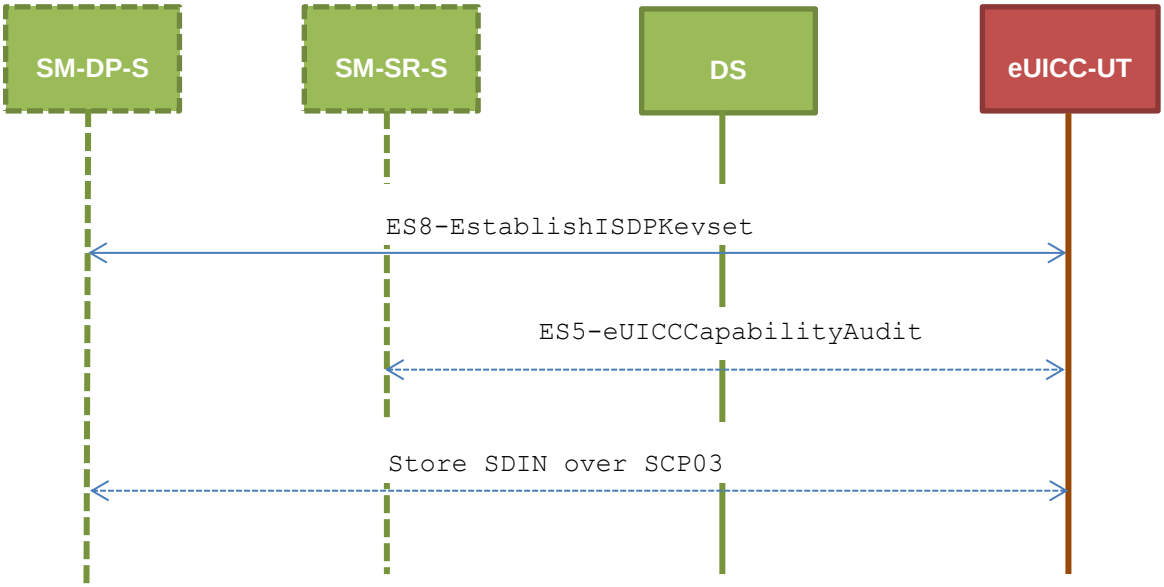
4.2.17.2 Test Cases

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} 5- The {RC} length is either 16 or 32 bytes	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'	

Step	Direction	Sequence / Description	Expected result	REQ
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'	

4.2.17.2.1.4 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		

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_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			

Step	Direction	Sequence / Description	Expected result	REQ
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 #SCP80_AUTH_KEY 4- The response data is equal to [R_AB_03RC] 5- Retrieve the {RC}	PM_REQ8, EUICC_REQ13, EUICC_REQ18
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 #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

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_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
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}		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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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

Step	Direction	Sequence / Description	Expected result	REQ
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
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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

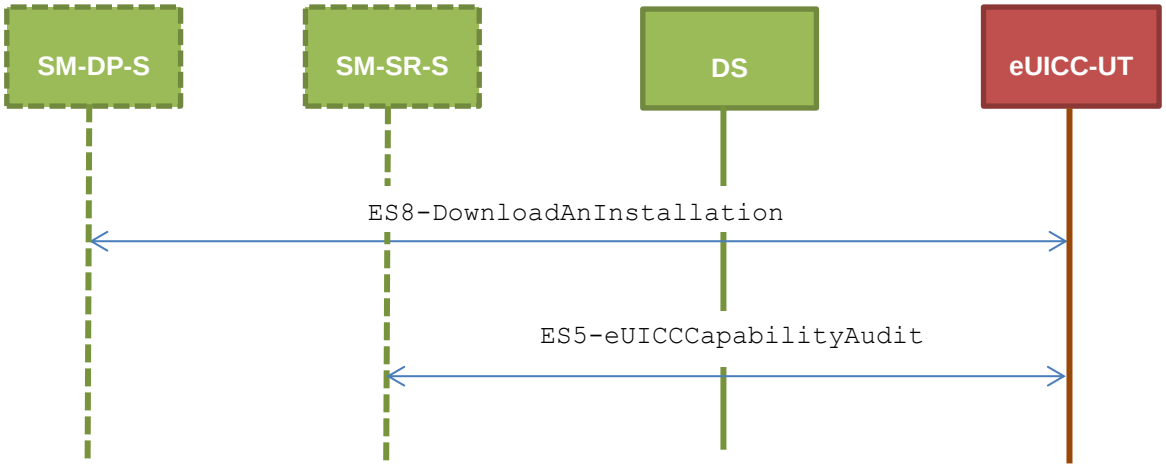
- PF_REQ7, PF_REQ4_1_3_3_1, PF_REQ4_1_3_3_2
- PM_REQ3, PM_REQ9
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ17, EUICC_REQ18, EUICC_REQ22, EUICC_REQ23, EUICC_REQ42, EUICC_REQ43, EUICC_REQ45, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ53, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61, EUICC_REQ4_1_3_3_1, EUICC_REQ4_1_3_3_2, EUICC_REQ4_1_3_3_3, EUICC_REQ4_1_3_3_4, EUICC_REQ4_1_3_3_5, EUICC_REQ4_1_3_3_6, EUICC_REQ4_1_3_3_7
- 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}` and `{SCP_KMAC}`
				EUICC_REQ17, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58

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- Verify the cryptographic checksum using `#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				

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

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 `#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

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 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			

Step	Direction	Sequence / Description	Expected result	REQ
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} and {SCP_KMAC}		EUICC_REQ17, EUICC_REQ54, EUICC_REQ57, EUICC_REQ58

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- Verify the cryptographic checksum using `#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)			

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. Contrary to the test case that uses CAT_TP (section 4.2.18.2.1), no script chaining has to be used over HTTP. 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, PF_REQ4_1_3_3_1, PF_REQ4_1_3_3_2
- 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, EUICC_REQ4_1_3_3_1, EUICC_REQ4_1_3_3_2, EUICC_REQ4_1_3_3_3, EUICC_REQ4_1_3_3_4, EUICC_REQ4_1_3_3_5, EUICC_REQ4_1_3_3_6, EUICC_REQ4_1_3_3_7
- 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			

Step	Direction	Sequence / Description	Expected result	REQ
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} and {SCP_KMAC}		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58, EUICC_REQ58_1, EUICC_REQ4_1_3_3_1, PF_REQ4_1_3_3_2
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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				

Step	Direction	Sequence / Description	Expected result	REQ
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, EUICC_REQ58_1
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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, EUICC_REQ58_1

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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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} and {SCP_KMAC}		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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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'	

4.2.18.2.2.3 Test Sequence N°3 – Nominal Case using random keys

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_INI_AUTH (#SCP03_KVN)) Use the SCP03 keys {SCP_KENC} and {SCP_KMAC}		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58, EUICC_REQ58_1, EUICC_REQ4_1_3_3_1,
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data formatted in an expanded remote	PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61

			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]	
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP (#ISD_P_AID1, SCP03T_REPLACE_SESSION_KEYS ()) Use the SCP03 keys {SCP_KENC} and {SCP_KMAC}	The response to the REPLACE_SESSION_KEYS command (i.e. TAG '87') SHALL be equal to [R_SCP03T_PROF_PROT_OK]	EUICC_REQ4_1_3_3_2, EUICC_REQ4_1_3_3_4, PF_REQ4_1_3_3_1, EUICC_REQ4_1_3_3_5
Loop until the Profile part index (named i) is equal to n-1				
6	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP (#ISD_P_AID1, SCP03T_SUB_SCRIPT ({PROFILE_PARTi})) Use the SCP03 keys #PPK-KENC, #PPK-MAC and #PPK-RMAC		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58, EUICC_REQ58_1, EUICC_REQ4_1_3_3_6
7	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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data formatted in an expanded remote	PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ61, SEC_REQ23

			command structure with indefinite length coding 5- For each SCP03t TLV command sent (i.e. TAG '86'), a response [R_SCP03T_EMPTY] is returned	
End loop				

8	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT_ISDP (`
		`#ISD_P_AID1,`
		`SCP03T_SUB_SCRIPT (`
		`{PROFILE_PARTn}))`
		
		Use the SCP03 keys `#PPK-ENC`, `#PPK-MAC` and `#PPK-RMAC`
				EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58, EUICC_REQ58_1

9	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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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 Random Session Key (#PPK-ENC) 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

10	Close HTTPS session as described in section 4.2.1.7			

11	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1])		EUICC_REQ22, EUICC_REQ54
12	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
13	DS → eUICC-UT	FETCH		
14	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
15	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

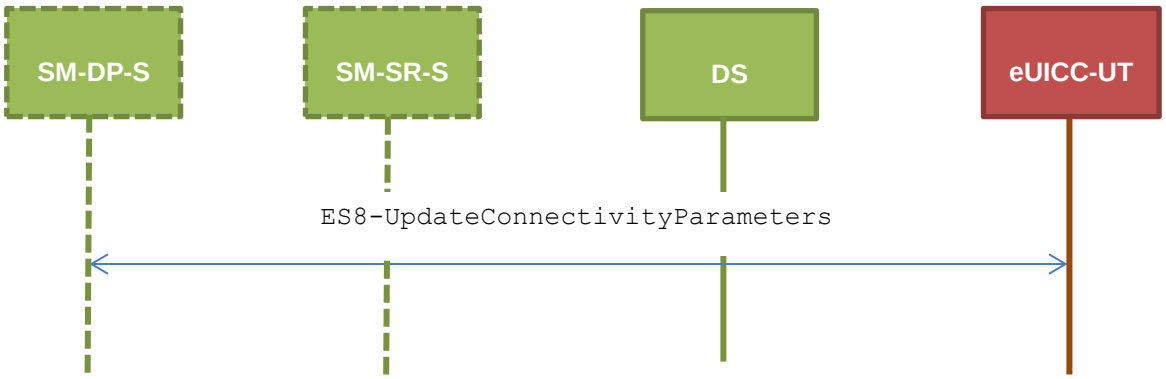
- EUICC_REQ13, EUICC_REQ14, EUICC_REQ16, EUICC_REQ17, EUICC_REQ18, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ23, EUICC_REQ31, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ54

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		

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'	

4.2.19.2.1.2 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
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			

Step	Direction	Sequence / Description	Expected result	REQ
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

Step	Direction	Sequence / Description	Expected result	REQ
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.19.2.1.5 Test Sequence N°5 – Nominal Case: Update HTTPS and 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_HTTPSSMS_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'	

4.2.19.2.2 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
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			

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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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			

4.2.20 ES5 (SM-SR – eUICC): SetEmergencyProfileAttribute

4.2.20.1 Conformance Requirements

References

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

Requirements

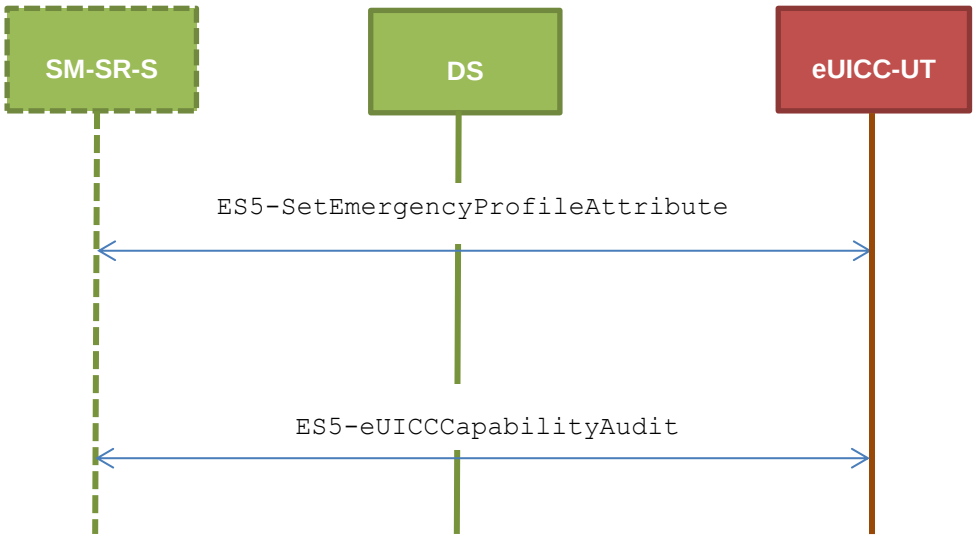
- EUICC_REQ29_1
- PF_REQ7

4.2.20.2 Test Cases

General Initial Conditions

- `#ISD_P_AID1` is present on the eUICC
- `#DEFAULT_ISD_P_AID` is present on the eUICC
- No Profile has the Emergency Profile Attribute set

Test Environment



4.2.20.2.1 TC.ES5.SEP.1: SetEmergencyProfileAttribute_SMS

Test Purpose

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

Referenced Requirements

- EUICC_REQ29_1
- PF_REQ7

Initial Conditions

- None

4.2.20.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- #DEFAULT_ISD_P_AID is Enabled
- #DEFAULT_ISD_P_AID is the Profile with the Fall-Back Attribute
- #ISD_P_AID1 is Disabled

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, [SET_EMERGENCY])		EUICC_REQ29_1
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_9000]	EUICC_REQ29_1
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_EMERGENCY])		PF_REQ7
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_EM]	EUICC_REQ29_1 PF_REQ7
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.20.2.1.2 Test Sequence N°2 – Error Case: The targeted Profile is Enabled

Initial Conditions

- #DEFAULT_ISD_P_AID is Disabled
- #DEFAULT_ISD_P_AID is the Profile with the Fall-Back Attribute
- #ISD_P_AID1 is Enabled

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, [SET_EMERGENCY])		EUICC_REQ29_1
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_REQ29_1
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_EMERGENCY])		PF_REQ7
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]	EUICC_REQ29_1 PF_REQ7
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.20.2.1.3 Test Sequence N°3 – Error Case: Targeted Profile has the Fall-Back Attribute

Initial Conditions

- #DEFAULT_ISD_P_AID is Enabled
- #ISD_P_AID1 is Disabled
- #ISD_P_AID1 is the Profile with the Fall-Back Attribute

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, [SET_EMERGENCY])		EUICC_REQ29_1
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_REQ29_1
6	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
7	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_EMERGENCY])		PF_REQ7
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]	EUICC_REQ29_1 PF_REQ7
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

4.2.20.2.2 TC.ES5.SEP.2: SetEmergencyProfileAttribute_HTTPS

Test Purpose

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

Referenced Requirements

- EUICC_REQ29_1
- PF_REQ7

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`

4.2.20.2.2.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- `#DEFAULT_ISD_P_AID` is Enabled
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-Back Attribute
- `#ISD_P_AID1` is Disabled

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_EMERGENCY])`
				EUICC_REQ29_1

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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_9000]	EUICC_REQ29_1
5	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT([GET_EMERGENCY])		PF_REQ7
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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #X_ADMIN_STATUS_OK 4- The HTTP content contains a response data equal to [R_AF_E3_ISDP1_EM]	EUICC_REQ29_1 PF_REQ7
7	Close HTTPS session as described in section 4.2.1.7			

4.2.21 ESX (SM-SR – eUICC): LocalEnableEmergencyProfile

4.2.21.1 Conformance Requirements

References

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

Requirements

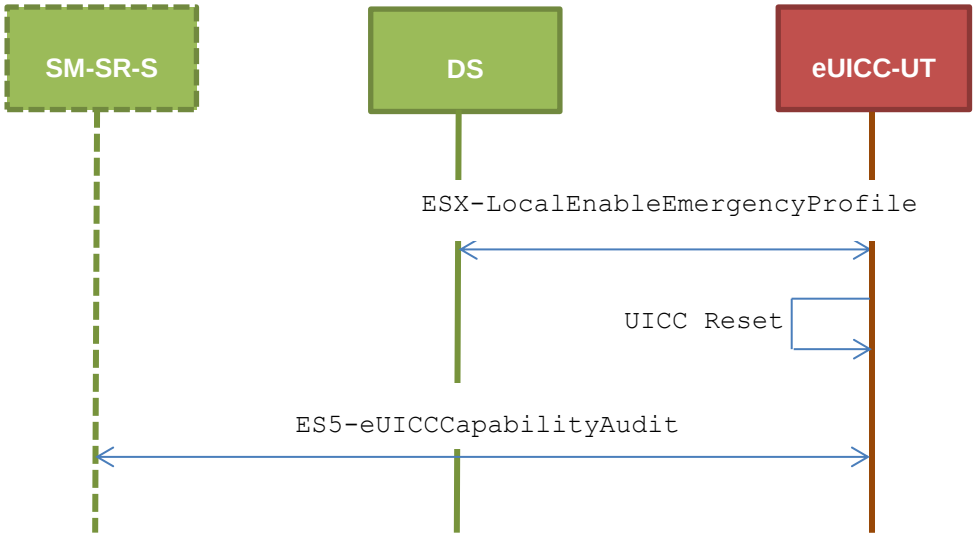
- EUICC_REQ31_1

4.2.21.2 Test Cases

General Initial Conditions

- #ISD_P_AID1 is present on the eUICC
- #DEFAULT_ISD_P_AID is present on the eUICC

Test Environment



4.2.21.2.1 TC.ESX.LEEP.1: LocalEnableEmergencyProfile

Test Purpose

To ensure it is possible to locally enable an Emergency Profile. After having enabled the Profile, a GET STATUS command is sent to make sure that the Profile state has changed.

Some error cases due to incompatible initial conditions are also defined.

Referenced Requirements

- EUICC_REQ31_1

Initial Conditions

- None

4.2.21.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- #DEFAULT_ISD_P_AID is Enabled
- #DEFAULT_ISD_P_AID is the Profile with the Fall-Back Attribute
- #ISD_P_AID1 is Disabled
- #ISD_P_AID1 is the Profile with the Emergency Profile Attribute

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	[ENVELOPE_LOCAL_ENABLE]		EUICC_REQ31_1
3	eUICC-UT → DS	ENVELOPE Response	The Response Data of the ENVELOPE response is equal to [R_E4E_SUCCESS]	
4	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 1	EUICC_REQ31_1
5	DS → eUICC-UT	FETCH		
6	eUICC-UT → DS	*PROACTIVE COMMAND:* REFRESH		
7	DS → eUICC-UT	RESET	ATR returned by eUICC	
8	DS → eUICC-UT	[TERMINAL_PROFILE]	1. Toolkit initialization (see Note 2) 2. Verify that no eUICC Notification is sent (i.e. no OPEN CHANNEL and no Envelope SMS is sent by the eUICC)	EUICC_REQ31_1
9	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])		
10	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
11	DS → eUICC-UT	FETCH		
12	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]	EUICC_REQ31_1
13	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

Step	Direction	Sequence / Description	Expected result	REQ
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. *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.*				

4.2.21.2.1.2 Test Sequence N°2 – Error Case: Emergency Profile already Enabled

Initial Conditions

- `#DEFAULT_ISD_P_AID` is Enabled
- `#DEFAULT_ISD_P_AID` is the Profile with the Fall-Back Attribute
- `#ISD_P_AID1` is Disabled
- `#ISD_P_AID1` is the Profile with the Emergency Profile Attribute

Step	Direction	Sequence / Description	Expected result	REQ
1	Perform test sequence N°3 specified in section 4.2.21.2.1.1			
2	DS → eUICC-UT	`[ENVELOPE_LOCAL_ENABLE]`		EUICC_REQ31_1
3	eUICC-UT → DS	`ENVELOPE Response`	The Response Data of the ENVELOPE response is equal to `[R_E4E_ALREADY ENABLED]` No proactive command is pending	

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

5	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
6	DS → eUICC-UT	FETCH		
7	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` The response data is equal to `[R_AB_E3_ISDP_LIST1]`	EUICC_REQ31_1
8	DS → eUICC-UT	*TERMINAL RESPONSE*	SW='9000'	

4.2.21.2.1.3 Test Sequence N°3 – Error Case: no Emergency Profile

Initial Conditions

- #DEFAULT_ISD_P_AID is Enabled
- #DEFAULT_ISD_P_AID is the Profile with the Fall-Back Attribute
- #ISD_P_AID1 is Disabled
- No Profile has its Emergency Profile 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_LOCAL_ENABLE]		EUICC_REQ31_1
3	eUICC-UT → DS	ENVELOPE Response	The Response Data of the ENVELOPE response is equal to [R_E4E_PROFILE_REF] No proactive command is pending	
4	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])		
5	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
6	DS → eUICC-UT	FETCH		
7	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 The response data is equal to [R_AB_E3_ISDP_LIST2]	EUICC_REQ31_1
8	DS → eUICC-UT	*TERMINAL RESPONSE*	SW='9000'	

4.2.22 ESX (SM-SR – eUICC): LocalDisableEmergencyProfile

4.2.22.1 Conformance Requirements

References

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

Requirements

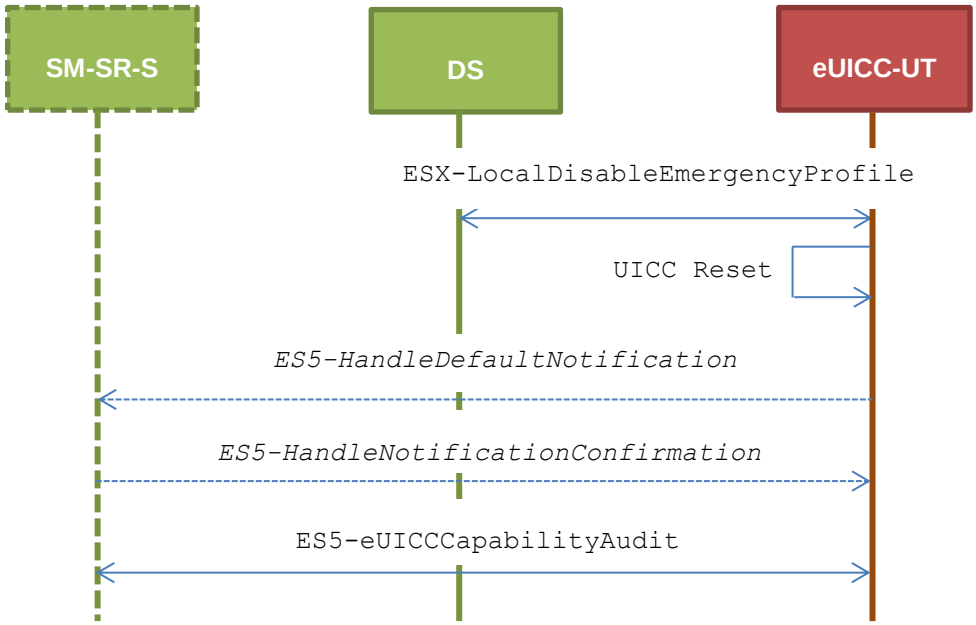
- EUICC_REQ31_2

4.2.22.2 Test Cases

General Initial Conditions

- `#ISD_P_AID1` is present on the eUICC
- `#DEFAULT_ISD_P_AID` is present on the eUICC

Test Environment



4.2.22.2.1 TC.ESX.LDEP.1: LocalDisableEmergencyProfile

Test Purpose

To ensure it is possible to locally disable an Emergency Profile. After having disabled the Profile, the notification mechanism MAY be triggered by the eUICC. Finally, a GET STATUS command is sent to make sure that the Profile state has changed.

An error cases due to incompatible initial conditions is also defined.

Referenced Requirements

- EUICC_REQ31_2

Initial Conditions

- None

4.2.22.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- #DEFAULT_ISD_P_AID is Disabled
- #DEFAULT_ISD_P_AID is the Profile with the Fall-Back Attribute
- #ISD_P_AID1 has been Enabled using the envelope [ENVELOPE_LOCAL_ENABLE]
- The previously Enabled Profile was #DEFAULT_ISD_P_AID
- #ISD_P_AID1 is the Profile with the Emergency Profile Attribute
- 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

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS→eUICC-UT	[ENVELOPE_LOCAL_DISABLE]		EUICC_REQ31_2
3	eUICC-UT→DS	ENVELOPE Response	The Response Data of the ENVELOPE response is equal to [R_E4E_SUCCESS]	
4	eUICC-UT→DS	*PROACTIVE COMMAND PENDING:* REFRESH	see Note 1	EUICC_REQ31_2
5	DS→eUICC-UT	FETCH		
6	eUICC-UT→DS	*PROACTIVE COMMAND:* REFRESH		
7	DS→eUICC-UT	RESET	ATR returned by eUICC	
8	DS→eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization see Note 2 and Note 3	
The steps 8 to 16 are optional. As the eUICC MAY or MAY NOT send notification to the SM-SR after an Emergency Profile Local Disabling, the next 9 steps are only applicable for eUICCs managing notifications for 'Profile change after Emergency Profile disabling'.				
9	eUICC-UT→DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
10	DS→eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
11	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_EMERGENCY (see Note 4) 6- Extract the {NOTIF_NUMBER}	EUICC_REQ31_2
12	DS→eUICC-UT	TERMINAL RESPONSE	SW='9000'	
13	DS→eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [NOTIF_CONFIRMATION])		
14	eUICC-UT→DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
15	DS→eUICC-UT	FETCH		
16	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_NOTIF]	
17	DS→eUICC-UT	TERMINAL RESPONSE	SW='9000'	
End of the optional steps				
18	DS→eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])		
19	eUICC-UT→DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
20	DS→eUICC-UT	FETCH		

Step	Direction	Sequence / Description	Expected result	REQ
21	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]	EUICC_REQ31_2
22	DS→eUICC-UT	TERMINAL RESPONSE	SW='9000'	
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. *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).* *Note 4: 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.*				

4.2.22.2.1.2 Test Sequence N°2 – Error case: Emergency Profile already Disabled

Initial Conditions

- #DEFAULT_ISD_P_AID is Enabled
- #DEFAULT_ISD_P_AID is the Profile with the Fall-Back Attribute
- #ISD_P_AID1 is Disabled
- #ISD_P_AID1 is the Profile with the Emergency Profile Attribute

Step	Direction	Sequence / Description	Expected result	REQ
1	Initialization sequence as described in section 4.2.1.1			
2	DS → eUICC-UT	[ENVELOPE_LOCAL_DISABLE]		EUICC_REQ31_1
3	eUICC-UT → DS	ENVELOPE Response	The Response Data of the ENVELOPE response is equal to [R_E4E_ALREADY DISABLED] No proactive command is pending	

Step	Direction	Sequence / Description	Expected result	REQ
4	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_R_TAR, [GET_ISDP1]; [GET_DEFAULT_ISDP])		
5	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		
6	DS → eUICC-UT	FETCH		
7	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 The response data is equal to [R_AB_E3_ISDP_LIST2]	EUICC_REQ31_1
8	DS → eUICC-UT	*TERMINAL RESPONSE*	SW='9000'	

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
- No PLMA is set in the SM-SR-UT on any Profile type

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`
- The variable `{SM_DP_ID_RPS}` in the `ProfileInfo`:
- SHALL be set to `#SM_DP_S_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)`		

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-SR-UT → EUM-S	Send the ES1-RegisterEIS response	The Status is equal to #SUCCESS	PROC_REQ14, EUICC_REQ32
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3- GetEIS, #VIRTUAL_EID_RPS, #MNO1_ID_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, with only profile #PROFILE1_RPS being present	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_EID 3- The Reason code is equal to #RC_OBJ_EXIST	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)		

Step	Direction	Sequence / Description	Expected result	REQ
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_VERIFICATION_FAILED	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	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

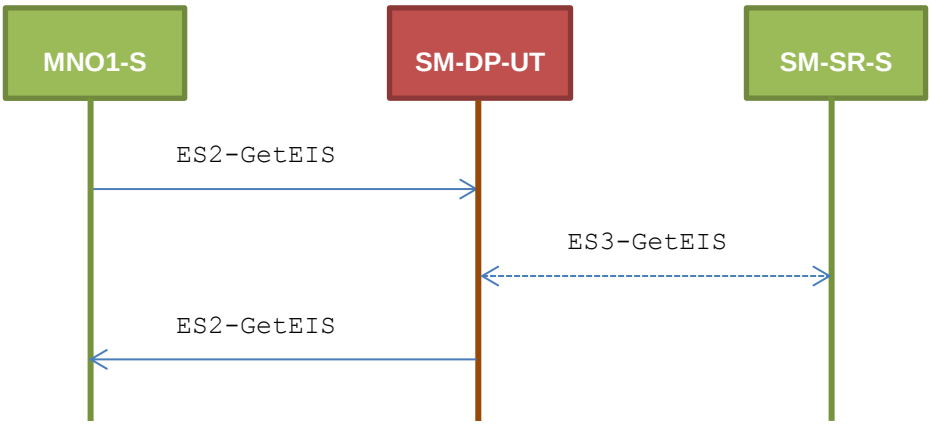
- PM_REQ10, PM_REQ14

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`
- The variable `{SM_DP_ID_RPS}` SHALL be set to `#SM_DP_UT_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	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The `MnoId` parameter is equal to `#MNO1_ID_RPS`	PM_REQ10, PM_REQ14

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-GetEIS, #EIS_ES3_RPS) Note: the SM-SR-S SHALL only include the profile #PROFILE1_RPS in this EIS		
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)		
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

Step	Direction	Sequence / Description	Expected result	REQ
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_ID_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

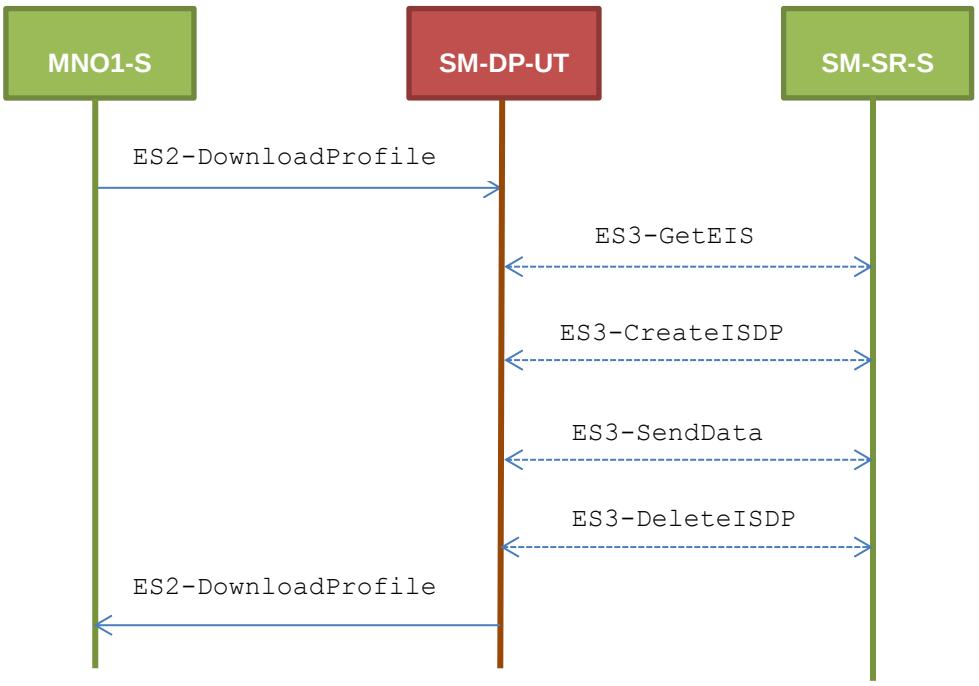
- PROC_REQ1, PROC_REQ2, PROC_REQ4
- PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17
- PF_REQ20

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. DOWNP.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. Four 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)*
- *send a delete profile before the profile is downloaded.*

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`
- The variable `{SM_DP_ID_RPS}` SHALL be set to `#SM_DP_UT_ID_RPS`

4.3.3.2.1.1 Test Sequence N°1 – Error Case: Keys Establishment Fails

Initial Conditions

- The Profile `#PF_PROFILE_TYPE_TO_DOWNLOAD` is well known to the SM-DP-UT and linked to a single `#PF_ICCID_TO_DOWNLOAD`
- 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},`
		`#PF_PROFILE_TYPE_TO_DOWNLOAD_RPS,`
		`#EP_FALSE_RPS)`		

2	SM-DP-UT → SM-SR-S	Send the `ES3-GetEIS` request	1- The `EID` parameter is equal to `#VIRTUAL_EID_RPS` 2- The `MnoId` parameter is equal to `#MNO1_ID_RPS`	PROC_REQ1, PM_REQ11, PM_REQ14

3	SM-SR-S → SM-DP-UT	`SEND_SUCCESS_RESP(`
		`ES3-GetEIS,`
		`#EIS_ES3_RPS)`
		Note: the SM-SR-S SHALL only include the profile `#PROFILE1_RPS` in this EIS		

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 `#PF_ICCID_TO_DOWNLOAD_RPS` 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 #PF_ICCID_TO_DOWNLOAD_RPS	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 euiccResponseData is equal to #EUICC_RESP1_RPS	PROC_REQ4, PM_REQ11

4.3.3.2.1.2 Test Sequence N°2 – Error Case: ISDP Creation Fails

Initial Conditions

- The Profile #PF_PROFILE_TYPE_TO_DOWNLOAD is well known to the SM-DP-UT and linked to a single #PF_ICCID_TO_DOWNLOAD
- 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}, #PF_ICCID_TO_DOWNLOAD_RPS, #EP_FALSE_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-GetEIS request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The MnoId parameter is equal to #MNO1_ID_RPS	PROC_REQ1, PM_REQ11, PM_REQ14
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-GetEIS, #EIS_ES3_RPS) Note: the SM-SR-S SHALL only include the profile #PROFILE1_RPS in this EIS		
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 #PF_ICCID_TO_DOWNLOAD_RPS 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	The Status is equal to #FAILED	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

4.3.3.2.1.4 Test sequence Nº 4 – Error Case: Download a Profile (only two first STORE DATA)

Initial Conditions

- The Profile #PF_PROFILE_TYPE_TO_DOWNLOAD is well known to the SM-DP-UT and is linked to a single #PF_ICCID_TO_DOWNLOAD

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-DownloadProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #PF_PROFILE_TYPE_TO_DOWNLOAD_RPS, #EP_FALSE_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-GetEIS request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 3- The MnoId parameter is equal to #MNO1_ID_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_ES3_RPS) Note: the SM-SR-S SHALL only include the profile #PROFILE1_RPS in this EIS		
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 #PF_ICCID_TO_DOWNLOAD_RPS 3- The MNO-ID parameter is equal to #MNO1_S_ID	PROC_REQ1, PM_REQ11, PM_REQ16
5	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-CreateISDP, #ISD_P_AID1)		
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 (INSTALL FOR PERSO where the Application AID equals #ISD_P_AID1, and First STORE DATA with DGI 3A01)	PROC_REQ2, PM_REQ11, PM_REQ17
7	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-SendData, {RC}) The {RC} is randomly generated (16 bytes long)		

Step	Direction	Sequence / Description	Expected result	REQ
8	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 APDU related to the ES8.EstablishISDPKeyset function (Second STORE DATA with DGI 3A02)	
9	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-SendData, {RECEIPT}) The {RECEIPT} is randomly generated (16 bytes long)		
10	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 #PF_ICCID_TO_DOWNLOAD_RPS	
11	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-DeleteISDP)		
12	SM-DP-UT → MNO1-S	Send the ES2-DownloadProfile response	The Status is equal to #FAILED	

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

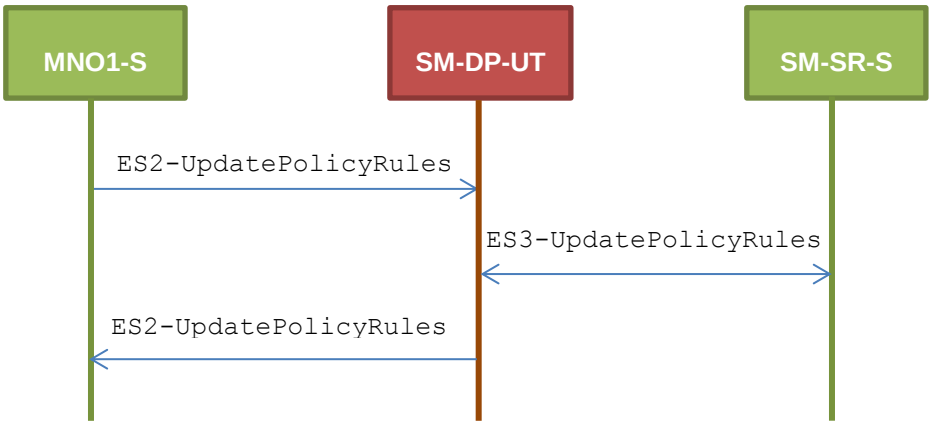
- PROC_REQ16
- PM_REQ12, PM_REQ19

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

Step	Direction	Sequence / Description	Expected result	REQ
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
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

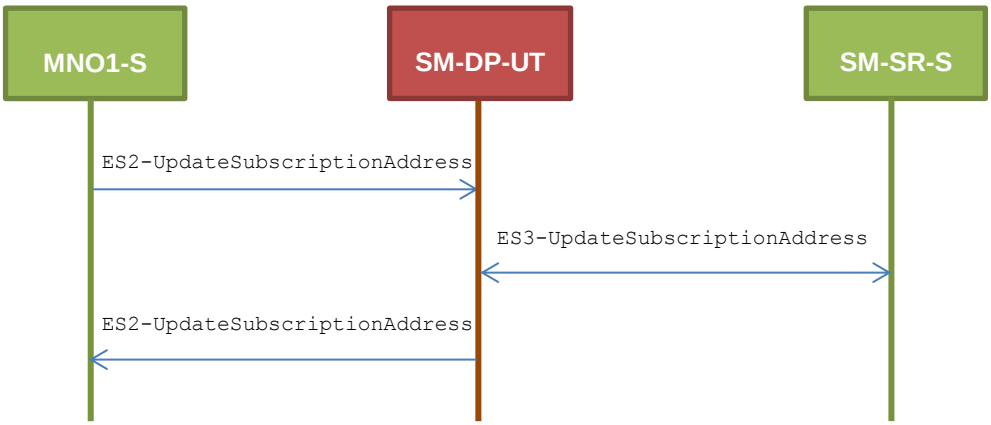
- PM_REQ13, PM_REQ20

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



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

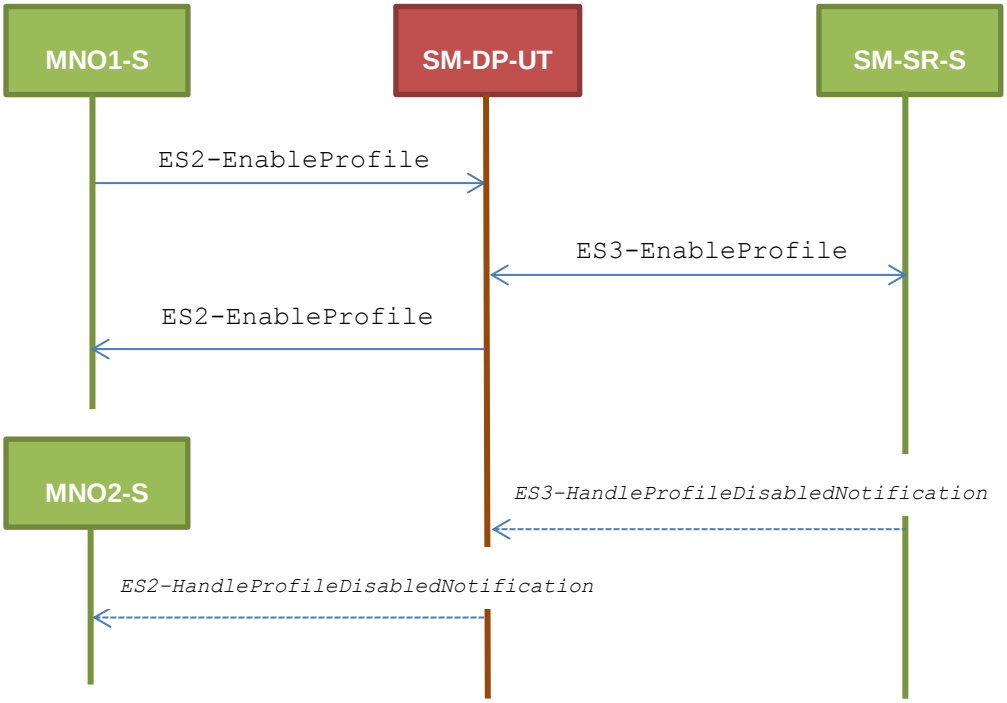
To ensure a Profile can be Enabled by the SM-DP through the SM-SR when a MNO requests it. After enabling the Profile, the SM-SR sends the notification

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*
- *the profile change procedure does not complete after enabling the target profile, and the profile change is rolled-back on the eUICC*

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)`		

Step	Direction	Sequence / Description	Expected result	REQ
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

Step	Direction	Sequence / Description	Expected result	REQ
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
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

4.3.6.2.1.4 Test Sequence N°4 – Error Case: Connectivity Failure and Roll-back Mechanism

Initial Conditions

- None

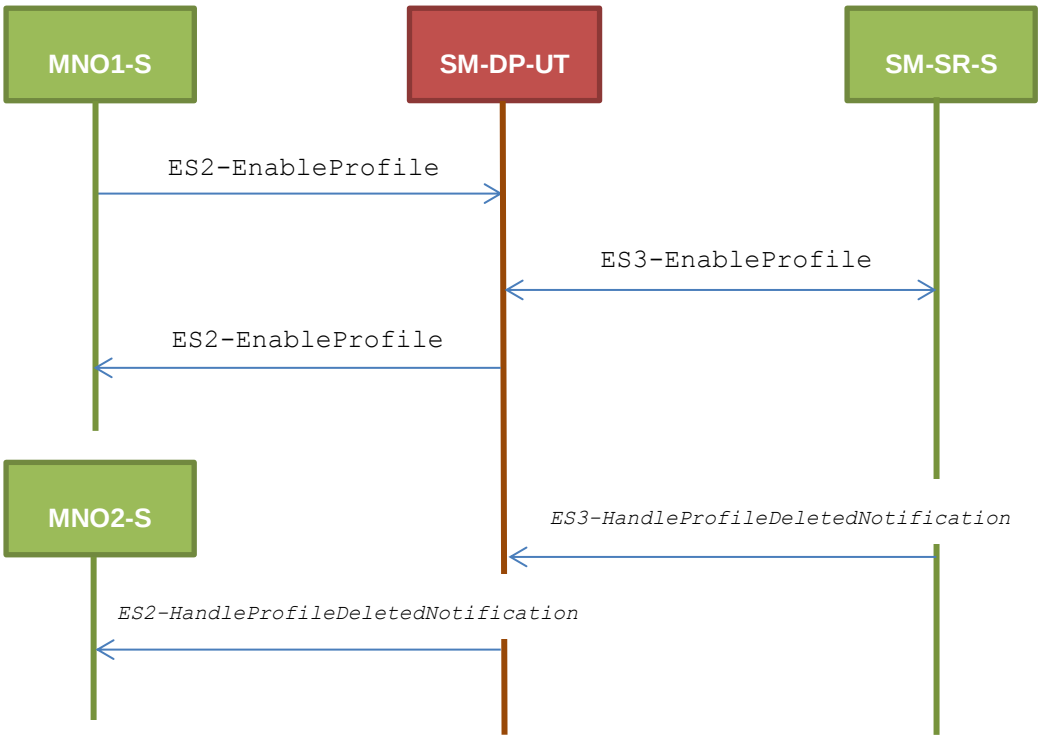
Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-EnableProfile, #VIRTUAL_EID_RPS, {SM_SR_ID_RPS}, #ICCID1_RPS)		
2	SM-DP-UT → SM-SR-S	Send the ES3-EnableProfile request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS	PROC_REQ7, PF_REQ12, PF_REQ18
3	SM-SR-S → SM-DP-UT	SEND_ERROR_RESP(ES3-EnableProfile, #FAILED, #SC_PROFILE, #RC_INACCESSIBLE)		
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 3- The Reason code is equal to #RC_INACCESSIBLE	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

Step	Direction	Sequence / Description	Expected result	REQ
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 DeletedNotification, #VIRTUAL_EID_RPS, #ICCID2_RPS, #MNO2_ID_RPS, #TIMESTAMP_RPS)		
6	SM-DP-UT → MNO2-S	Send the ES2-HandleProfile DeletedNotification 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

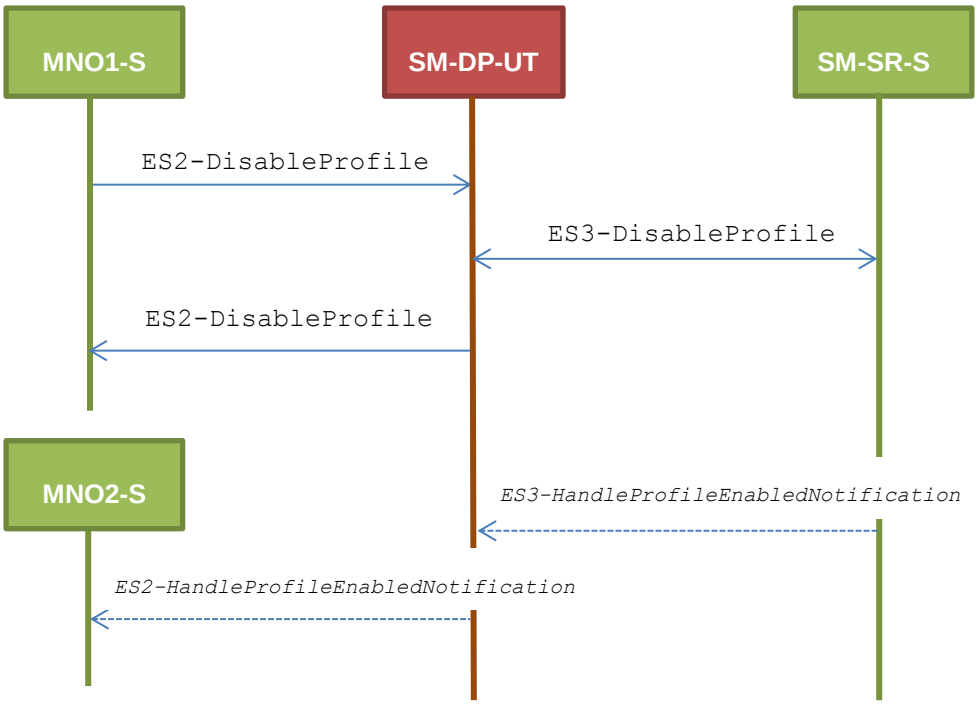
- PROC_REQ10
- PF_REQ13, PF_REQ16, PF_REQ19, PF_REQ22

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*
- *the profile change procedure does not complete after disabling the target profile, and the profile change is rolled-back on the eUICC*

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)		

Step	Direction	Sequence / Description	Expected result	REQ
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

3	SM-SR-S → SM-DP-UT	`SEND_ERROR_RESP(`
		`ES3-DisableProfile,`
		`#FAILED,`
		`#SC_POL1,`
		`#RC_REFUSED)`		

Step	Direction	Sequence / Description	Expected result	REQ
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

4.3.7.2.1.4 Test Sequence N°4 – Nominal Case: POL2 with "Profile Deletion is Mandatory when it is Disabled"

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, #WARNING, #SC_POL2, #RC_OBJ_EXIST)		
4	SM-DP-UT → MNO1-S	Send the ES2-DisableProfile response	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_POL2 3- The Reason code is equal to #RC_OBJ_EXIST	PROC_REQ10, PF_REQ13

4.3.7.2.1.5 Test Sequence N°5 – Error Case: Connectivity Failure and Roll-back Mechanism

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_PROFILE, #RC_INACCESSIBLE)		
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_PROFILE 3- The Reason code is equal to #RC_INACCESSIBLE	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

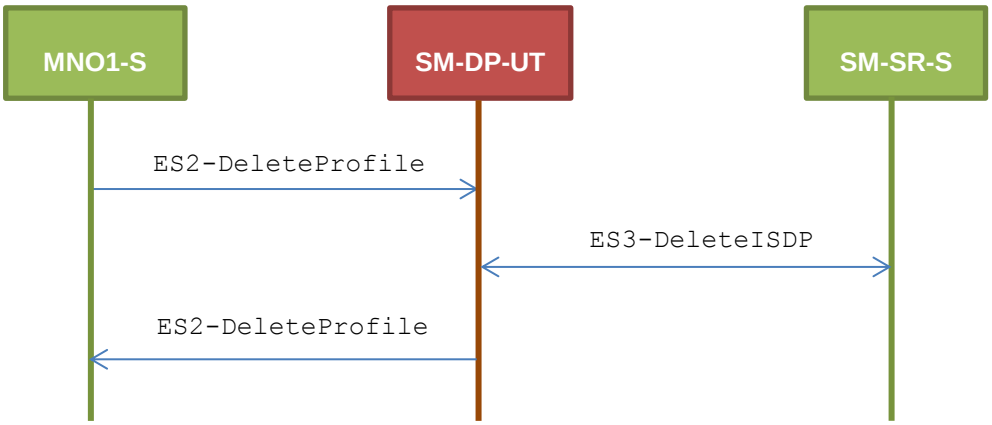
- PROC_REQ12
- PF_REQ14, PF_REQ20

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)*
- *the Profile identified by its ICCID is unknown from the SM-SR*

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

Step	Direction	Sequence / Description	Expected result	REQ
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
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

4.3.8.2.1.4 Test Sequence N°4 – Error Case: ISD-P identified by its AID does not exist on the targeted eUICC

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, #WARNING, #SC_ISDP, #RC_NOT_PRESENT)		
4	SM-DP-UT → MNO1-S	Send the ES2-DeleteProfile response	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_ISDP 3- The Reason code is equal to #RC_NOT_PRESENT	PROC_REQ12, PF_REQ14

4.3.8.2.1.5 Test Sequence N°5 – Error Case: Profile not present in the EIS

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}, #ICCID_UNKNOWN_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 #ICCID_UNKNOWN_RPS	PROC_REQ12, PF_REQ14, PF_REQ20
3	SM-SR-S → SM-DP-UT	SEND_ERROR_RESP(ES3-DeleteISDP, #FAILED, #SC_PROFILE_ICCID, #RC_UNKNOWN)		
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_PROFILE_ICCID 3- The Reason code is equal to #RC_UNKNOWN	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

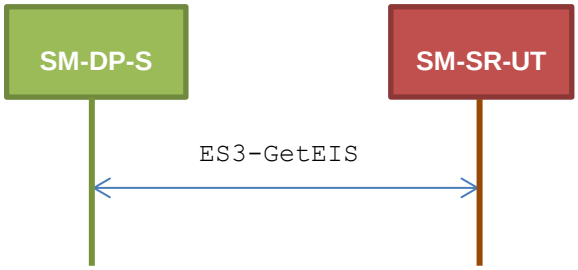
- PM_REQ14

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

- None

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`
  - `{SM_SR_ID_RPS}` has been set to `#SM_SR_UT_ID_RPS`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_ID_RPS`

- No PLMA is set in the SM-SR-UT on any Profile type

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #VIRTUAL_EID_RPS, #MNO1_ID_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, with only profile #PROFILE1_RPS being present	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, #MNO1_ID_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_ID_UNKNOWN	PM_REQ14

4.3.9.2.1.3 Test Sequence N°3 – Nominal Case with PLMA to see other profiles

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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_ID_RPS
- The PLMA #PLMA_MNO2_FOR_MNO1_RPS is granted by MNO2 to MNO1, to allow MNO1 to see the Profile (for example, by executing steps 1 to 3 of test sequence 4.4.4.2.1.1

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #VIRTUAL_EID_RPS, #MNO1_ID_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, with both profiles #PROFILE1_RPS and #PROFILE2_RPS being present	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

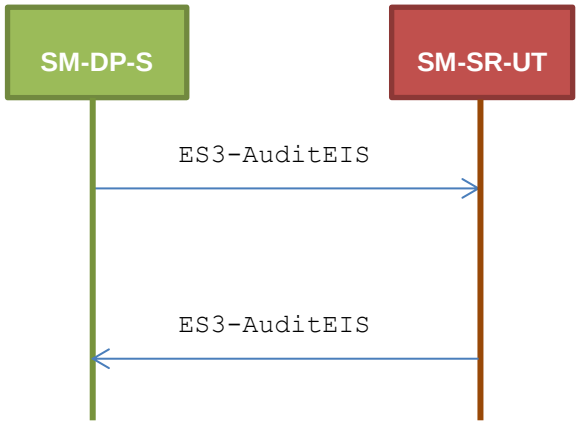
- PM_REQ15

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, #MNO1_ID_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

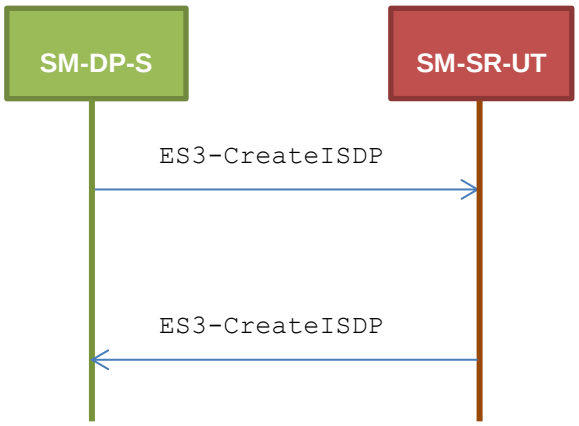
- PM_REQ16

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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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

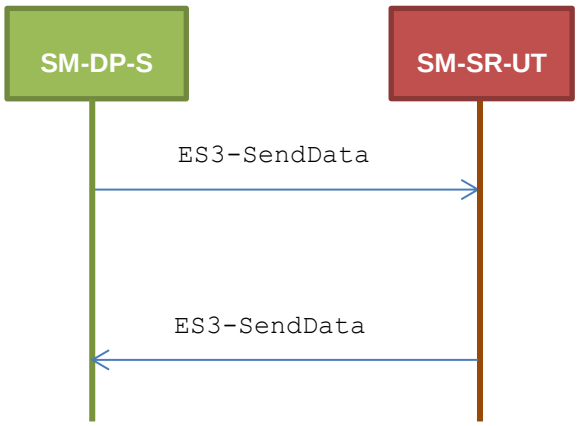
- PM_REQ17

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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_ID_RPS`

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SendData, #VIRTUAL_EID_RPS, #MNO1_ID_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

4.3.13 ES3 (SM-DP – SM-SR): UpdatePolicyRules

4.3.13.1 Conformance Requirements

References

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

Requirements

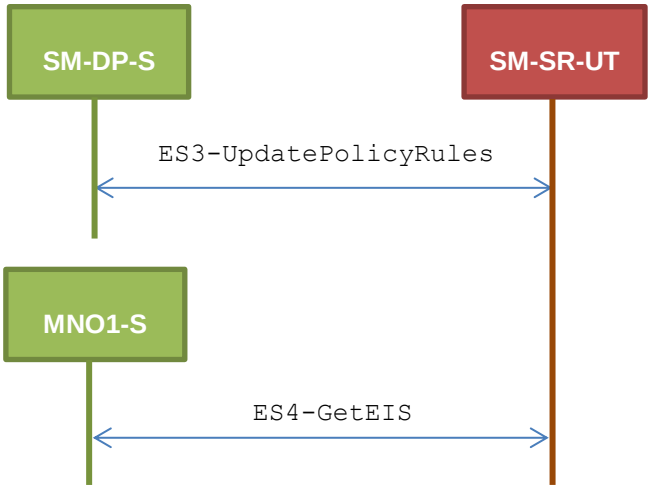
- PROC_REQ16
- PM_REQ19, PM_REQ22

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

- None

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)
  - `{SM_SR_ID_RPS}` has been set to `#SM_SR_UT_ID_RPS`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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, #MNO1_ID_RPS, #POL2_DIS_RPS)		

Step	Direction	Sequence / Description	Expected result	REQ
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

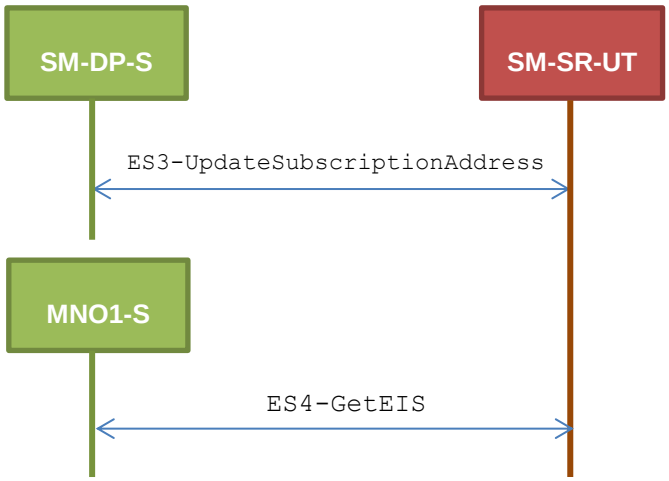
- PM_REQ20, PM_REQ22

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

- None

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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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, #MNO1_ID_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

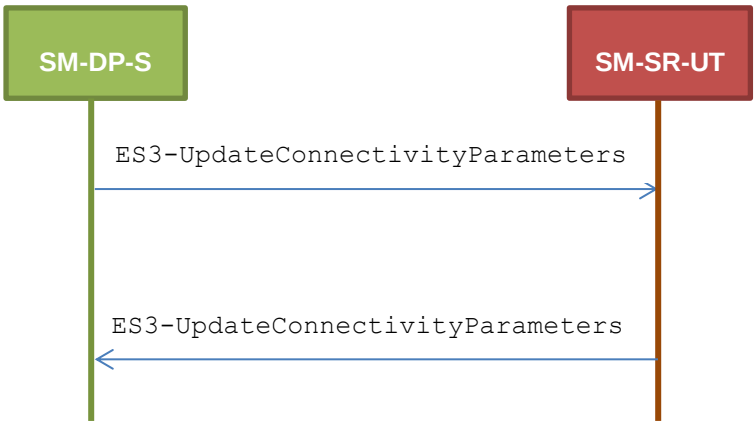
- PM_REQ21

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



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, #MNO1_ID_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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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, #MNO1_ID_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

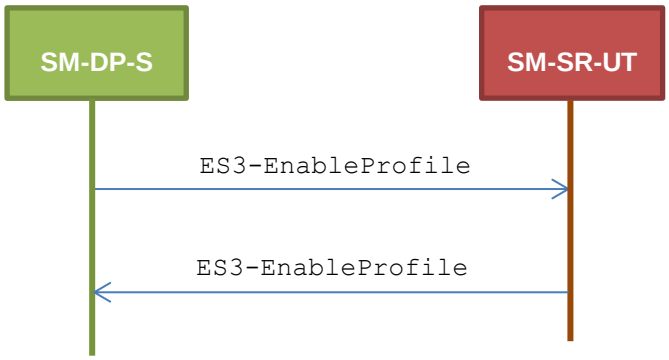
- PF_REQ18

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*
- *the SM-DP is acting on behalf on the MNO who owns the target Profile*

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,`
		`#MNO1_ID_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 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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS, #MNO1_ID_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, #MNO1_ID_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	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, #MNO1_ID_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

4.3.16.2.1.5 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 in Disabled state
- No PLMA has been configured

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS, #MNO2_ID_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_PLMA 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

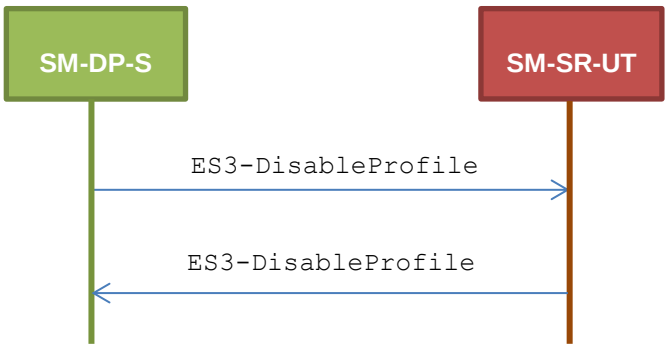
- PF_REQ19

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,`
		`#MNO1_ID_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 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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_ID_RPS`

Step	Direction	Sequence / Description	Expected result	REQ

1	SM-DP-S → SM-SR-UT	`SEND_REQ(`
		`ES3-DisableProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS,`
		`#MNO1_ID_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, #MNO1_ID_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	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, #MNO1_ID_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

4.3.17.2.1.5 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 in Enabled state
- No PLMA has been configured

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	`SEND_REQ( ES3-DisableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS, #MNO2_ID_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_PLMA` 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

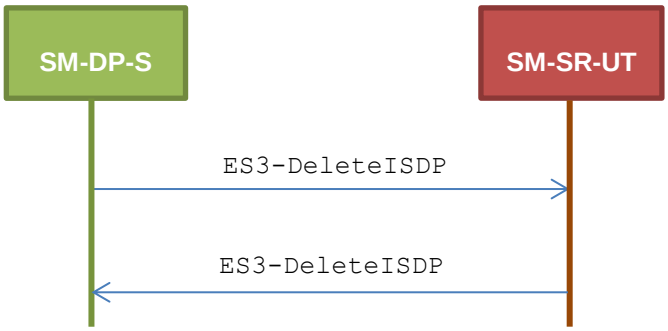
- PF_REQ20

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, #MNO1_ID_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 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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_ID_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DeleteISDP, #VIRTUAL_EID_RPS, #ICCID1_RPS, #MNO1_ID_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, #MNO1_ID_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, #MNO1_ID_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

4.3.18.2.1.5 Test Sequence N°5 – Error Case: Profile not present in the EIS

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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_ID_RPS
- The Profile identified by the #ICCID_UNKNOWN is unknown from 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, #ICCID_UNKNOWN_RPS, #MNO1_ID_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_UNKNOWN	PF_REQ20

4.3.18.2.1.6 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`
- 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,`
		`#MNO2_ID_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_PLMA` 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

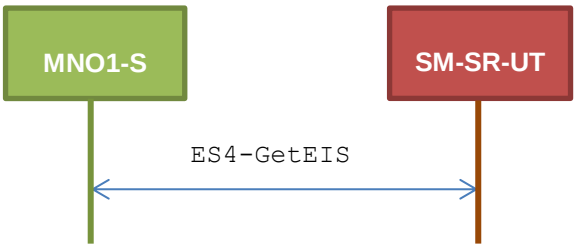
- PM_REQ22

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

- None

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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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

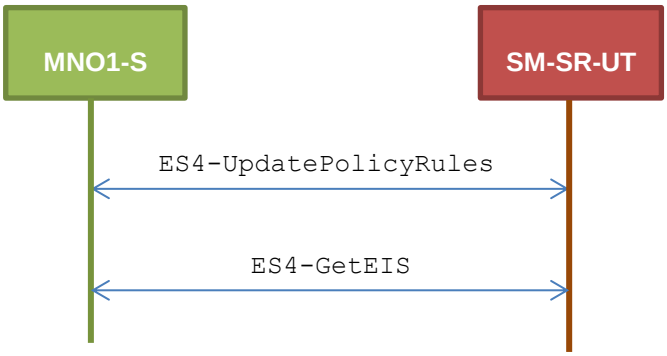
- PM_REQ22, PM_REQ23

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

- None

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)
  - {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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)		
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

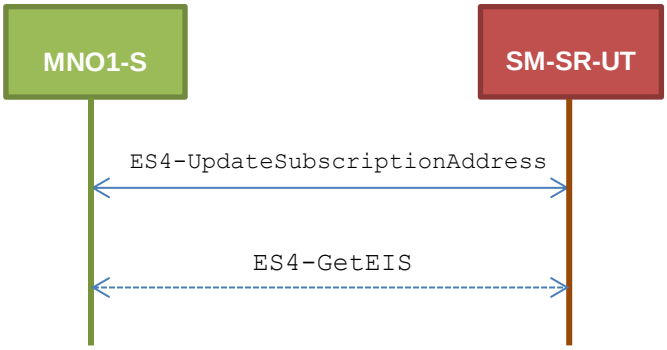
- PM_REQ22, PM_REQ24

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

- None

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)
  - `{SM_SR_ID_RPS}` has been set to `#SM_SR_UT_ID_RPS`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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

Step	Direction	Sequence / Description	Expected result	REQ
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

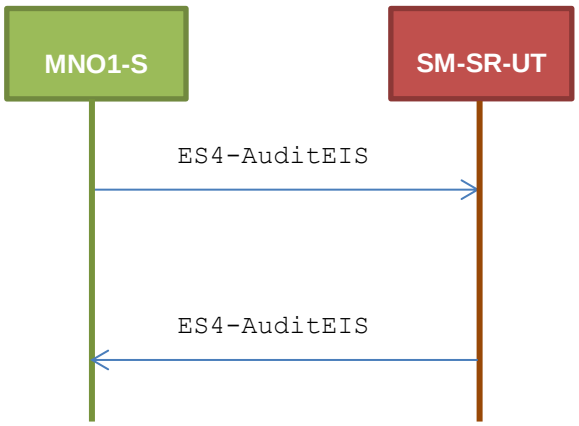
- PM_REQ25

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)		
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

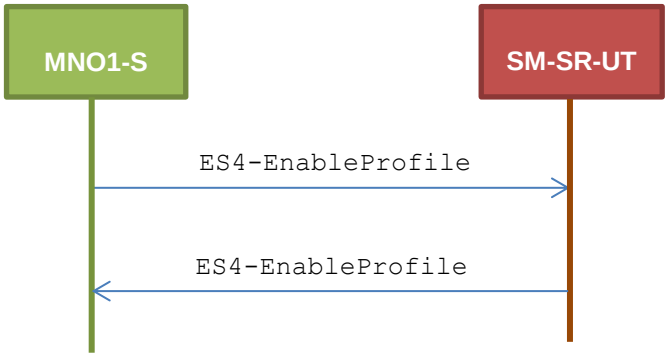
- PF_REQ24

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 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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_ID_RPS`

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	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_PLMA 3- The Reason code is equal to #RC_REFUSED	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

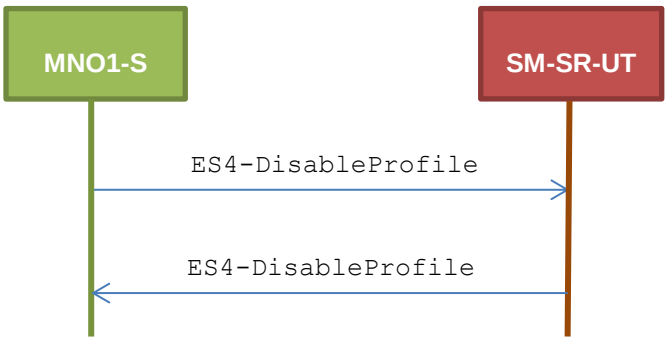
- PF_REQ25

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)		

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_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 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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_ID_RPS

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_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)		

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	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)		
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)		

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_PLMA` 3- The `Reason code` is equal to `#RC_REFUSED`	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

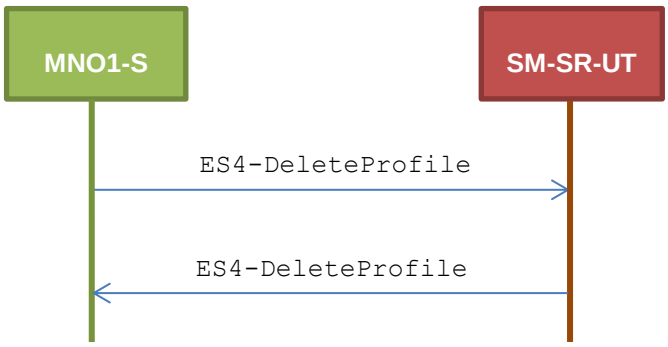
- PF_REQ26

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 Deleted 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 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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_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`
  - `{SM_DP_ID_RPS}` has been set to `#SM_DP_S_ID_RPS`

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_PLMA 3- The Reason code is equal to #RC_REFUSED	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

4.3.25.2.1.6 Test Sequence N°6 – Error Case: Profile not present in the EIS

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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_ID_RPS
- The Profile identified by the #ICCID_UNKNOWN is unknown from the SM-SR-UT

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #VIRTUAL_EID_RPS, #ICCID_UNKNOWN_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_UNKNOWN 4- The euiccResponseData is not present	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

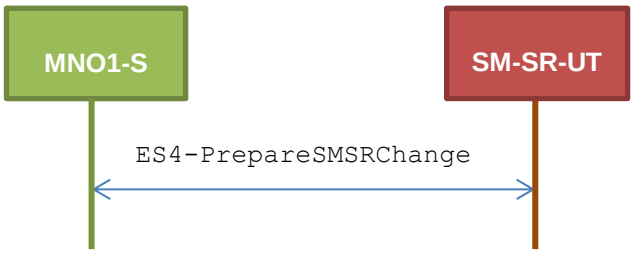
- EUICC_REQ35

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

4.3.26.2.1.3 Test Sequence N°3 – Error Case: The new SM-SR does not know the current SM-SR

Initial Conditions

- SM-SR-UT does not know #CUR_SR_S_ID

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)		

Step	Direction	Sequence / Description	Expected result	REQ
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_SM_SR` 3- The Reason code is equal to `#RC_ID_UNKNOWN`	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

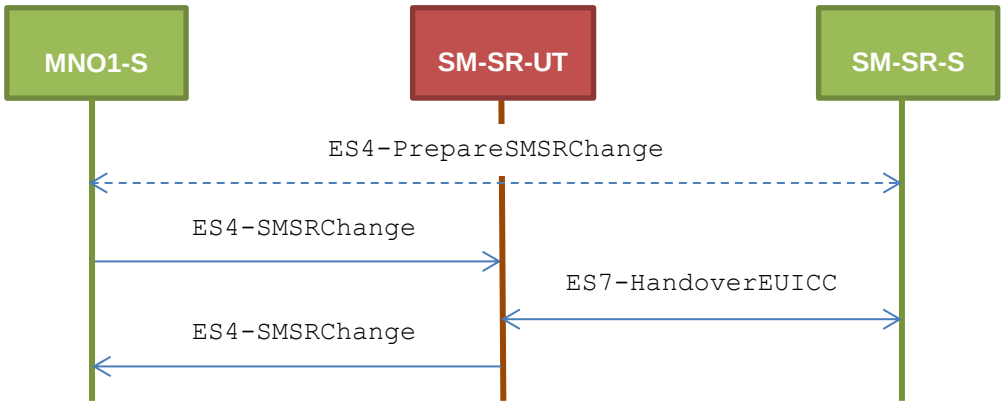
- EUICC_REQ36, EUICC_REQ39, PROC_REQ13_2

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*
- *the targeted SM-SR is unknown*

Referenced Requirements

- EUICC_REQ36, EUICC_REQ39

Initial Conditions

- The variable {SM_SR_ID_RPS} SHALL be set to #SM_SR_UT_ID_RPS
- The variable {SM_DP_ID_RPS} SHALL be set to #SM_DP_S_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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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_ES7_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
- {SM_DP_ID_RPS} has been set to #SM_DP_S_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_ES7_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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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_ES7_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

4.3.27.2.1.4 Test Sequence N°4 – Error Case: Unknown Targeted SM-SR

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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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_UNK_ID_RPS)		
2	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_SM_SR 3- The Reason code is equal to #RC_UNKNOWN	EUICC_REQ36

4.3.27.2.1.5 Test Sequence N°5 – Error Case: Handover Expires before Authenticate SM-SR

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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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_ES7_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #EXPIRED, #SC_FUNCTION, #RC_TTL_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_FUNCTION 3- The Reason code is equal to #RC_TTL_EXPIRED	EUICC_REQ36 PROC_REQ13_2

4.3.27.2.1.6 Test Sequence N°6 – Error Case: SM-SR Change expires before Authenticate SM-SR

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
  - {SM_DP_ID_RPS} has been set to #SM_DP_S_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, #SHORT_VP_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC request	The EIS is equal to #EIS_ES7_RPS	EUICC_REQ36, EUICC_REQ39
3	*Wait at least the number of seconds specified in #SHORT_VALIDITY_PERIOD* *Do not send any request or response from SM-SR-S*			

Step	Direction	Sequence / Description	Expected result	REQ
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_FUNCTION` 3- The `Reason code` is equal to `#RC_TTL_EXPIRED`	EUICC_REQ36 PROC_REQ13_2

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

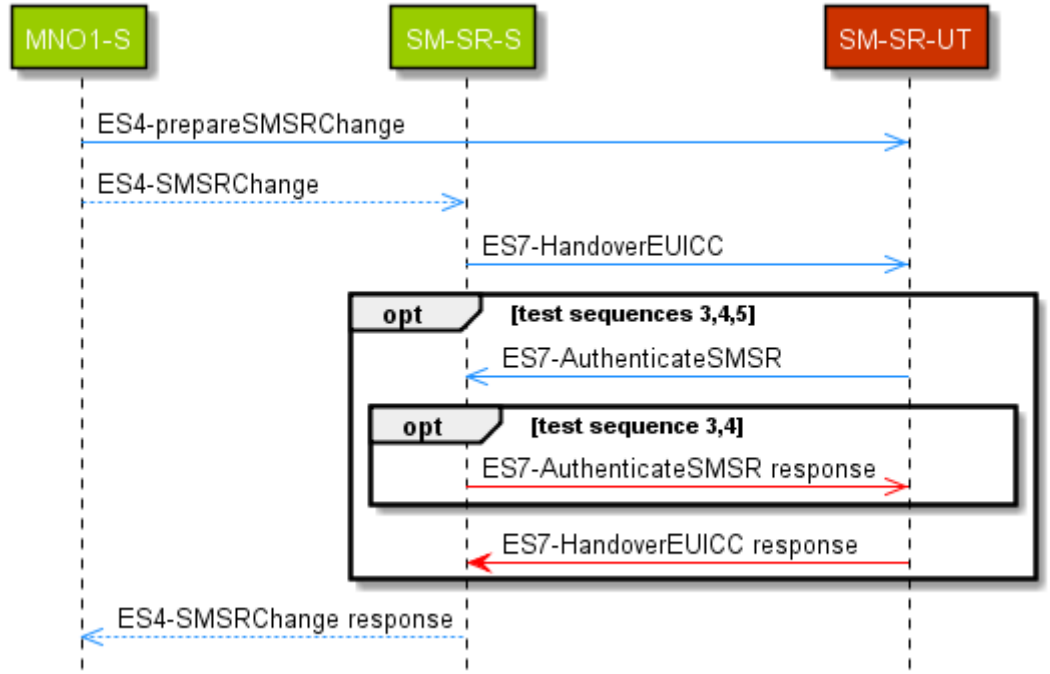
4.3.28.2 Test Cases

General Initial Conditions

- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)
- `#EUM_S_PK_ECDSA` well known to the SM-SR-UT
- `#SM_SR_S_ID` and its access point well known to the SM-SR-UT

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant OP1 as "MNO1-S" #99CC00
participant SR1 as "SM-SR-S" #99CC00
participant SR2 as "SM-SR-UT" #CC3300

OP1->>SR2: ES4-prepareSMSRChange
OP1-->>SR1: ES4-SMSRChange
SR1->>SR2: ES7-HandoverEUICC

Opt test sequences 3,4,5
SR2->>SR1: ES7-AuthenticateSMSR

Opt test sequence 3,4
SR1-[#red]>>SR2: ES7-AuthenticateSMSR response
End

'SR2->>SR1: ES7-CreateAdditionalKeyset
'SR1->SR2: ES7-CreateAdditionalKeyset response
SR2-[#red]>SR1: ES7-HandoverEUICC response
End
SR1-->>OP1: ES4-SMSRChange response

@enduml
```



Note that the function `ES4-SMSRChange` SHALL NOT be performed by the simulators (in the schema above, the corresponding request and response 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

- 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
- None

4.3.28.2.1.1 Test Sequence N°1 – Error Case: Invalid ECASD

Initial Conditions

- `#MNO1_S_ID` is well known to the SM-SR-UT
- `#MNO2_S_ID` is well known to the SM-SR-UT

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, #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

4.3.28.2.1.2 Test Sequence N°2 – Error Case: One MNO owning a profile on this eUICC is unknown by the new SM-SR

Initial Conditions

- `#MNO1_S_ID` is well known to the SM-SR-UT
- `#MNO2_S_ID` is unknown to the SM-SR-UT

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 the ES7-HandoverEUICC request	The EIS is equal to #EIS_ES7_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_EXT_RES 3- The Reason code is equal to #RC_ID_UNKNOWN	EUICC_REQ39

4.3.28.2.1.3 Test Sequence N°3 – Error Case: AuthenticateSMSR failed

Initial Conditions

- #MNO1_S_ID is well known to the SM-SR-UT
- #MNO2_S_ID is well known to the SM-SR-UT

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, #EIS_ES7_RPS)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR Request	1- The Eid is equal to #VIRTUAL_EID_RPS 2- The SmsrCertificate is a valid SM-SR certificate (tag 73/C8 equals 02)	

Step	Direction	Sequence / Description	Expected result	REQ
5	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-AuthenticateSMSR, #FAILED, #SC_FUN_PROV, RC_EXECUTION_ERROR)		
6	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC Response	The Status is equal to #FAILED	EUICC_REQ39

4.3.28.2.1.4 Test Sequence N°4 – Error Case: AuthenticateSMSR expired

Initial Conditions

- #MNO1_S_ID is well known to the SM-SR-UT
- #MNO2_S_ID is well known to the SM-SR-UT

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, #EIS_ES7_RPS)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR Request	1- The Eid is equal to #VIRTUAL_EID_RPS 2- The SmsrCertificate is a valid SM-SR certificate (tag 73/C8 equals 02)	
5	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-AuthenticateSMSR, #EXPIRED #SC_FUNCTION, RC_TTL_EXPIRED)		
6	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC Response	The Status is equal to #FAILED	EUICC_REQ39

4.3.28.2.1.5 Test Sequence N°5 – Error Case: no reply from AuthenticateSMSR

Initial Conditions

- #MNO1_S_ID is well known to the SM-SR-UT
- #MNO2_S_ID is well known to the SM-SR-UT

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, #EIS_ES7_RPS, #SHORT_VP_RPS)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR Request	1- The Eid is equal to #VIRTUAL_EID_RPS 2- The SmsrCertificate is a valid SM-SR certificate (tag 73/C8 equals 02) 3- The value of the ValidityPeriod is lower or equal to #SHORT_VALIDITY_PERIOD	
Wait at least the number of seconds specified in #SHORT_VALIDITY_PERIOD *Do not send a response*				
5	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC Response	The Status is equal to #FAILED	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

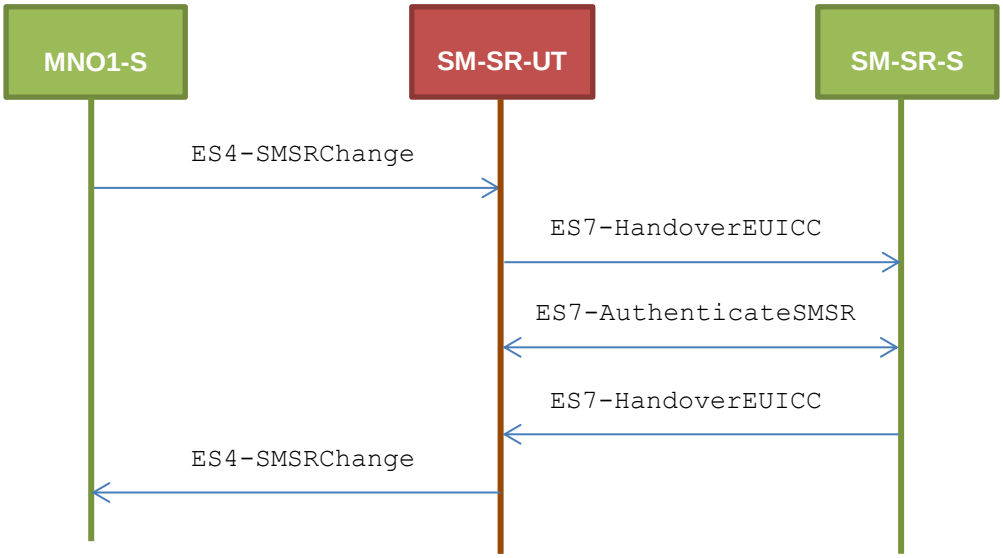
- EUICC_REQ36, EUICC_REQ39, EUICC_REQ40

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`

- {SM_SR_ID_RPS} has been set to #SM_SR_UT_ID_RPS
- {SM_DP_ID_RPS} has been set to #SM_DP_S_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_ES7_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_SM_SR_CERT 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_SM_SR_CERT, #RC_EXPIRED)		
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_SM_SR_CERT 3- The Reason code is equal to #RC_EXPIRED	EUICC_REQ39

4.3.29.2.1.2 Test Sequence N°2 – Error Case: SM-SR certificate signature cannot be verified

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_ES7_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #VIRTUAL_EID_RPS, #INVALID_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_SM_SR_CERT 3- The Reason code is equal to #RC_VERIFICATION_FAILED	EUICC_REQ40
5	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #FAILED, #SC_SM_SR_CERT, #RC_VERIFICATION_FAILED)		
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_SM_SR_CERT 3- The Reason code is equal to #RC_VERIFICATION_FAILED	EUICC_REQ39

4.3.29.2.1.3 Test Sequence N°3 – Error Case: The target SMSRid is unknown

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_UK_SR_S_ID_RPS)		
2	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_SM_SR 3- The Reason code is equal to #RC_UNKNOWN	EUICC_REQ39

4.3.30 ES7 (SM-SR – SM-SR): CreateAdditionalKeySet

4.3.30.1 Conformance Requirements

References

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

Requirements

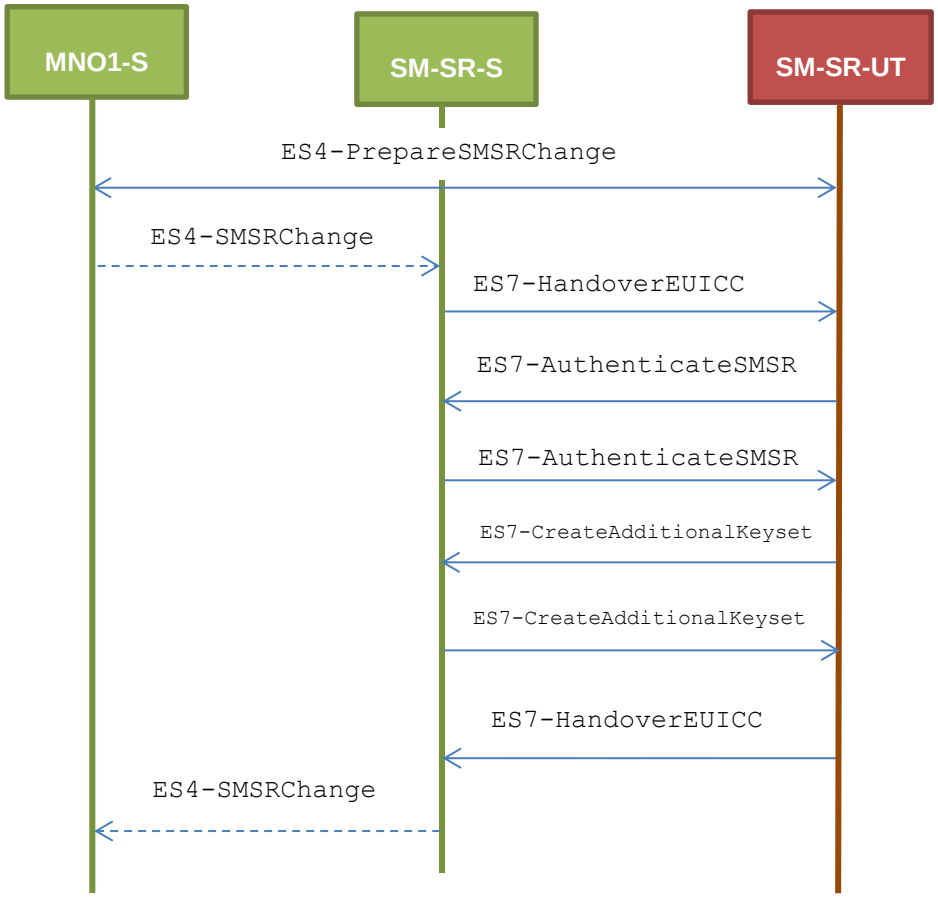
- EUICC_REQ35, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40, PROC_REQ13

4.3.30.2 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 is well known to the SM-SR-UT
- The variable {SM_SR_ID_RPS} SHALL be set to #SM_SR_S_ID_RPS
- The eUICC identified by #VIRTUAL_EID is not provisioned on 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).

4.3.30.2.1 TC.ES7.CAK.1: CreateAdditionalKeyset

Test Purpose

To ensure the method CreateAdditionalKeyset is well implemented on the SM-SR. This test proposes to simulate that an invalid receipt has been generated by the eUICC. In this case, the new SM-SR SHALL send a corresponding error code to the former SM-SR through the method HandoverEUICC.

Referenced Requirements

- EUICC_REQ35, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40, PROC_REQ13

Initial Conditions

- None

4.3.30.2.1.1 Test Sequence N°1 – Error Case: Invalid Receipt

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 PROC_REQ13
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-HandoverEUICC, #EIS_ES7_RPS)		
4	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The smsrCertificate parameter is present and contain all mandatory TLVs 3- Tag '73' of the SM-SR certificate contains tags 'C8' and 'C9' (tag 'C8' is set to '02')	EUICC_REQ40 PROC_REQ13
5	SM-SR-S→ SM-SR-UT	SEND_SUCCESS_RESP(ES7-AuthenticateSMSR, {RC}) The {RC} is randomly generated (16 bytes long)		
6	SM-SR-UT→ SM-SR-S	Send the ES7-CreateAdditionalKeyset request	1- All mandatory input parameters are present 2- The EID parameter is equal to #VIRTUAL_EID_RPS 3- scenarioParameter SHALL be set to '09', '0B', '0D' or '0F' 4- hostId parameter SHALL be empty if and only if scenarioParameter indicates that Host and Card ID are included in the key derivation process (i.e. bit3 is set to 1)	EUICC_REQ38 PROC_REQ13

Step	Direction	Sequence / Description	Expected result	REQ
7	SM-SR-S→ SM-SR-UT	SEND_SUCCESS_RESP(ES7-CreateAdditionalKeyset, {DR}, {RECEIPT}) If scenarioParameter (passed in step 6) indicates that a derivation random is included in the key derivation process (i.e. bit2 is set to 1), a {DR} of 16 bytes SHALL be randomly generated . Otherwise, the {DR} SHALL be set to an empty value.The {RECEIPT} is randomly generated (16 bytes long)		
8	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_CERT_REQ 3- The Reason code is equal to #RC_VERIFICATION_FAILED	EUICC_REQ39 PROC_REQ13

4.3.31 ES2 (MNO – SM-DP): Usage of WSA fields

4.3.31.1 Conformance Requirements

References

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

Requirements

- SOAP_REQ_B211_1, SOAP_REQ_B211_2, SOAP_REQ_B211_4, SOAP_REQ_B211_5

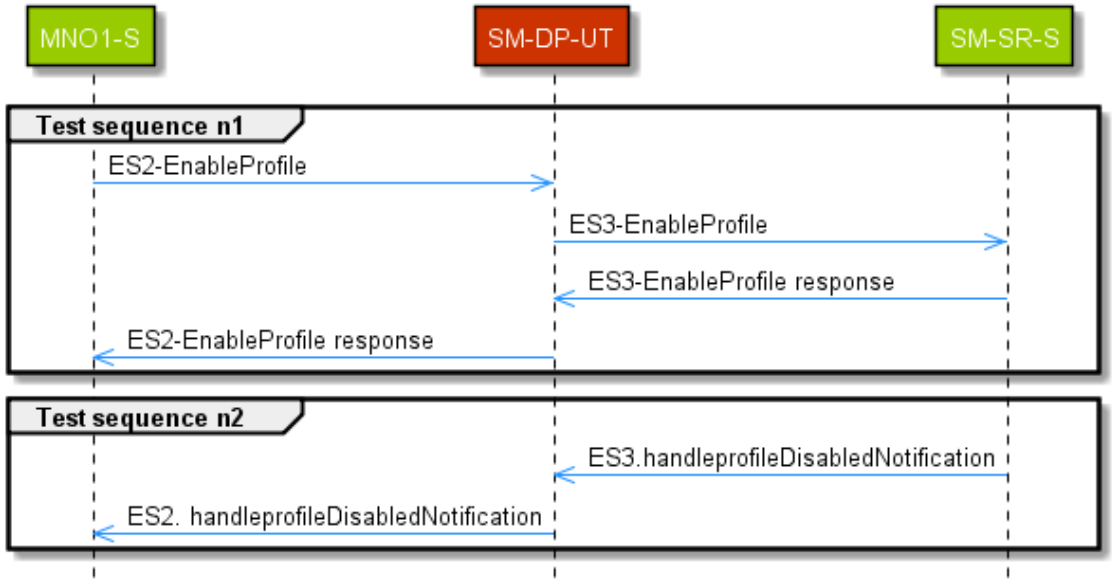
4.3.31.2 Test Cases

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

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black
      skinparam lifelineStrategy solid

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "MNO1-S" as OP #99CC00
participant "SM-DP-UT" as DP #CC3300
participant "SM-SR-S" as SR #99CC00

group Test sequence n°1
OP->>DP: ES2-EnableProfile
DP->>SR: ES3-EnableProfile

SR->>DP: ES3-EnableProfile response
DP->>OP: ES2-EnableProfile response
end

group Test sequence n°2
SR->>DP: ES3.handleprofileDisabledNotification
DP->>OP: ES2. handleprofileDisabledNotification
end

@enduml
```



4.3.31.2.1 TC.ES2.WSA.1: WSA field usage through the SM-DP

Test Purpose

To ensure an Operator can match an ES2 response to the corresponding ES2 request, and that a tracing context is maintained across the chain of calls down to the SM-SR.

Referenced Requirements

- SOAP_REQ_B211_1, SOAP_REQ_B211_2, SOAP_REQ_B211_4, SOAP_REQ_B211_5

Initial Conditions

- None

4.3.31.2.1.1 Test Sequence N°1 – WSA field from MNO down to SM-SR

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result + *comment*	REQ
1	MNO1-S → SM-DP-UT	SEND_SOAP_REQ(rps3:ES2-EnableProfileRequest, #VIRTUAL_EID_RPS, #ICCID1_RPS, <wsa:From><wsa:Address>http://example.com/?EntityId=#MNO1_S_ID</wsa:Address></wsa:From>, <wsa:To>PF_SM_DP_UT_ES2_URI?EntityId=#SM_DP_ID</wsa:To>, <wsa:MessageId>#RPS_MESSAGE_ID?TransactionId=#RPS_TRANSACTION_ID?ContextId=#RPS_CONTEXT_ID?MessageDate={CURRENT_DATE}</wsa:MessageId> <wsa:Action>http://gsma.com/ES2/PlatformManagement/ES2-EnableProfile</wsa:Action>)	The simulator shall record the {CURRENT_DATE} added in the request. This value is referred to as {DATE_OF_REQUEST} in the following.	
2	SM-DP-UT → SM-SR-S	Send the ES3-EnableProfile Request	1- The <rps:ContextId> is present and equal to #RPS_CONTEXT_ID 2- The <rps:MnoId> is present and equal to #MNO1_S_ID Note: The transport technology may or may not be SOAP; this is why the fields above are expressed in terms of the abstract <rps:xxx> fields	SOAP_REQ_B211_4
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-Enableprofile)		

Step	Direction	Sequence / Description	Expected result + *comment*	REQ
4	SM-DP-UT → MNO1-S	Send the ES2-EnableProfile Response	1- The response is a SOAP message 2- The Status is equal to #SUCCESS 3- The <wsa:From> is present and contains EntityId=# SM_DP_ID 4- The <wsa:To> is present and contains EntityId=#MNO1_S_ID 5- The <wsa:Action> is present and equals to "http://gsma.com/ES2/PlatformManagementCallback/ES2-EnableProfile 6- The <wsa:MessageId> is present and contains TransactionId=#RPS_TRANSACTION_ID 7- The <wsa:RelatesTo> is present and is equal to #RPS_MESSAGE_ID?TransactionId=#RPS_TRANSACTION_ID?ContextId=#RPS_CONTEXT_ID?MessageDate={DATE_OF_REQUEST} (i.e. the full value of the <wsa:MessageId> of the request at step 1)	SOAP_REQ_B211_1 SOAP_REQ_B211_2 SOAP_REQ_B211_5

4.3.31.2.1.2 Test Sequence N°2 – WSA fields from SM-SR up to MNO

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-SR-S → SM-DP-UT	SEND_NOTIF(ES3-HandleProfileDisabledNotification, #VIRTUAL_EID_RPS, #ICCID1_RPS, <MnoId>#MNO1_S_ID</MnoId>, <ProfileType>#PROFILE_TYPE1</ProfileType>)		

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-DP-UT → MNO1-S	Send the ES2-HandleProfileDisabledNotification notification	1- The notification is a SOAP message 2- The <wsa:From> is present and contains EntityId=# SM_DP_ID 3- The <wsa:To> is present and contains EntityId=#MNO1_S_ID 4- The <wsa:Action> is present and equal to "http://gsma.com/ES2/PlatformManagement/ES2-HandleProfileDisabledNotification" 5- The <wsa:MessageId> is present and contains ProfileId=#PROFILE_TYPE1	SOAP_REQ_B211_1 SOAP_REQ_B211_2 SOAP_REQ_B211_4

4.3.32 ES4 (M2MSP – SM-SR): SetEmergencyProfileAttribute not authorised

4.3.32.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.26_1, PF_REQ_5.4.23, PF_REQ_5.5.18

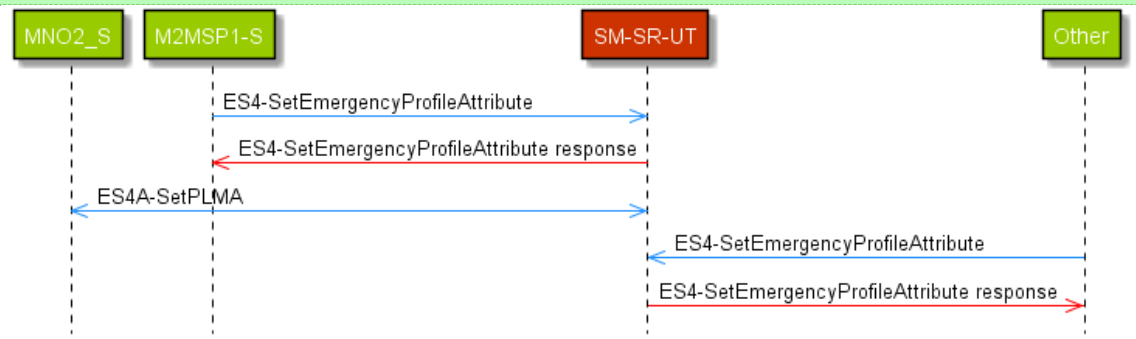
4.3.32.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO2_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO2-S and the SM-SR-UT
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS
- No PLMA is granted by MNO1 nor MNO2 on any Profile Type

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "MNO2_S" as OP2 #99CC00
participant "M2MSP1-S" as SP1 #99CC00
participant "SM-SR-UT" as SR #CC3300
participant Other #99CC00

SP1->>SR: ES4-SetEmergencyProfileAttribute
SR-[#red]>>SP1: ES4-SetEmergencyProfileAttribute response

OP2<<->>SR: ES4A-SetPLMA

Other->>SR: ES4-SetEmergencyProfileAttribute
SR-[#red]>>Other: ES4-SetEmergencyProfileAttribute response

@enduml
```



4.3.32.2.1 TC.ES4.SEPA.1: SetEmergencyProfileAttribute not authorized

Test Purpose

To ensure M2M SP cannot set the Emergency Profile Attribute if the appropriate authorisations are not granted.

Referenced Requirements

- PROC_REQ_3.26_1, PF_REQ_5.4.23, PF_REQ_5.5.18

Initial Conditions

- None

4.3.32.2.1.1 Test Sequence N°1 – Error Case: setEmergencyProfileAttribute by Operator rejected

This test sequence is FFS.

4.3.32.2.1.2 Test Sequence N°2 – Error case: setEmergencyProfileAttribute by M2M SP rejected

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-SetEmergencyProfileAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		
2	SM-SR-UT → M2MSP1-S	Send the ES4-SetEmergencyProfileAttribute Response	1- The Status is equal to #FAILED *(because M2MSP1 doesn't have authorization from MNO2 to set the Emergency profile Attribute on MNO2's Profile)* 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.26_1, PF_REQ_5.5.18
3	MNO2-S → SM-SR-UT	SEND_REQ(ES4A-SetPLMA, #PLMA_MNO2_FOR_M2MSP1_RPS,)		
4	SM-SR-UT → MNO2-S	Send the ES4A-SetPLMA Response	The Status is equal to #SUCCESS	
5	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO2_FOR_M2MSP1_RPS 2- The completion timestamp is present	
6	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetEmergencyProfileAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS, #MNO1_ID_RPS)		
7	SM-SR-UT → SM-DP-S	Send the ES3-SetEmergencyProfileAttribute Response	1- The Status is equal to #FAILED *(because MNO1 doesn't have authorisation from MNO2 to set the Fall-Back Attribute on MNO2's Profile)* 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.26_1, PF_REQ_5.4.23

4.3.33 ES4 (M2M SP – SM-SR): Enable Profile by M2M SP with errors

4.3.33.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.17.1, PROC_REQ_3.20.2, PF_REQ24, PF_REQ26, PF_REQ_5.4.16

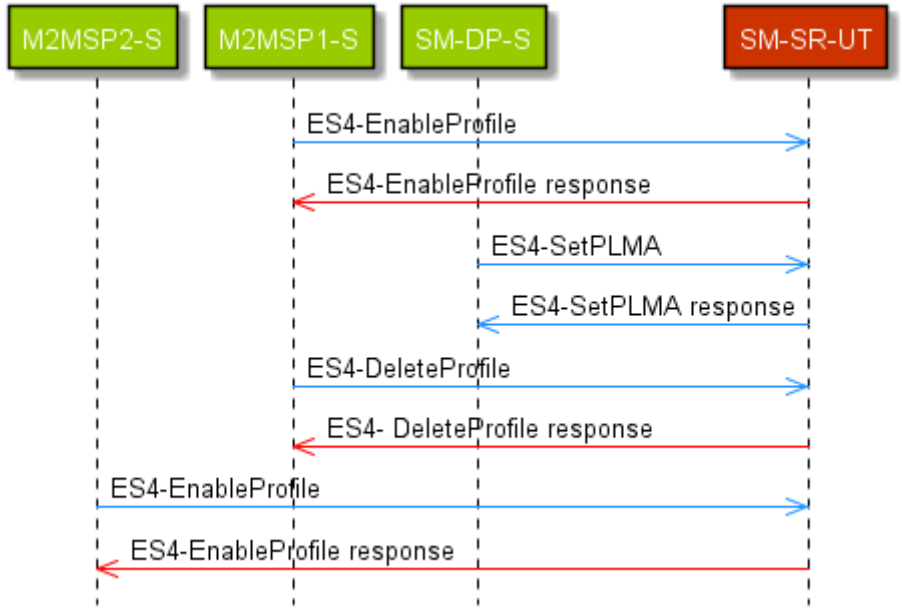
4.3.33.2 Test Cases

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
- `#M2MSP1_S_ID` and `#M2MSP1_S_ACCESSPOINT` well known to the SM-SR-UT
- `#M2MSP2_S_ID` and `#M2MSP2_S_ACCESSPOINT` well known to the SM-SR-UT
- No PLMA is granted by MNO1 nor MNO2 on any Profile Type

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "M2MSP2-S" as SP2 #99CC00
participant "M2MSP1-S" as SP1 #99CC00
participant "SM-DP-S" as OP1 #99CC00
participant "SM-SR-UT" as SR #CC3300


SP1->>SR: ES4-EnableProfile
SR-[#red]>>SP1: ES4-EnableProfile response

OP1->>SR: ES4-SetPLMA
SR->>OP1: ES4-SetPLMA response

SP1->>SR: ES4-DeleteProfile
SR-[#red]>>SP1: ES4- DeleteProfile response

SP2->>SR: ES4-EnableProfile
SR-[#red]>>SP2: ES4-EnableProfile response


@enduml
```



4.3.33.2.1 TC.ES4. EPM2MSP.1: Enable Profile by M2M SP with errors

Test Purpose

To ensure a Profile Life Cycle Management command can be executed on the targeted Profile by the SM-SR, when an M2M SP requests it, only if:

- *A PLMA has been configured by the MNO owning the profile for the requester M2M SP*
- *The commend has been explicitly authorized by the MNO*

Referenced Requirements

- PROC_REQ_3.17.1, PROC_REQ_3.20.2, PF_REQ24, PF_REQ26, PF_REQ_5.4.16

Initial Conditions

- None

4.3.33.2.1.1 Test Sequence N°1 – Normal Case: Unauthorised call rejected

Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.17.1, PF_REQ24
2	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_M2MSP1_RPS, #MNO1_S_ID)		PROC_REQ_3.20.2, PF_REQ_5.4.16
3	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA response	The Status is equal to #SUCCESS	
4	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PF_REQ26
5	M2MSP2-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PF_REQ24

4.3.34 ES4 (M2M SP– SM-SR): GetPLMA

4.3.34.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.20.2, PROC_REQ_3.20.5, PF_REQ_5.4.16, PF_REQ_5.5.17

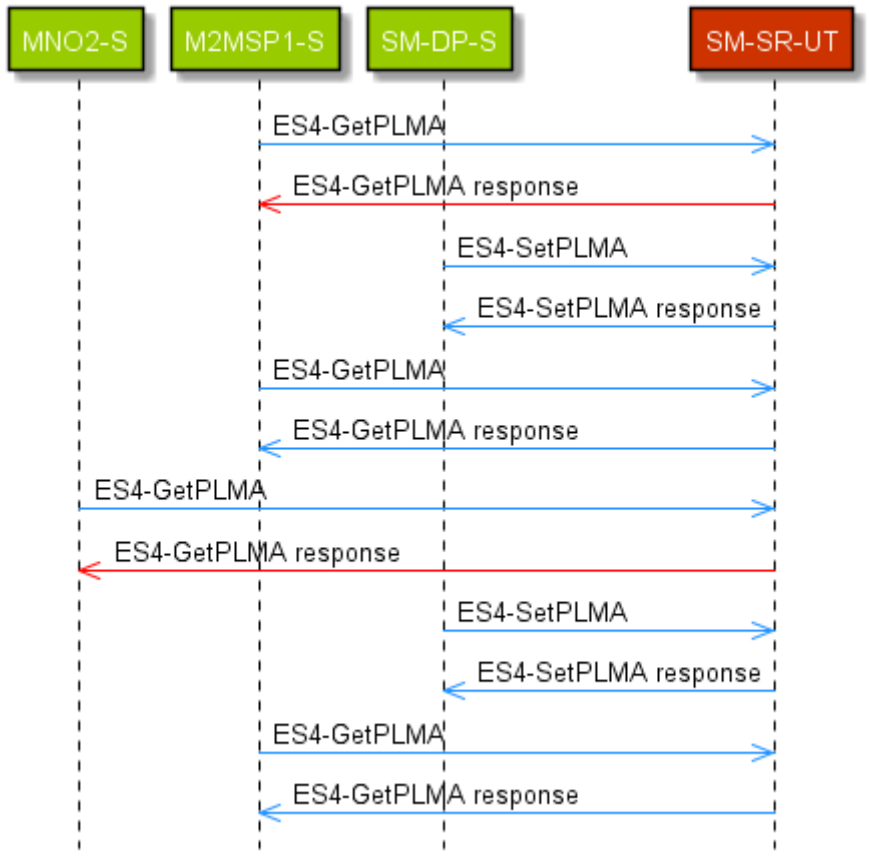
4.3.34.2 Test Cases

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`
- `#M2MSP1_S_ID` and `#M2MSP1_S_ACCESSPOINT` well known to the SM-SR-UT
- No PLMA is granted by MNO1 nor MNO2 on any Profile Type

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "MNO2-S" as OP2 #99CC00
participant "M2MSP1-S" as SP1 #99CC00
participant "SM-DP-S" as OP1 #99CC00
participant "SM-SR-UT" as SR #CC3300


SP1->>SR: ES4-GetPLMA
SR-[#red]>>SP1: ES4-GetPLMA response

OP1->>SR: ES4-SetPLMA
SR->>OP1: ES4-SetPLMA response

SP1->>SR: ES4-GetPLMA
SR->>SP1: ES4-GetPLMA response

OP2->>SR: ES4-GetPLMA
SR-[#red]>>OP2: ES4-GetPLMA response

OP1->>SR: ES4-SetPLMA
SR->>OP1: ES4-SetPLMA response

SP1->>SR: ES4-GetPLMA
SR->>SP1: ES4-GetPLMA response


@enduml
```



4.3.34.2.1 TC.ES4.GPLMA.1: Retrieve PLMA

Test Purpose

To ensure PLMA(s) can be retrieved from a SM-SR, only if:

- *At least one PLMA exists for the requester M2M SP and the targeted eUICC*
- *To verify that in case a M2M SP has been granted a PLMA on a given Profile, it will be able to retrieve all PLMAs granted for this Profile (including for other M2M SPs)*

Referenced Requirements

- PROC_REQ_3.20.2, PROC_REQ_3.20.5, PF_REQ_5.4.16, PF_REQ_5.5.17

Initial Conditions

- None

4.3.34.2.1.1 Test Sequence N°1 – Normal Case: Retrieve PLMAs from various origins

Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-GetPLMA, #ICCID1_RPS)	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_FUN_REQ 3- The Reason code is equal to #RC_NOT_ALLOWED	PF_REQ_5.5.17
2	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_M2MSP1_RPS, #MNO1_S_ID)		PROC_REQ_3.20.2, PF_REQ_5.4.16
3	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA response	The Status is equal to #SUCCESS	
4	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-GetPLMA, #ICCID1_RPS)	1- The Status is equal to #SUCCESS 2- The <Plma> parameter is equal to #PLMA_MNO1_FOR_M2MSP1_RPS	PROC_REQ_3.20.5 PF_REQ_5.5.17
5	MNO2-S → SM-SR-UT	SEND_REQ(ES4A-GetPLMA, #ICCID1_RPS)	1- The Status is equal to #FAILED 2- The Subject code is equal to #SC_FUN_REQ 3- The Reason code is equal to #RC_NOT_ALLOWED	PF_REQ_5.7.2

Step	Direction	Sequence / Description	Expected result	REQ
6	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_M2MSP2_RPS, #MNO1_S_ID)		PROC_REQ_3.20.2 PF_REQ_5.4.16
7	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA Response	The Status is equal to #SUCCESS	
8	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-GetPLMA, #ICCID1_RPS)	1- The Status is equal to #SUCCESS 2- The response contains two <Plma> parameters, in any order 3- One <Plma> parameter is equal to #PLMA_MNO1_FOR_M2MSP1_RPS 4- One other <Plma> parameter is equal to PLMA_MNO1_FOR_M2MSP2_RPS	PROC_REQ_3.20.5 PF_REQ_5.5.17

4.3.35 ES2 (MNO - SM-DP): AuditEIS

4.3.35.1 Conformance Requirements

References

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

Requirements

- PM_REQ15, PF_REQ_5.3.12

4.3.35.2 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

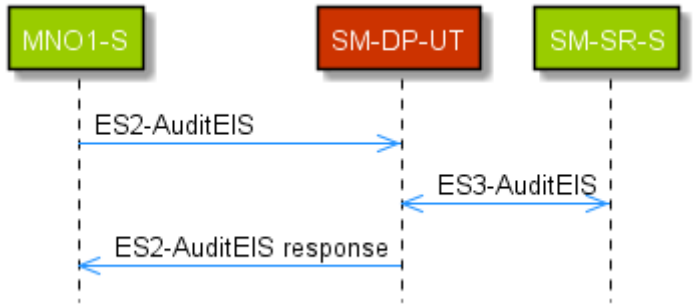
4.3.35.2.1 TC.ES2.AEIS.1: AuditEIS via ES2

Test Purpose

- *To ensure that an Operator is able to retrieve an eUICC capability by calling the AuditEIS function via ES2/ES3*
- *Only information that are related to profiles owned by the calling MNO can be retrieved through this function*

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "MNO1-S" as OP1 #99CC00
participant "SM-DP-UT" as DP #CC3300
participant "SM-SR-S" as SR #99CC00



OP1->>DP: ES2-AuditEIS

DP<<->>SR: ES3-AuditEIS

DP->>OP1: ES2-AuditEIS response



@enduml
```



Referenced Requirements

- PM_REQ15, PF_REQ_5.3.12

Initial Conditions

- None

4.3.35.2.1.1 Test Sequence N°1 – Normal Case: AuditEIS via ES2

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-DP-UT	SEND_REQ(ES2-AUDIT-EIS, #VIRTUAL_EID_RPS, #SM_SR_ID_RPS})		PF_REQ_5.3.12

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-DP-UT → SM-SR-S	Send the ES3-AuditEIS request	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The MnoId parameter is equal to #MNO1_ID_RPS	PM_REQ15
3	SM-SR-S → SM-DP-UT	SEND_SUCCESS_RESP(ES3-AuditEIS, #EIS_ES3_RPS) Note: the SM-SR-S SHALL only include the profile #PROFILE1_RPS in this EIS		PM_REQ15
4	SM-DP-UT → MNO1-S	Send the ES2-AuditEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_ES2_RPS	PF_REQ_5.3.12

4.3.36 ES4 (MNO – SM-SR and M2MSP – SM-SR): SetFallBackAttribute not authorised

4.3.36.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.27_1, PROC_REQ_3.27_2, PROC_REQ_3.29_1, PF_REQ_5.5.21

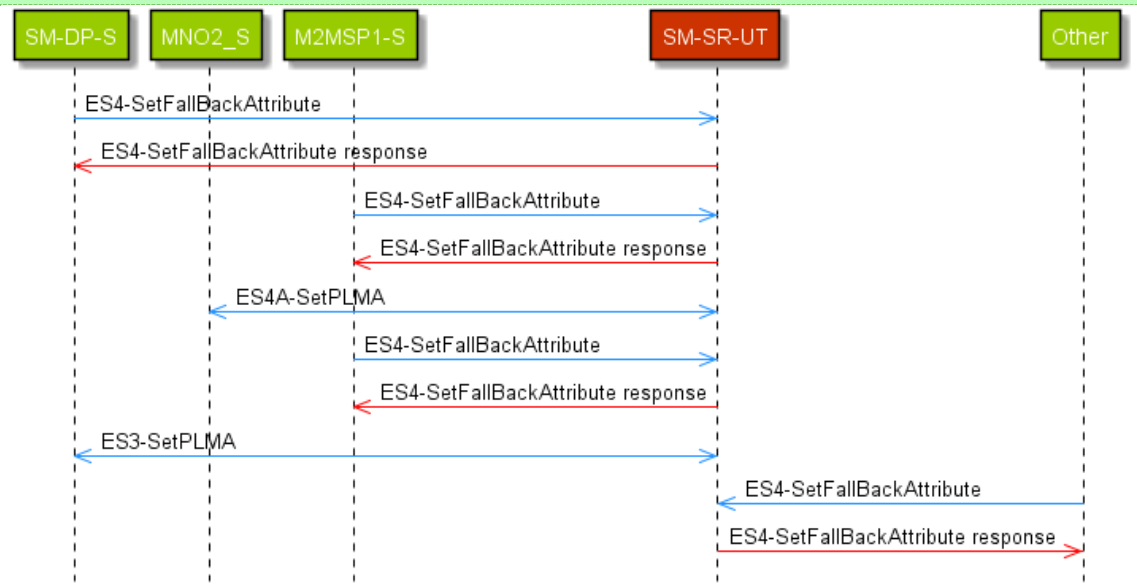
4.3.36.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO2_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO2-S and the SM-SR-UT
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- #M2MSP2_S_ID and # M2MSP2_S_ACCESSPOINT well known to the SM-SR-UT
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS
- No PLMA is granted by MNO1 nor MNO2 on any Profile Type

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant OP1 as "SM-DP-S" #99CC00
participant OP2 as "MNO2_S" #99CC00
participant SP as "M2MSP1-S" #99CC00
participant SR as "SM-SR-UT" #CC3300
participant Other #99CC00

OP1->>SR: ES4-SetFallBackAttribute
SR-[#red]>>OP1: ES4-SetFallBackAttribute response

SP->>SR: ES4-SetFallBackAttribute
SR-[#red]>>SP: ES4-SetFallBackAttribute response

OP2<<->>SR: ES4A-SetPLMA

SP->>SR: ES4-SetFallBackAttribute
SR-[#red]>>SP: ES4-SetFallBackAttribute response

OP1<<->>SR: ES3-SetPLMA

Other->>SR: ES4-SetFallBackAttribute
SR-[#red]>>Other: ES4-SetFallBackAttribute response

@enduml
```



4.3.36.2.1 TC.ES4.SFBA.1: SetFallBackAttribute not authorized

Test Purpose

To ensure an Operator or M2M SP cannot set the Fall-Back Attribute if the appropriate authorisations are not granted.

Referenced Requirements

- PROC_REQ_3.27_1, PROC_REQ_3.27_2, PROC_REQ_3.29_1, PF_REQ_5.5.21

Initial Conditions

- None

4.3.36.2.1.1 Test Sequence N°1 – Error Case: setFallBackAttribute by Operator rejected

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result + *comment*	REQ
1	MNO2-S → SM-SR-UT	SEND_REQ(ES4-SetFallBackAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		
2	SM-SR-UT → MNO2-S	Send the ES4-SetFallBackAttribute response	1- The Status is equal to #FAILED *(because MNO2 doesn't have authorization from MNO1 to "unset" the Fall-Back Attribute from MNO1's Profile)* 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.27_1 PROC_REQ_3.27_2
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_MNO2_RPS, #MNO1_ID_RPS)		
4	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA response	The Status is equal to #SUCCESS	
5	SM-SR-UT → MNO2-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO1_FOR_MNO2_RPS 2- The completion timestamp is present	
6	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-SetFallBackAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		

Step	Direction	Sequence / Description	Expected result + *comment*	REQ
7	SM-SR-UT → M2MSP1-S	Send the ES4-SetFallBackAttribute response	1- The Status is equal to #FAILED *(because M2MSP1 doesn't have authorization on "set" or "unset" the Fall-Back Attribute on any Profile)* 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.27_1 PROC_REQ_3.27_2

4.3.36.2.1.2 Test Sequence N°2 – Error case: setFallBackAttribute by M2M SP rejected

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-SetFallBackAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		
2	SM-SR-UT → MNO2-S	Send the ES4-SetFallBackAttribute response	1- The Status is equal to #FAILED (because M2MSP1 doesn't have authorization from MNO2 to set the Fall-Back Attribute on MNO2's Profile) 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.29_1
3	MNO2-S → SM-SR-UT	SEND_REQ(ES4A-SetPLMA, #PLMA_MNO2_FOR_M2MSP1_RPS)		
4	SM-SR-UT → MNO2-S	Send the ES4A-SetPLMA response	The Status is equal to #SUCCESS	
5	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO2_FOR_M2MSP1_RPS 2- The completion timestamp is present	

Step	Direction	Sequence / Description	Expected result	REQ
6	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-SetFallBackAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		
7	SM-SR-UT → M2MSP1-S	Send the ES4-SetFallBackAttribute response	1- The Status is equal to #FAILED *(because M2MSP1 doesn't have authorisation from MNO1 to "unset" the Fall-Back Attribute from MNO1's Profile)* 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.29_1
8	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_M2MSP1_RPS, #MNO1_ID_RPS)		
9	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA response	The Status is equal to #SUCCESS	
10	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO1_FOR_M2MSP1_RPS 2- The completion timestamp is present	
11	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetFallBackAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS, #MNO1_ID_RPS)		
12	SM-SR-UT → SM-DP-S	Send the ES3-SetFallBackAttribute response	1- The Status is equal to #FAILED *(because MNO1 doesn't have authorisation from MNO2 to set the Fall-Back Attribute on MNO2's Profile)* 2- The Subject code is equal to #SC_PLMA 3- The Reason code is equal to #RC_REFUSED	PROC_REQ_3.29_1

4.4 OTA Layer Testing

4.4.1 Generic Sub-Sequences

4.4.1.1 Set Fall-Back Attribute from SM-SR-UT

Some Test Sequences related to Fall-Back Attribute management by an SM-SR-UT will need to stub the behaviour of an eUICC. In order to not depend on a real eUICC, the following sub-sequence is defined:

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-SR-UT → Device-Network-S	ES5-CreateISDP function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. SetFallbackAttribute (STORE DATA with DGI 3A05 3- The targeted ISD-P (tag 4F) is #ISD_P_AID3	PF_REQ9
2	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including `[R_AB_9000]` or `[R_AF_9000]` depending on the transport protocol		PF-REQ9

4.4.1.2 EnableProfile from SM-SR-UT

Some Test Sequences related to enabling a Profile by an SM-SR-UT will need to stub the behaviour of an eUICC. In order to not depend on a real eUICC, the following sub-sequence is defined:

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-SR-UT → Device-Network-S	ES5-EnableProfile function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. Enableprofile(STORE DATA with DGI 3A03 3- The targeted ISD-P (tag 4F) is #ISD_P_AID2	PF_REQ4
2	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including `[R_AB_9000]` or `[R_AF_9000]` depending on the transport protocol		
3	Device-Network-S → SM-SR	The Device-Network-S sends a notification confirming the proper enabling of the new Profile, over SMS, CAT_TP or HTTPs		

Step	Direction	Sequence / Description	Expected result	REQ
4	SM-SR-UT → Device-Network-S	ES5-HandleNotificationConfirmation function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. HandleNotificationConfirmation(STORE DATA with DGI 3A08 3- The notification sequence number (tag 4E) is the same as in the notification at step3	EUICC_REQ29
5	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including `[R_AB_9000]` or `[R_AF_9000]` depending on the transport protocol		

4.4.1.3 DisableProfile from SM-SR-UT

Some Test Sequences related to disabling a Profile by an SM-SR-UT will need to stub the behaviour of an eUICC. In order to not depend on a real eUICC, the following sub-sequence is defined:

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-SR-UT → Device-Network-S	ES5-DisableProfile function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. DisableProfile(STORE DATA with DGI 3A04 3- The targeted ISD-P (tag 4F) is #ISD_P_AID3	PF_REQ5
2	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including `[R_AB_9000]` or `[R_AF_9000]` depending on the transport protocol		
3	Device-Network-S → SM-SR	The Device-Network-S sends a notification confirming the proper enabling of the new Profile, over SMS, CAT_TP or HTTPs		

Step	Direction	Sequence / Description	Expected result	REQ
4	SM-SR-UT → Device-Network-S	ES5-HandleNotificationConfirmation function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. HandleNotificationConfirmation(STORE DATA with DGI 3A08 3- The notification sequence number (tag 4E) is the same as in the notification at step3	EUICC_REQ29
5	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including `[R_AB_9000]` or `[R_AF_9000]` depending on the transport protocol		

4.4.1.4 Set Emergency Profile Attribute from SM-SR-UT

Some Test Sequences related to Emergency Profile Attribute management by an SM-SR-UT will need to stub the behaviour of an eUICC. In order to not depend on a real eUICC, the following sub-sequence is defined:

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-SR-UT → Device-Network-S	ES5-SetEmergencyProfileAttribute function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. SetEmergencyProfileAttribute (STORE DATA with DGI 3A09 3- The targeted ISD-P (tag 4F) is #ISD_P_AID3	
2	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including `[R_AB_9000]` or `[R_AF_9000]` depending on the transport protocol		

4.4.1.5 First part of ISD-R Keyset Establishment from SM-SR-UT

Some Test Sequences related to SM-SR Change from an SM-SR1 being the SM-SR-UT will need to stub the behaviour of an eUICC. In order to not depend on a real eUICC, the following sub-sequence is defined:

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-SR-UT → Device-Network-S	ES5- EstablishISDRKeySet function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. EstablishISDRKeySet,1st STORE DATA (STORE DATA with DGI 3A01 3- The Certificate contained within the DGI 3A01 (starting with tag 7F21) is equal to #VALID_SM_SR_CERTIFICATE	PROC_REQ13
2	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including [R_AB_RC] or [R_AF_RC] depending on the transport protocol		

4.4.1.6 Second part of ISD-R Keyset Establishment from SM-SR-UT

Some Test Sequences related to SM-SR Change from an SM-SR1 being the SM-SR-UT will need to stub the behaviour of an eUICC. In order to not depend on a real eUICC, the following sub-sequence is defined:

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-SR-UT → Device-Network-S	ES5- EstablishISDRKeySet function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	The Device-Network-S decrypts the SMS or CAT-TP or HTTPS packet, and verifies: 1- The command is targeting the ISD-R (TAR=000001) 2- The command is a ES5. EstablishISDRKeySet,2nd STORE DATA (STORE DATA with DGI 3A02	PROC_REQ13
2	Device-Network-S → SM-SR	The Device-Network-S sends the OTA response to the SM-SR-UT, including [R_AB_RECEIPT] or [R_AF_RECEIPT] depending on the transport protocol		

4.4.2 ES3 (SM-DP – SM-SR): AuditEIS

This test case is defined as FFS pending a future version of this document.

4.4.3 ES3 (SM-DP – SM-SR) and ES4 (MNO - SM-SR): usage of WSA fields

4.4.3.1 Conformance Requirements

References

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

Requirements

- SOAP_REQ_B211_1, SOAP_REQ_B211_2, SOAP_REQ_B211_4, SOAP_REQ_B211_5

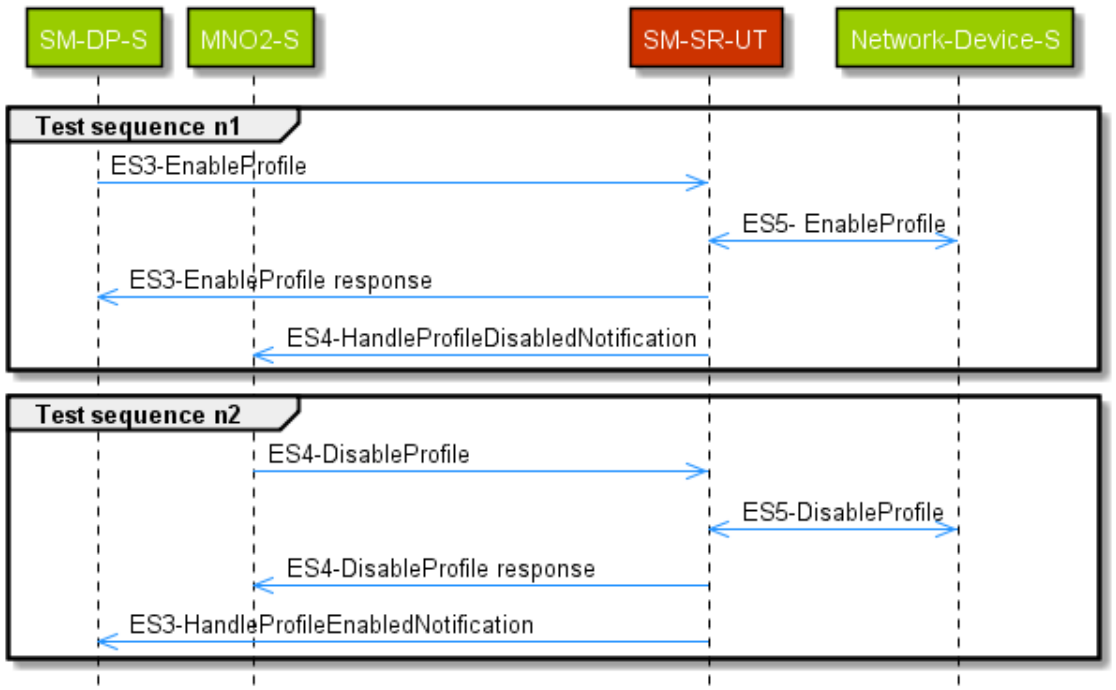
4.4.3.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO2_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO2-S and the SM-SR-UT
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- #M2MSP2_S_ID and #M2MSP2_S_ACCESSPOINT well known to the SM-SR-UT
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS
- No PLMA is granted by MNO1 nor MNO2 on any Profile Type

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "SM-DP-S" as OP1 #99CC00
participant "MNO2-S" as OP2 #99CC00
participant "SM-SR-UT" as SR #CC3300
participant "Network-Device-S" as eUICC #99CC00

group Test sequence n1
OP1->>SR: ES3-EnableProfile
SR<<->>eUICC: ES5- EnableProfile
SR->>OP1: ES3-EnableProfile response

SR->>OP2: ES4-HandleProfileDisabledNotification
end

group Test sequence n2
OP2->>SR: ES4-DisableProfile
SR<<->>eUICC: ES5-DisableProfile
SR->>OP2: ES4-DisableProfile response

SR->>OP1: ES3-HandleProfileEnabledNotification
end

@enduml
```



4.4.3.2.1 TC.ES3ES4.WSA.1: WSA fields in request/response/notification

Test Purpose

To ensure an Operator and an SM-DP can match an ES4 (respectively, ES3) response to the corresponding request, and that the notifications includes the concerned Profile, and the ES3 notification includes the target MnoId.

Referenced Requirements

- SOAP_REQ_B211_1, SOAP_REQ_B211_2, SOAP_REQ_B211_4, SOAP_REQ_B211_5

Initial Conditions

- None

4.4.3.2.1.1 Test Sequence N°1 – WSA fields in ES3 request/response and ES4 notification

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result + comment	REQ
1	SM-DP-S → SM-SR-UT	SEND_SOAP_REQ(rps3:ES3-EnableProfileRequest, #VIRTUAL_EID_RPS, #ICCID1_RPS, <wsa:From><wsa:Address>http://example.com/?EntityId=#SM_DP_S_ID?MnoId=#MNO1_S_ID </wsa:Address></wsa:From>, <wsa:To>PF_SM_SR_UT_ES3_URI?EntityId=#SM_SR_ID</wsa:To>, <wsa:MessageId>#RPS_MESSAGE_ID?TransactionId=#RPS_TRANSACTION_ID?ContextId=#RPS_CONTEXT_ID?MessageDate={CURRENT_DATE}</wsa:MessageId> <wsa:Action>http://gsma.com/ES3/PlatformManagement/ES3-EnableProfile</wsa:Action>)	The simulator shall record the {CURRENT_DATE} added in the request. This value is referred to as {DATE_OF_REQUEST} in the following.	
2	Execute sub-sequence 4.4.1.2 EnableProfile from SM-SR-UT			

Step	Direction	Sequence / Description	Expected result + comment	REQ
3	SM-SR-UT → SM-DP-S	Send the ES3-EnableProfile Response	1- The response is a SOAP message 2- The Status is equal to #SUCCESS 3- The <wsa:From> is present and contains EntityId=#SM_SR_ID 4- The <wsa:To> is present and contains EntityId=#SM_DP_S_ID 5- The <wsa:To> also contains MnoId=#MNO1_S_ID 6- The <wsa:Action> is present and equals "http://gsma.com/ES3/PlatformManagementCallback/ES3-EnableProfile" 7- The <wsa:MessageId> is present and contains TransactionId=#RPS_TRANSACTION_ID 8- The <wsa:RelatesTo> is present and is equal to #RPS_MESSAGE_ID?TransactionId=#RPS_TRANSACTION_ID?ContextId=#RPS_CONTEXT_ID?MessageDate={DATE_OF_REQUEST} (i.e. the full value of the <wsa:MessageId> of the request at step 1)	SOAP_REQ_B211_1 SOAP_REQ_B211_2 SOAP_REQ_B211_5
Check notifications				
4	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileDisabledNotification Notification	1- The notification is a SOAP message 2- The <wsa:From> is present and contains EntityId=#SM_SR_ID 3- The <wsa:To> is present and contains EntityId=#MNO2_S_ID 4- The <wsa:Action> is present and equal to "http://gsma.com/ES42/PlatformManagement/ES4-HandleProfileDisabledNotification" 5- The <wsa:MessageId> is present and contains ProfileId=#PROFILE_TYPE2	SOAP_REQ_B211_1 SOAP_REQ_B211_2 SOAP_REQ_B211_4

4.4.3.2.1.2 Test Sequence N°2 – WSA fields in ES4 request/response and ES3 notification

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result + comment	REQ
1	MNO2-S → SM-SR-UT	SEND_SOAP_REQ(rps3:ES4-DisableProfileRequest, #VIRTUAL_EID_RPS, #ICCID2_RPS, <wsa:From><wsa:Address>http://example.com/?EntityId=#MNO2_S_ID</wsa:Address></wsa:From>, <wsa:To>PF_SM_SR_UT_ES4_URI?EntityId=#SM_SR_ID</wsa:To>, <wsa:MessageId>#RPS_MESSAGE_ID?TransactionId=#RPS_TRANSACTION_ID?ContextId=#RPS_CONTEXT_ID?MessageDate={CURRENT_DATE}</wsa:MessageId> <wsa:Action>http://gsma.com/ES4/PlatformManagement/ES4-DisableProfile</wsa:Action>)	The simulator shall record the {CURRENT_DATE} added in the request. This value is referred to as {DATE_OF_REQUEST} in the following.	
2	Execute sub-sequence 4.4.1.3 DisableProfile from SM-SR-UT			
3	SM-SR-UT → MNO2-S	Send the ES4-DisableProfile Response	1- The response is a SOAP message 2- The Status is equal to #SUCCESS 3- The <wsa:From> is present and contains EntityId=#SM_SR_ID 4- The <wsa:To> is present and contains EntityId=#MNO2_S_ID 5- The <wsa:Action> is present and equals "http://gsma.com/ES4/PlatformManagementCallback/ES4-DisableProfile" 6- The <wsa:MessageId> is present and contains TransactionId=#RPS_TRANSACTION_ID 7- The <wsa:RelatesTo> is present and is equal to #RPS_MESSAGE_ID?TransactionId=#RPS_TRANSACTION_ID?ContextId=#RPS_CONTEXT_ID?MessageDate={DATE_OF_REQUEST} (i.e. the full value of the <wsa:MessageId> of the request at step 1)	SOAP_REQ_B211_1 SOAP_REQ_B211_2 SOAP_REQ_B211_5
Check notifications				

Step	Direction	Sequence / Description	Expected result + comment	REQ
4	SM-SR-UT → SM-DP-S	Send the ES3-HandleProfileEnabledNotification Notification	1- The notification is a SOAP message 2- The <wsa:From> is present and contains EntityId=#SM_SR_ID 3- The <wsa:To> is present and contains EntityId=#SM_DP_S _ID 4- The <wsa:To> also contains MnoId=#MNO1_S_ID 5- The <wsa:Action> is present and equal to "http://gsma.com/ES3/PlatformManagement/ES3-HandleProfileEnabledNotification" 6- The <wsa:MessageId> is present and contains ProfileId=#PROFILE_TYPE1	SOAP_REQ_B211_1 SOAP_REQ_B211_2 SOAP_REQ_B211_4

4.4.4 ES3 (SM-DP - SM-SR): DisableProfile by M2M SP (via the SM-DP of a MNO)

4.4.4.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.20.1, PF_REQ19, PF_REQ27, PF_REQ_5.7.1, PF_REQ_5.4.20

4.4.4.2 Test Cases

General Initial Conditions

- #MNO1_S_ID well known to the SM-SR-UT
- #MNO2_S_ID and #MNO2_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
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- #M2MSP2_S_ID and well known to the SM-SR-UT
- No PLMA is granted by MNO1 on any Profile Type
- No ONC is configured for MNO1
- No PLMA is granted by MNO2 on any Profile Type
- No ONC is configured for MNO2
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS

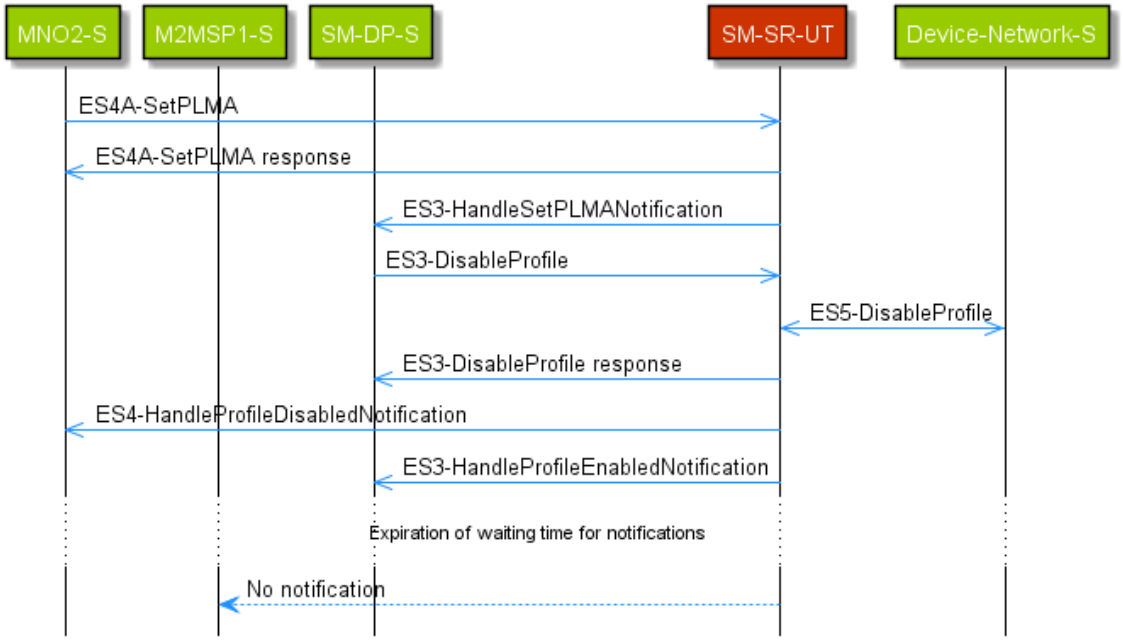
4.4.4.2.1 TC.ES3.EPM2MSP.1: DisableProfile by M2M SP

Test Purpose

To ensure that an MNO in the role of M2M SP is able to Disable a Profile via ES2/ES3, as soon as it is authorized by the MNO owning the profile.
To verify that notifications are sent to relevant parties on Profile status change.

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "MNO2-S" as OP2 #99CC00
participant "M2MSP1-S" as SP1 #99CC00
participant "SM-DP-S" as OP1 #99CC00
participant "SM-SR-UT" as SR #CC3300
participant "Device-Network-S" as eUICC #99CC00

OP2->>SR: ES4A-SetPLMA
SR->>OP2: ES4A-SetPLMA response

SR->>OP1: ES3-HandleSetPLMANotification

OP1->>SR: ES3-DisableProfile

SR<<->>eUICC: ES5-DisableProfile

SR->>OP1: ES3-DisableProfile response

SR->>OP2: ES4-HandleProfileDisabledNotification

SR->>OP1: ES3-HandleProfileEnabledNotification

... Expiration of waiting time for notifications...
SR-->SP1: No notification


@enduml
```



Referenced Requirements

- PROC_REQ_3.20.1, PF_REQ19, PF_REQ27, PF_REQ_5.7.1, PF_REQ_5.4.20

Initial Conditions

- None

4.4.4.2.1.1 Test Sequence N°1 – Normal Case: PLMA for M2M SP via ES3 and no ONC, Disable Profile by M2M SP

Initial Conditions

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO2-S → SM-SR-UT	SEND_REQ(ES4A-SetPLMA, #PLMA_MNO2_FOR_MNO1_RPS, #MNO2_S_ID)		PROC_REQ_3.20.1, PF_REQ_5.7.1
2	SM-SR-UT → MNO2-S	Send the ES4A-SetPLMA response	The Status is equal to #SUCCESS	
3	SM-SR-UT → SM-DP-S	Send the ES3-HandleSetPLMANotification notification	1- The <Plma> parameter is equal to #PLMA_MNO2_FOR_MNO1_RPS 2- The completion timestamp is present 3- The <rps:MnoId> is present and equal to #MNO1_S_ID 4- The <rps:ProfileType> is present and equal to #PROFILE_TYPE2	PROC_REQ_3.20.1, PF_REQ_5.4.20
4	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-DisableProfile, #VIRTUAL_EID_RPS, #ICCID2_RPS, #MNO1_ID_RPS)		PF_REQ19
5	See sub-sequence 4.4.1.3 DisableProfile from SM-SR-UT			
6	SM-SR-UT → SM-DP-S	Send the ES3-DisableProfile response	The Status is equal to #SUCCESS	
7	SM-SR-UT → MNO2-S	Send the ES4A-HandleProfileDisabledNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is present 4- The <rps:ProfileType> is present and equal to #PROFILE_TYPE2	PF_REQ27

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-UT → SM-DP-S	Send the ES3-HandleProfileEnabledNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- The completion timestamp is present 4- The <rps:MnoId> is present and equal to #MNO1_S_ID 5- The <rps:ProfileType> is present and equal to #PROFILE_TYPE1	PF_REQ22
9	Check that M2MSP1 does not receive notification after 1mn			
Note: steps 6-7-8 can occur in any order				

4.4.5 ES4 (MNO – SM-SR and M2MSP – SM-SR): SetFallBackAttribute authorized

4.4.5.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.27_1, PROC_REQ_3.27_2, PROC_REQ_3.29_1, PF_REQ_5.4.28, PF_REQ_5.5.21, PF_REQ_5.5.22, PF_REQ_5.5.23

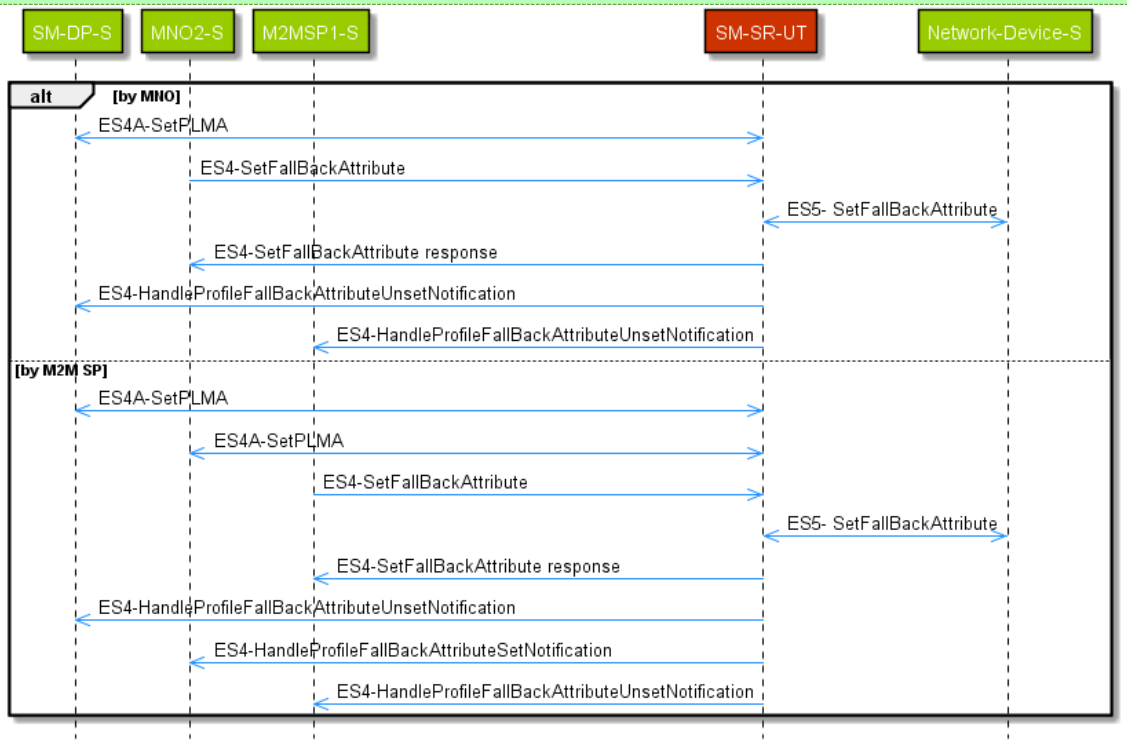
4.4.5.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO2_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO2-S and the SM-SR-UT
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- #M2MSP2_S_ID and # M2MSP2_S_ACCESSPOINT well known to the SM-SR-UT
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS
- No PLMA is granted by MNO1 nor MNO2 on any Profile Type

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant OP1 as "SM-DP-S" #99CC00
participant OP2 as "MNO2-S" #99CC00
participant SP as "M2MSP1-S" #99CC00
participant SR as "SM-SR-UT" #CC3300
participant eUICC as "Network-Device-S" #99CC00

alt by MNO
OP1<<->>SR: ES4A-SetPLMA

OP2->>SR: ES4-SetFallBackAttribute
SR<<->>eUICC: ES5- SetFallBackAttribute
SR->>OP2: ES4-SetFallBackAttribute response

SR->>OP1: ES4-HandleProfileFallBackAttributeUnsetNotification
SR->>SP: ES4-HandleProfileFallBackAttributeUnsetNotification

else by M2M SP
OP1<<->>SR: ES4A-SetPLMA

OP2<<->>SR: ES4A-SetPLMA

SP->>SR: ES4-SetFallBackAttribute
SR<<->>eUICC: ES5- SetFallBackAttribute
SR->>SP: ES4-SetFallBackAttribute response

SR->>OP1: ES4-HandleProfileFallBackAttributeUnsetNotification
SR->>OP2: ES4-HandleProfileFallBackAttributeSetNotification
SR->>SP: ES4-HandleProfileFallBackAttributeUnsetNotification
End

@enduml
```



4.4.5.2.1 TC.ES4.SFBA.2: SetFallBackAttribute authorised

Test Purpose

To ensure an Operator or M2M SP can set the Fall-Back Attribute if the appropriate authorisations are granted.

Referenced Requirements

- PROC_REQ_3.27_1, PROC_REQ_3.27_2, PROC_REQ_3.29_1, PF_REQ_5.4.28, PF_REQ_5.5.21, PF_REQ_5.5.22, PF_REQ_5.5.23

Initial Conditions

- None

4.4.5.2.1.1 Test Sequence N°1 – Normal Case: Authorised MNO call processed, and authorized notifications sent

Test Sequence Purpose

To ensure that when the authorisation is set by the Operator whose Profile currently has the Fall-Back Attribute set, another Operator can set the Fall-Back Attribute on its own Profile, implying it unsets it from initial Operator's Profile.
To ensure also that depending on the authorisation set by both Operators, the M2M SP receives or not the notification that the Fall-Back Attribute has been set or unset.

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result + comment	REQ
Set authorisations				
1	SM-DP-S → SM-SR-UT	`SEND_REQ( ES3-SetPLMA, #PLMA_MNO1_FOR_MNO2_RPS, #MNO1_ID_RPS )`	*(to allow MNO2 to "unset" the Fall-Back Attribute on Profile1)*	
2	SM-SR-UT → SM-DP-S	Send the `ES3-SetPLMA` response	4- The Status is equal to #SUCCESS 5- The `MnoId` parameter is equal to `#MNO1_ID_RPS`	
3	SM-SR-UT → MNO2-S	Send the `ES4-HandleSetPLMANotification` Notification	1- The `Plma` parameter is equal to #PLMA_MNO1_FOR_MNO2_RPS 2- The completion timestamp is present	

Step	Direction	Sequence / Description	Expected result + comment	REQ
4	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_M2MSP1_RPS, #MNO1_ID_RPS)	*(to allow M2MSP1 to receive Fall-Back Attribute notification unset on Profile1)*	
5	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA Response	1- The Status is equal to #SUCCESS 2- The MnoId parameter is equal to #MNO1_ID_RPS	
6	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO1_FOR_M2MSP1_RPS 2- The completion timestamp is present	
Now execute the command				
7	MNO2-S → SM-SR-UT	SEND_REQ(ES4-SetFallBackAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		
8	Execute sub-sequence 4.4.1.1 Set Fall-Back Attribute from SM-SR-UT			
9	SM-SR-UT → MNO2-S	Send the ES4-SetFallBackAttribute, response	The Status is equal to #SUCCESS	PROC_REQ_3.27_1 PROC_REQ_3.27_2
Check notifications, and verification of state updated in EIS				
10	SM-SR-UT → SM-DP-S	Send the ES3-HandleProfileFallBackAttributeUnsetNotification Notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- The completion timestamp is present 4- The MnoId parameter is equal to #MNO1_ID_RPS 5- The ProfileType parameter is equal to #PROFILE_TYPE1	PF_REQ_5.4.28

Step	Direction	Sequence / Description	Expected result + comment	REQ
11	SM-SR-UT → M2MSP1-S	Send the ES4-HandleProfileFallBackAttributeUnsetNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- The completion timestamp is present 4- The <rps:ProfileType> is present and equal to #PROFILE_TYPE1	PF_REQ_5.5.23
12	SM-SR-UT → M2MSP1-S	Check that M2MSP1-S does not receive notification ES4-HandleProfileFallBackAttributeSetNotification after 1mn		
13	MNO2-S → SM-SR-UT	SEND_REQ(ES4-getEIS, #VIRTUAL_EID_RPS)		
14	SM-SR-UT → MNO2-S	Send the ES4- GetEIS response	1- The ProfileInfo with #ICCID2_RPS has its FallbackAttribute true 2- The ProfileInfo with #ICCID1_RPS has its FallbackAttribute false	PROC_REQ_3.27_1, PF_REQ_5.5.21
Note: Steps 9-10-11 can occur in any order				

4.4.5.2.1.2 Test Sequence N°2 – Normal Case: Authorised call by M2M SP processed, and notifications sent

Initial Conditions

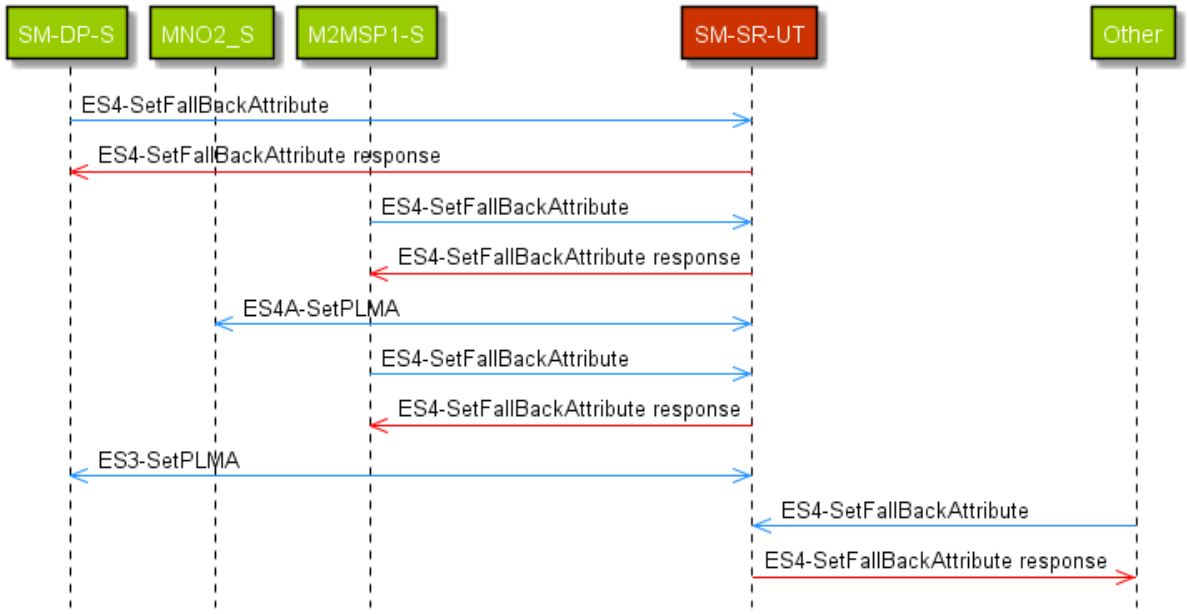
- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO2-S → SM-SR-UT	SEND_REQ(ES4A-SetPLMA, #PLMA_MNO2_FOR_M2MSP1_RPS)	*(to allow M2MSP1 to set the Fall-Back Attribute on Profile2)*	
2	SM-SR-UT → MNO2-S	Send the ES4A-SetPLMA response	The Status is equal to #SUCCESS	

Step	Direction	Sequence / Description	Expected result	REQ
3	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO2_FOR_M2MSP1_RPS 2- The completion timestamp is present	
4	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA,#PLMA_MNO1_FOR_M2MSP1_RPS, #MNO1_ID_RPS)	*(to allow M2MSP1 to "unset" the Fall-Back Attribute on Profile1, and to receive notification of it).*	
5	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA response	The Status is equal to #SUCCESS	
6	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO1_FOR_M2MSP1_RPS 2- The completion timestamp is present	
Now execute the command				
7	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-SetFallBackAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		
	Execute sub-sequence 4.4.1.1 Set Fall-Back Attribute from SM-SR-UT			
8	SM-SR-UT → M2MSP1-S	Send the ES4-SetFallBackAttribute response	The Status is equal to #SUCCESS	PROC_REQ_3.29_1 PF_REQ_5.5.21
Check notifications, and verification of state updated in EIS				
9	SM-SR-UT → SM-DP-S	Send the ES3-HandleProfileFallBackAttributeUnsetNotification Notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- The completion timestamp is present 4- The MnoId parameter is equal to #MNO1_ID_RPS 5- The ProfileType parameter is equal to #PROF_TYPE1_RPS	PROC_REQ_3.29_1 PF_REQ_5.4.28

Step	Direction	Sequence / Description	Expected result	REQ
10	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileFallBackAttributeSetNotification Notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is present 4- The MnoId parameter is equal to #MNO2_ID_RPS 5- The ProfileType parameter is equal to #PROF_TYPE2_RPS	PROC_REQ_3.29_1 PF_REQ_5.5.22
11	SM-SR-UT → M2MSP1-S	Send the ES4-HandleProfileFallBackAttributeUnsetNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- The completion timestamp is present 4- The <rps:ProfileType> is present and equal to #PROFILE_TYPE1	PROC_REQ_3.29_1 PF_REQ_5.5.23
12	SM-SR-UT → M2MSP1-S	Check that M2MSP1-S does not receive notification ES4-HandleProfileFallBackAttributeSetNotification after 1mn		PROC_REQ_3.29_1 PF_REQ_5.5.22
13	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-getEIS, #VIRTUAL_EID_RPS)		
14	SM-SR-UT → M2MSP1-S	Send the ES4- GetEIS response	1- The ProfileInfo with #ICCID2_RPS has its FallbackAttribute true 2- The ProfileInfo with #ICCID1_RPS has its FallbackAttribute false	PROC_REQ_3.29_1
Note: Steps 8-9-10-11 can occur in any order				

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "SM-DP-S" as OP1 #99CC00
participant "MNO2_S" as OP2 #99CC00
participant "M2MSP1-S" as SP #99CC00
participant "SM-SR-UT" as SR #CC3300
participant Other #99CC00
OP1->>SR: ES4-SetFallBackAttribute
SR-[#red]>>OP1: ES4-SetFallBackAttribute response

SP->>SR: ES4-SetFallBackAttribute
SR-[#red]>>SP: ES4-SetFallBackAttribute response

OP2<<->>SR: ES4A-SetPLMA

SP->>SR: ES4-SetFallBackAttribute
SR-[#red]>>SP: ES4-SetFallBackAttribute response

OP1<<->>SR: ES3-SetPLMA

Other->>SR: ES4-SetFallBackAttribute
SR-[#red]>>Other: ES4-SetFallBackAttribute response

@enduml
```



4.4.6 ES4 (MNO – SM-SR and M2MSP – SM-SR): SetEmergencyProfileAttribute authorized

4.4.6.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.25_1, PROC_REQ_3.26_1, PF_REQ_5.4.24, PF_REQ_5.5.18, PF_REQ_5.5.19

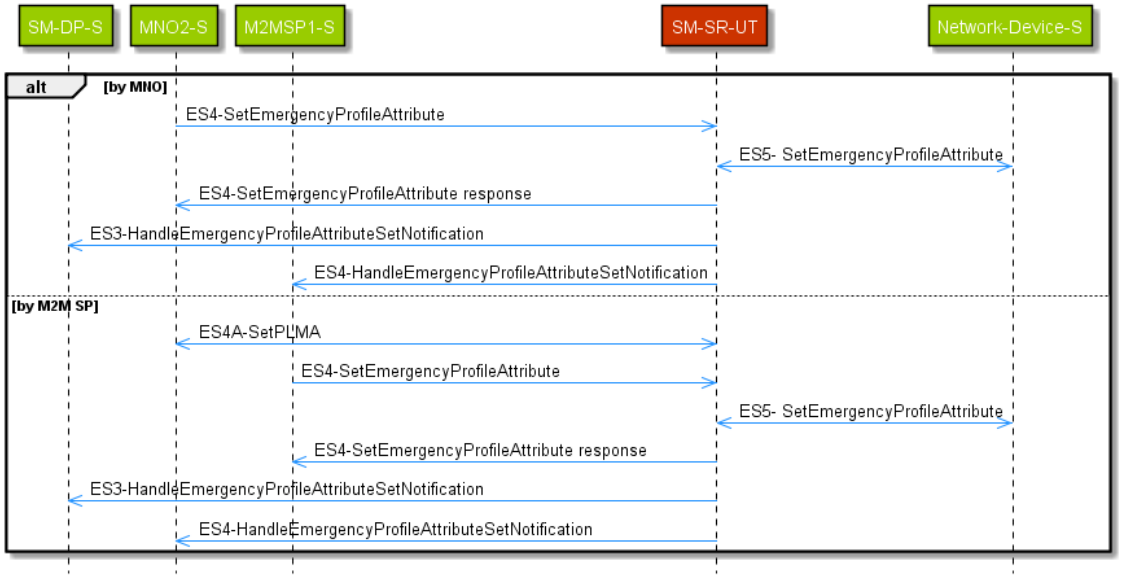
4.4.6.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #MNO2_S_ACCESSPOINT well known to the SM-SR-UT
- A direct connection exists between the MNO2-S and the SM-SR-UT
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- #M2MSP2_S_ID and # M2MSP2_S_ACCESSPOINT well known to the SM-SR-UT
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS
- No PLMA is granted by MNO1 nor MNO2 on any Profile Type

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "SM-DP-S" as OP1 #99CC00
participant "MNO2-S" as OP2 #99CC00
participant "M2MSP1-S" as SP #99CC00
participant "SM-SR-UT" as SR #CC3300
participant "Network-Device-S" as eUICC #99CC00

alt by MNO
OP2->>SR: ES4-SetEmergencyProfileAttribute
SR<<->>eUICC: ES5- SetEmergencyProfileAttribute
SR->>OP2: ES4-SetEmergencyProfileAttribute response

SR->>OP1: ES3-HandleEmergencyProfileAttributeSetNotification
SR->>SP: ES4-HandleEmergencyProfileAttributeSetNotification

else by M2M SP
OP2<<->>SR: ES4A-SetPLMA

SP->>SR: ES4-SetEmergencyProfileAttribute
SR<<->>eUICC: ES5- SetEmergencyProfileAttribute
SR->>SP: ES4-SetEmergencyProfileAttribute response

SR->>OP1: ES3-HandleEmergencyProfileAttributeSetNotification
SR->>OP2: ES4-HandleEmergencyProfileAttributeSetNotification
End

@enduml
```



4.4.6.2.1 TC.ES4.SEPA.2: SetEmergencyProfileAttribute authorised

Test Purpose

To ensure an Operator or M2M SP can set the Emergency Profile Attribute if the appropriate authorisations are granted.

Referenced Requirements

- PROC_REQ_3.25_1, PROC_REQ_3.26_1, PF_REQ_5.4.24, PF_REQ_5.5.18, PF_REQ_5.5.19

Initial Conditions

- None

4.4.6.2.1.1 Test Sequence N°1 – Normal Case: MNO call processed, and authorized notifications sent

Test Sequence Purpose

To ensure that when no Emergency Profile exists yet on the eUICC, an Operator can set the Emergency Profile Attribute on its own Profile, and all other Operators who have a Profile on the same eUICC receive a notification, and the M2M SP receives or not the notification.

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result + comment	REQ
Configure authorisations				
1	MNO2-S → SM-SR-UT	`SEND_REQ( ES4A-SetPLMA, #PLMA_MNO2_FOR_M2MSP1_RPS, )`	*(to allow M2MSP1 to receive notification that the Emergency Profile Attribute has been set)*	
2	SM-SR-UT → MNO2-S	Send the `ES4A-SetPLMA` response	The Status is equal to `#SUCCESS`	
3	SM-SR-UT → M2MSP1-S	Send the `ES4-HandleSetPLMANotification` Notification	1- The `Plma` parameter is equal to #PLMA_MNO2_FOR_M2MSP1_RPS 2- The completion timestamp is present	
Now execute the command				
4	MNO2-S → SM-SR-UT	`SEND_REQ( ES4-SetEmergencyProfileAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS )`		
5	Execute sub-sequence 4.4.1.4 Set Emergency profile Attribute from SM-SR-UT			

Step	Direction	Sequence / Description	Expected result + comment	REQ
6	SM-SR-UT → MNO2-S	Send the ES4-SetEmergencyProfileAttribute response	The Status is equal to #SUCCESS	PROC_REQ_3.25_1 PF_REQ_5.5.18
Check notifications, and verification of state updated in EIS				
7	SM-SR-UT → SM-DP-S	Send the ES3-HandleEmergencyProfileAttribute SetNotification Notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is not present, or it is present and equal to #ICCID2_RPS 3- The completion timestamp is present 4- The MnoId parameter is equal to #MNO_ID_RPS	PF_REQ_5.4.24
8	SM-SR-UT → M2MSP1-S	Send the ES4-HandleEmergencyProfileAttribute SetNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is present	PF_REQ_5.5.19
9	MNO2-S → SM-SR-UT	SEND_REQ(ES4-getEIS, #VIRTUAL_EID_RPS)		
10	SM-SR-UT → MNO2-S	Send the ES4-GetEIS response	The EIS contains an AdditionalProperty with the key equal to "gsma.ESIM.EmergencyProfileAttribute.AID", and the value equal to #ISP_P_AID3	PROC_REQ_3.25_1, PF_REQ_5.5.18
Note: Steps 6-7-8 can occur in any order				

4.4.6.2.1.2 Test Sequence N°2 – Normal Case: Authorised call by M2M SP processed, and notifications sent

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO2-S → SM-SR-UT	SEND_REQ(ES4A-SetPLMA, #PLMA_MNO2_FOR_M2MSP1_RPS,)	*(to allow M2MSP1 to set the Emergency Profile Attribute on Profile2)*	
2	SM-SR-UT → MNO2-S	Send the ES4A-SetPLMA response	The Status is equal to #SUCCESS	
3	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification Notification	1- The Plma parameter is equal to #PLMA_MNO2_FOR_M2MSP1_RPS 2- The completion timestamp is present	
Now execute the command				
4	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-SetEmergencyProfileAttribute, #VIRTUAL_EID_RPS, #ICCID2_RPS)		
5	Execute sub-sequence 4.4.1.4 Set Emergency Profile Attribute from SM-SR-UT			
6	SM-SR-UT → M2MSP1-S	Send the ES4-SetEmergencyProfileAttribute response	The Status is equal to #SUCCESS	PROC_REQ_3.26_1 PF_REQ_5.5.18
Check notifications, and verification of state updated in EIS				
7	SM-SR-UT → SM-DP-S	Send the ES3-HandleEmergencyProfileAttributeSetNotification Notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is not present, or it is present and equal to #ICCID2_RPS 3- The completion timestamp is present 4- The MnoId parameter is equal to #MNO1__ID_RPS	PROC_REQ_3.26_1 PF_REQ_5.4.24
8	SM-SR-UT → MNO2-S	Send the ES4-HandleEmergencyProfileAttributeSetNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is present	PROC_REQ_3.26_1 PF_REQ_5.5.19

Step	Direction	Sequence / Description	Expected result	REQ
9	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-getEIS, #VIRTUAL_EID_RPS)		
10	SM-SR-UT → M2MSP1-S	Send the ES4-GetEIS response	The EIS contains an AdditionalProperty with the key equal to "gsma.ESIM.EmergencyProfileAttribute.AID", and the value equal to #ISP_P_AID3	PROC_REQ_3.26_1 PF_REQ_5.5.18
Note: Steps 6-7-8 can occur in any order				

4.4.7 ES4 (M2M SP - SM-SR): EnableProfile by M2M SP

4.4.7.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.17.1, PROC_REQ_3.20.2, PF_REQ22, PF_REQ24, PF_REQ27, PF_REQ_5.4.16, PF_REQ_5.4.20, PF_REQ_5.4.21, PF_REQ_5.5.16

4.4.7.2 Test Cases

General Initial Conditions

- #MNO1_S_ID well known to the SM-SR-UT
- #MNO2_S_ID and #MNO2_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
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- No PLMA is granted by MNO1 on any Profile Type
- No ONC is configured for MNO1
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS

4.4.7.2.1 TC.ES4.EPM2MSP.2: EnableProfile by M2M SP

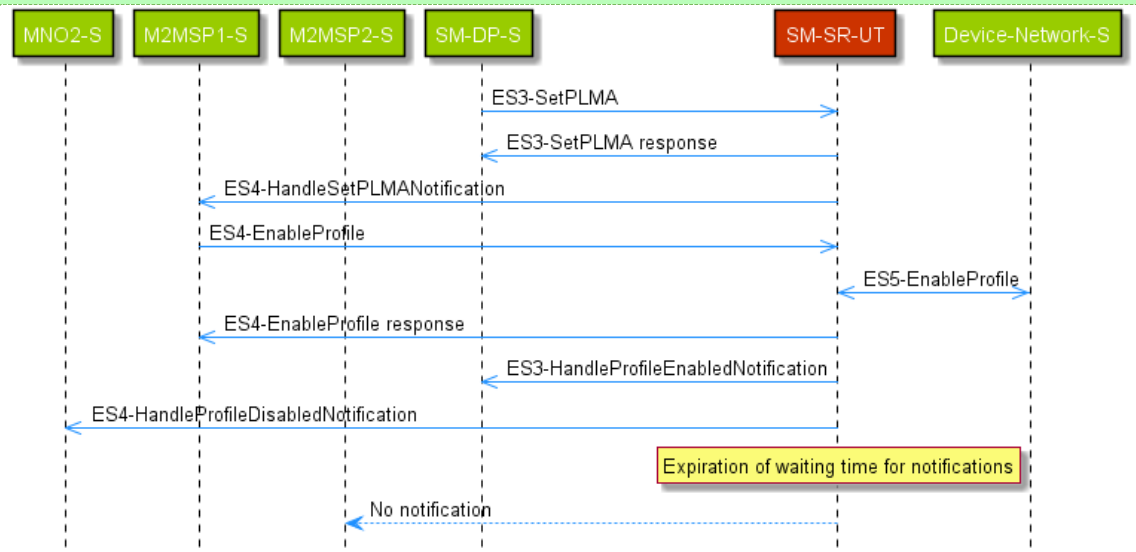
Test Purpose

To ensure that a M2M SP is able to Enable a Profile as soon as it is authorized by the MNO owning the profile.

To verify that notifications are sent to relevant parties on Profile status change.

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant "MNO2-S" as OP2 #99CC00
participant "M2MSP1-S" as SP1 #99CC00
participant "M2MSP2-S" as SP2 #99CC00
participant "SM-DP-S" as OP1 #99CC00
participant "SM-SR-UT" as SR #CC3300
participant "Device-Network-S" as eUICC #99CC00

OP1->>SR: ES3-SetPLMA
SR->>OP1: ES3-SetPLMA response

SR->>SP1: ES4-HandleSetPLMANotification

SP1->>SR: ES4-EnableProfile

SR<<->>eUICC: ES5-EnableProfile

SR->>SP1: ES4-EnableProfile response

SR->>OP1: ES3-HandleProfileEnabledNotification

SR->>OP2: ES4-HandleProfileDisabledNotification

rnote over SR
      Expiration of waiting time for notifications
End rnote

SR-->SP2: No notification


@enduml
```



Referenced Requirements

- PROC_REQ_3.17.1, PROC_REQ_3.20.2, PF_REQ22, PF_REQ24, PF_REQ27, PF_REQ_5.4.16, PF_REQ_5.4.20, PF_REQ_5.4.21, PF_REQ_5.5.16

Initial Conditions

- None

4.4.7.2.1.1 Test Sequence N°1 – Normal Case: PLMA for M2M SP and no ONC, Enable Profile by M2M SP

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_M2MSP1_RPS, #MNO1_S_ID)		PROC_REQ_3.20.2, PF_REQ_5.4.16
2	SM-SR-UT → SM-DP-S	Send the ES3-SetPLMA response	The Status is equal to #SUCCESS	
3	SM-SR-UT → M2MSP1-S	Send the ES4-HandleSetPLMANotification notification	1- The <Plma> parameter is equal to #PLMA_RPS 2- The completion timestamp is present	PROC_REQ_3.20.2, PF_REQ_5.5.16
4	M2MSP1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #VIRTUAL_EID_RPS, #ICCID1_RPS)		PROC_REQ_3.17.1, PF_REQ24
5	See sub-sequence 4.4.1.2 EnableProfile from SM-SR-UT			
6	SM-SR-UT → M2MSP1-S	Send the ES4-EnableProfile response	The Status is equal to #SUCCESS	
7	SM-SR-UT → SM-DP-S	Send the ES3-HandleProfileEnabledNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID1_RPS 3- The completion timestamp is present 4- If the response is sent using SOAP protocol, verify that the <wsa:To> contains the query parameter MnoId=#MNO1_S_ID 5- If the response is sent using SOAP protocol, verify that the <wsa:MessageId> contains the query parameter ProfileType=#PROFILE_TYPE1	PROC_REQ_3.17.1, PF_REQ22

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-UT → MNO2-S	Send the ES4-HandleProfileDisabledNotification notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is present	PROC_REQ_3.17.1, PF_REQ27
9	Check that M2MSP2 does not receive notification after 1mn			

4.4.8 ES4 (M2M SP - SM-SR): EnableProfile by M2M SP with ONC set

4.4.8.1 Conformance Requirements

References

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

Requirements

- PROC_REQ_3.17.1, PROC_REQ_3.20.2, PROC_REQ_3.21.2, PF_REQ24, PF_REQ27, PF_REQ_5.4.16

4.4.8.2 Test Cases

General Initial Conditions

- #MNO1_S_ID well known to the SM-SR-UT
- #MNO2_S_ID and #MNO2_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
- #M2MSP1_S_ID and # M2MSP1_S_ACCESSPOINT well known to the SM-SR-UT
- No PLMA is granted by MNO1 on any Profile Type
- No ONC is configured for MNO1
- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_ES1_RPS

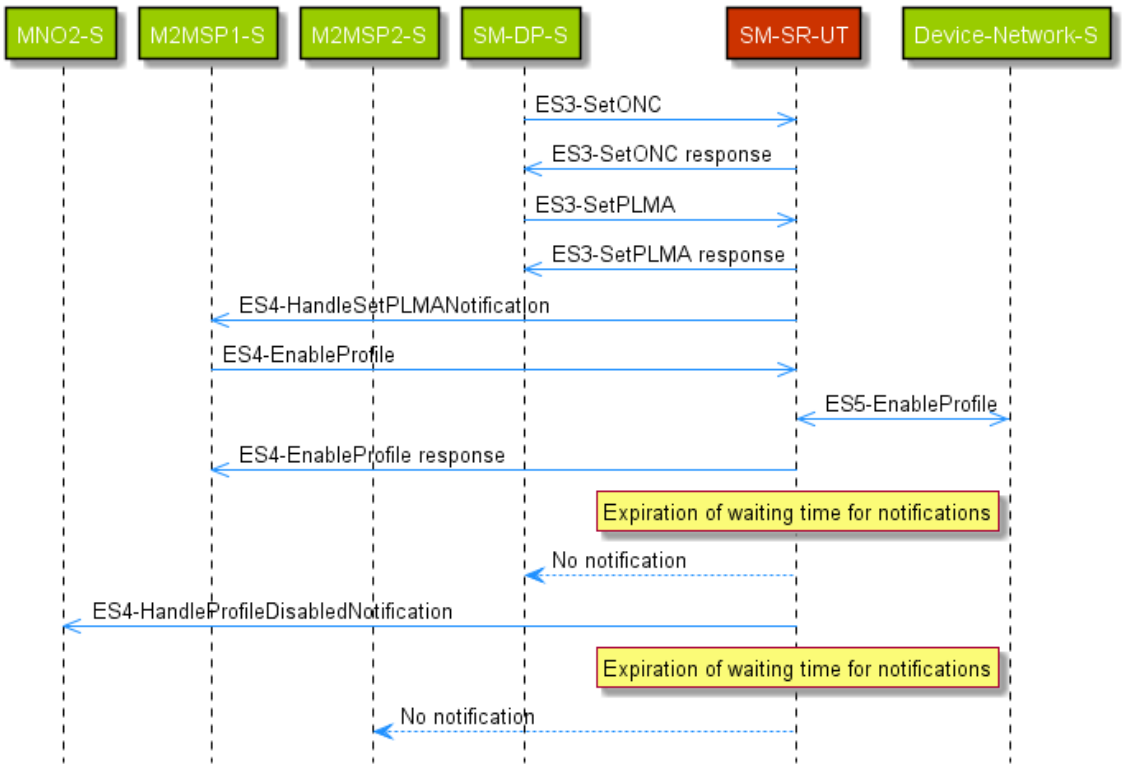
4.4.8.2.1 TC.ES4.EPM2MSP.3: EnableProfile by M2M SP with ONC

Test Purpose

- *To ensure that a M2M SP is able to Enable a Profile as soon as it is authorized by the MNO owning the profile.*
- *To verify that notifications are sent to relevant parties on Profile status change.*
- *To verify that Operators are not receiving notifications on Profile status change when explicitly requested via ONC configuration.*

Test Environment

```
@startuml
skinparam sequence {
     ArrowColor DodgerBlue
     LifeLineBorderColor Black

     ParticipantBorderColor Black
     ParticipantFontColor White
}
hide footbox
participant "MNO2-S" as OP2 #99CC00
participant "M2MSP1-S" as SP1 #99CC00
participant "M2MSP2-S" as SP2 #99CC00
participant "SM-DP-S" as OP1 #99CC00
participant "SM-SR-UT" as SR #CC3300
participant "Device-Network-S" as eUICC #99CC00

OP1->>SR: ES3-SetONC
SR->>OP1: ES3-SetONC response

OP1->>SR: ES3-SetPLMA
SR->>OP1: ES3-SetPLMA response

SR->>SP1: ES4-HandleSetPLMANotification

SP1->>SR: ES4-EnableProfile

SR<<->>eUICC: ES5-EnableProfile

SR->>SP1: ES4-EnableProfile response

rnote over SR
     Expiration of waiting time for notifications
End rnote

SR-->OP1: No notification

SR->>OP2: ES4-HandleProfileDisabledNotification

rnote over SR
     Expiration of waiting time for notifications
End rnote

SR-->SP2: No notification


@enduml
```



Referenced Requirements

- PROC_REQ_3.17.1, PROC_REQ_3.20.2, PROC_REQ_3.21.2, PF_REQ24, PF_REQ27, PF_REQ_5.4.16, PF_REQ_5.4.21, PF_REQ_5.5.16

Initial Conditions

- None

4.4.8.2.1.1 Test Sequence N°1 – Normal Case: PLMA for M2M SP and ONC set, Enable Profile by M2M SP

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetONC, #ONC_MNO1_RPS, #MNO1_S_ID)		PROC_REQ_3.21.2, PF_REQ_5.4.21
2	SM-SR-UT → SM-DP-S	Send the ES3-SetONC response	The Status is equal to #SUCCESS	
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SetPLMA, #PLMA_MNO1_FOR_M2MSP1_RPS, #MNO1_S_ID)		PROC_REQ_3.20.2, PF_REQ_5.4.16

Step	Direction	Sequence / Description	Expected result	REQ
4	SM-SR-UT → SM-DP-S	Send the `ES3-SetPLMA` response	The Status is equal to #SUCCESS	
5	SM-SR-UT → M2MSP1-S	Send the `ES4-HandleSetPLMANotification` notification	1- The <Plma> parameter is equal to #PLMA_RPS 2- The completion timestamp is present	PROC_REQ_3.20.2, PF_REQ_5.5.16

6	M2MSP1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-EnableProfile,`
		`#VIRTUAL_EID_RPS,`
		`#ICCID1_RPS`
		`)`
				PROC_REQ_3.17.1, PF_REQ24

7	See sub-sequence 4.4.1.2 EnableProfile from SM-SR-UT			
8	SM-SR-UT → M2MSP1-S	Send the `ES4-EnableProfile` response	The Status is equal to #SUCCESS	
9	Check that SM-DP does not receive notification after 1mn			
10	SM-SR-UT → MNO2-S	Send the `ES4-HandleProfileDisabledNotification` Notification	1- The EID parameter is equal to #VIRTUAL_EID_RPS 2- The ICCID is equal to #ICCID2_RPS 3- The completion timestamp is present	PROC_REQ_3.17.1, PF_REQ27
11	Check that M2MSP2 does not receive notification after 1mn			

4.4.9 ES4 (MNO – SM-SR): SMSRChange

4.4.9.1 Conformance Requirements

References

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

Requirements

- EUICC_REQ36, EUICC_REQ39, PROC_REQ13_2, PROC_REQ13_3

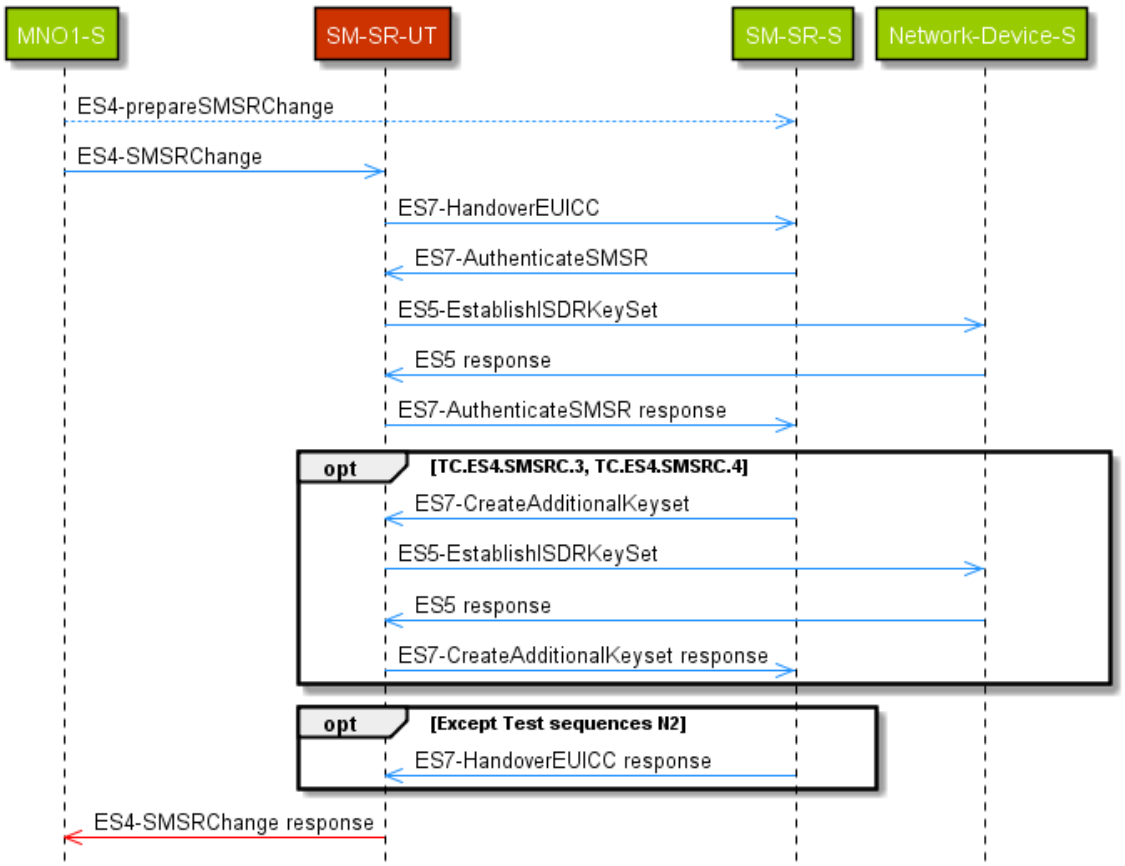
4.4.9.2 Test Cases

General Initial Conditions

- All necessary settings have been initialized on SM-SR-UT to accept the SM-SR change (i.e. business agreement…)

Test Environment

```
@startuml
skinparam sequence {
     ArrowColor DodgerBlue
     LifeLineBorderColor Black

     ParticipantBorderColor Black
     ParticipantFontColor White
}
hide footbox
participant OP1 as "MNO1-S" #99CC00
participant SR1 as "SM-SR-UT" #CC3300
participant SR2 as "SM-SR-S" #99CC00
participant NDS as "Network-Device-S" #99CC00

OP1-->>SR2: ES4-prepareSMSRChange
OP1->>SR1: ES4-SMSRChange
SR1->>SR2: ES7-HandoverEUICC

SR2->>SR1: ES7-AuthenticateSMSR
SR1->>NDS: ES5-EstablishISDRKeySet
NDS->>SR1: ES5 response
SR1->>SR2: ES7-AuthenticateSMSR response

Opt TC.ES4.SMSRC.3, TC.ES4.SMSRC.4
SR2->>SR1: ES7-CreateAdditionalKeyset
SR1->>NDS: ES5-EstablishISDRKeySet
NDS->>SR1: ES5 response
SR1->>SR2: ES7-CreateAdditionalKeyset response
End

Opt Except Test sequences N2
SR2->>SR1: ES7-HandoverEUICC response
End
SR1-[#red]>>OP1: ES4-SMSRChange response

@enduml
```



Note that the function ES4-PrepareSMSRChange SHALL NOT be performed by the simulators (in the schema above, they are only informative messages).

4.4.9.2.1 TC.ES4.SMSRC.2: SMSRChange fails in case Handover fails or expires after authenticate SM-SR success

Test Purpose

To ensure the method SMSRChange fails if the AuthenticateSM-SR has been performed but the handover fails or expires or doesn't complete.

Referenced Requirements

- EUICC_REQ36, EUICC_REQ39, PROC_REQ13_2

Initial Conditions

- The variable `{SM_SR_ID_RPS}` SHALL be set to `#SM_SR_UT_ID_RPS`

4.4.9.2.1.1 Test Sequence N°1 – ES7.HandoverEUICC expires

Initial Conditions

- None

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, #SHORT_VP_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC Request	1- The EIS is equal to #EIS_ES7_RPS 2- The value of the ValidityPeriod is lower or equal to #SHORT_VALIDITY_PERIOD	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #VIRTUAL_EID_RPS, #VALID_SM_SR_CERTIFICATE)		
4	Execute sub-sequence 4.4.1.5 First part of ISD-R Keyset Establishment from SM-SR-UT			
5	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR response	The Status is equal to #SUCCESS	
6	SM-SR-S→ SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #EXPIRED, #SC_FUNCTION, #RC_TTL_EXPIRED)		
7	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_FUNCTION 3- The Reason code is equal to #RC_TTL_EXPIRED	EUICC_REQ36 PROC_REQ13_2

4.4.9.2.1.2 Test Sequence N°2 – ES7.HandoverEUICC doesn't complete

Initial Conditions

- None

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, #SHORT_VP_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC Request	The EIS is equal to #EIS_ES7_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #VIRTUAL_EID_RPS, #VALID_SM_SR_CERTIFICATE)		
4	Execute sub-sequence 4.4.1.5 First part of ISD-R Keyset Establishment from SM-SR-UT			
5	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR response	The Status is equal to #SUCCESS	
6	*Wait at least the number of seconds specified in #SHORT_VALIDITY_PERIOD* *Do not send any request or response from SM-SR-S*			
7	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_FUNCTION 3- The Reason code is equal to #RC_TTL_EXPIRED	EUICC_REQ36 PROC_REQ13_2

4.4.9.2.2 TC.ES4.SMSRC.3: SMSRChange fails in case Handover fails after CreateAdditionalKeyset success

Test Purpose

To ensure the method SMSRChange fails if the AuthenticateSM-SR has been performed but the handover doesn't complete

Referenced Requirements

- EUICC_REQ36, EUICC_REQ39, PROC_REQ13_2
- **Initial Conditions**The variable {SM_SR_ID_RPS} SHALL be set to #SM_SR_UT_ID_RPS

4.4.9.2.2.1 Test Sequence N°1 – ES7.HandoverEUICC fails

Test Sequence Purpose

To ensure that when SM-SR2 declares the ES7.Handover failed after createAdditionalKeyset response, the SM-SR1 (here SM-SR-UT) will declare the overall SM-SR Change failed.

Initial Conditions

- None

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, #SHORT_VP_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC Request	The EIS is equal to #EIS_ES7_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #VIRTUAL_EID_RPS, #VALID_SM_SR_CERTIFICATE)		
4	*Execute sub-sequence 4.4.1.5 First part of ISD-R Keyset Establishment from SM-SR-UT*			
5	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR response	The Status is equal to #SUCCESS	
6	SM-SR-S → SM-SR-UT	SEND_REQ(ES7-CreateAdditionalKeyset, #VIRTUAL_EID_RPS, #KEY_VERSION_RPS, #INIT_SEQ_COUNTER_RPS, #ECC_KEY_LENGTH_RPS, #SC3_NO_DR_RPS, #EPHEMERAL_PK_RPS, #SIGNATURE_RPS) Note: no <HostId> is passed		
7	*Execute sub-sequence 4.4.1.6 Second part of ISD-R Keyset Establishment from SM-SR-UT*			

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 Receipt is equal to the {RECEIPT} value returned by the Network-Device-S	
9	SM-SR-S → SM-SR-UT	SEND_ERROR_RESP(ES7-HandoverEUICC, #FAILED, #SC_CERT_REQ, #R C_VERIFICATION_FAILED)		
10	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_CERT_REQ 3- The Reason code is equal to #RC_VERIFICATION_FAILED	PROC_REQ13_2

4.4.9.2.3 TC.ES4.SMSRC.4: SMSRChange expires in case Handover doesn't complete after CreateAdditionalKeyset success

Test Purpose

To ensure that if the handover doesn't complete after a successful createAdditionalKeyset response, the SM-SR1 declares the overall SM-SR Change expired

Referenced Requirements

- EUICC_REQ36, EUICC_REQ39, PROC_REQ13_3

Initial Conditions

- The variable {SM_SR_ID_RPS} SHALL be set to #SM_SR_UT_ID_RPS

4.4.9.2.3.1 Test Sequence N°1 – ES7.HandoverEUICC doesn't complete

Initial Conditions

- None

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, #SHORT_VP_RPS)		
2	SM-SR-UT→ SM-SR-S	Send the ES7-HandoverEUICC Request	The EIS is equal to #EIS_ES7_RPS	EUICC_REQ36, EUICC_REQ39
3	SM-SR-S→ SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #VIRTUAL_EID_RPS, #VALID_SM_SR_CERTIFICATE)		
4	*Execute sub-sequence 4.4.1.5 First part of ISD-R Keyset Establishment from SM-SR-UT*			
5	SM-SR-UT→ SM-SR-S	Send the ES7-AuthenticateSMSR response	The Status is equal to #SUCCESS	
6	SM-SR-S → SM-SR-UT	SEND_REQ(ES7-CreateAdditionalKeyset, #VIRTUAL_EID_RPS, #KEY_VERSION_RPS, #INIT_SEQ_COUNTER_RPS, #ECC_KEY_LENGTH_RPS, #SC3_NO_DR_RPS, #EPHEMERAL_PK_RPS, #SIGNATURE_RPS) Note: no <HostId> is passed		
7	*Execute sub-sequence 4.4.1.6 Second part of ISD-R Keyset Establishment from SM-SR-UT*			
8	SM-SR-UT→ SM-SR-S	Send the ES7-CreateAdditionalKeyset Response	1- The Status is equal to #SUCCESS 2- The Receipt is equal to {RECEIPT}	
9	*Wait at least the number of seconds specified in #SHORT_VALIDITY_PERIOD* *Do not send any request or response from SM-SR-S*			

Step	Direction	Sequence / Description	Expected result	REQ
10	SM-SR-UT → MNO1-S	Send the ES4-SMSRChange Response	1- The Status is equal to #EXPIRED 2- The Subject code is equal to #SC_FUNCTION 3- The Reason code is equal to #RC_TTL_EXPIRED	PROC_REQ13_3

4.4.10 ES5 (SM-SR – eUICC): CreateISDP

4.4.10.1 Conformance Requirements

References

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

Requirements

- PROC_REQ1_1, PROC_REQ1_2, PM_REQ16_1, PF_REQ3, PM_REQ16, EUICC_REQ50, EUICC_REQ51

4.4.10.2 Test Cases

General Initial Conditions

- #MNO1_S_ID well known to the SM-SR-UT
- #SM_DP_S_ID and #SM_DP_S_ACCESSPOINT well known to the SM-SR-UT

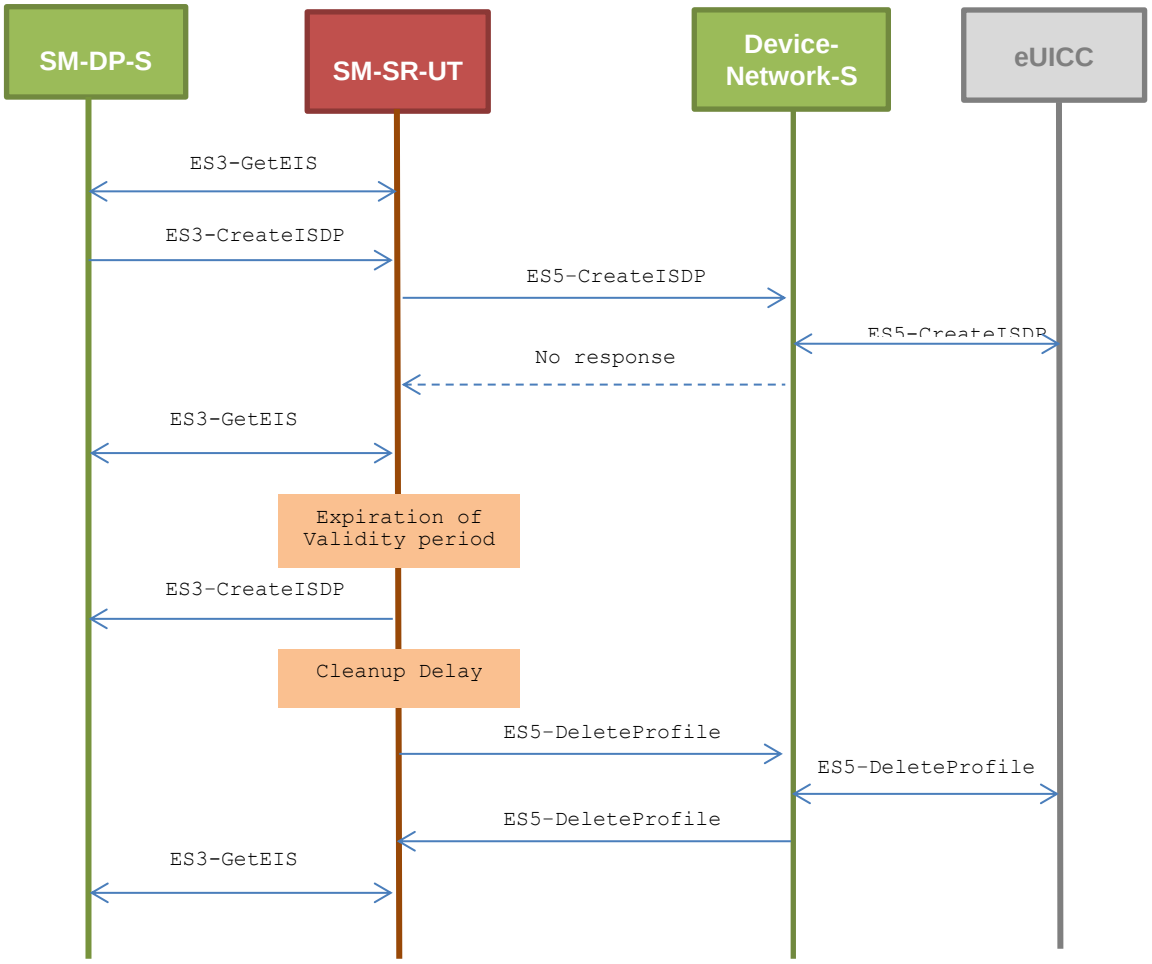
4.4.10.2.1 TC.ES5.CreateISDP.1: ISDP_Auto_Deletion

Test Purpose

To ensure that the ISD-P creation procedure is well implemented by the SM-SR. This test case proposes to verify the behavior of the SM-SR in case the procedure fails, in particular:

- *when the SM-SR does not receive a function execution response from the eUICC during the ISD-P creation, the SM-SR SHALL trigger the ES5.DeleteISDP function on the targeted ISD-P*

Test Environment



Referenced Requirements

- PROC_REQ1_1, PROC_REQ1_2

Initial Conditions

- None

4.4.10.2.1.1 Test Sequence N°1 – Error Case: ISD-P Creation fails due to disrupted connection

Initial Conditions

- The eUICC identified by `#EID` has been provisioned on the SM-SR-UT using the `#EIS_RPS`

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #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_RPS	
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-CreateISDP, #EID_RPS, #NEW_ICCID_RPS, #MNO1_ID_RPS, # NO_REQUIRED_MEM_RPS, #MORE_TODO_RPS) The validity period in the request is set to 320 seconds.		
4	Start a timer of 400 seconds, and proceed with steps 5 to 11 while the timer is running			
5	SM-SR-UT → Device-Network-S	ES5-CreateISDP function is received by the Device-Network-S over SMS, CAT_TP or HTTPs.	The ISD-P AID present in the INSTALL command is extracted by the test tool	
6	Device-Network-S → eUICC	The Device-Network-S transfers the OTA command to the eUICC.		
7	eUICC → Device-Network-S	The eUICC sends the OTA response to the Device-Network-S The Device-Network-S does not forward the response to the SM-SR-UT	[R_AB_009000] or [R_AF_009000] is returned by the eUICC	
While the timer (started in step 4) is running and the SM-SR is waiting for the eUICC response, the SM-DP-S sends an ES3-GetEIS message in order to verify that the Profile having a state “In-Creation” is not returned.				
8	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #EID_RPS)		
9	SM-SR-UT → SM-DP-S	Send the ES3-GetEIS Response	The response is equal to the one received in step 2.	PROC_REQ1_1
While the timer (started in step 4) is still running, the SM-SR may send some commands to the eUICC, using SMS, CAT-TP, or HTTP.				

Step	Direction	Sequence / Description	Expected result	REQ
10	SM-SR-UT → Device-Network-S	The SM-SR-UT sends some command script to the Device-Network-S (optional step) over SMS, CAT-TP, or HTTP	The Device-Network-S decrypts the transport message, and records the command script. This step being optional, if the SM-SR-UT sends nothing, the Device-Network-S simply records nothing.	PROC_REQ1_2
11	Device-Network-S→ eUICC	The Device-Network-S transfers the OTA command to the eUICC. Note: this step is conditional. If the SM-SR doesn't send commands at step 10, this step 11 can be skipped (see NOTE 1)		
12	eUICC→ Device-Network-S	The eUICC sends the OTA response to the Device-Network-S Note: this step is conditional. If the SM-SR doesn't send commands at step 10, this step 12 can be skipped (see NOTE 1)		
13	Device-Network-S→ SM-SR-UT	The Device-Network-S forwards the response to the SM-SR-UT Note: this step is conditional. If the SM-SR doesn't send commands at step 10, this step 13 can be skipped (see NOTE 1)	The Device-Network-S decrypts the transport message, and records the response script.	
The ES3-CreateISDP response is expected to be received before the expiration of the timer started at step 4, related to the validity Period.				
14	SM-SR-UT → SM-DP-S	Send the `ES3-CreateISDP` Response	The Status is equal to either #EXPIRED, #FAILED.	PM_REQ16
15	*Start a new timer of CLEANUP_DELAY. Wait until the timer expires, or the command* ES5-DeleteProfile *is triggered by the SM-SR-UT.* *The CLEANUP_DELAY SHALL be given by the SM-SR-UT to the Test Tool Provider.* While this timer is running, the SM-SR may send some commands to the eUICC, using SMS, CAT-TP, or HTTP.			
16	SM-SR-UT → Device-Network-S	The SM-SR-UT sends some command script to the Device-Network-S (optional step) over SMS, CAT-TP, or HTTP.	The Device-Network-S decrypts the transport message, and records the command script. This step being optional, if the SM-SR-UT sends nothing, the Device-Network-S simply records nothing.	PROC_REQ1_2
17	Device-Network-S→ eUICC	The Device-Network-S transfers the OTA command to the eUICC. Note: this step is conditional. If the SM-SR doesn't send commands at step 16, this step 17 can be skipped (see NOTE 1)		

Step	Direction	Sequence / Description	Expected result	REQ
18	eUICC→ Device-Network-S	The eUICC sends the OTA response to the Device-Network-S Note: this step is conditional. If the SM-SR doesn't send commands at step 16, this step 18 can be skipped (see NOTE 1)		
19	Device-Network-S→ SM-SR-UT	The Device-Network-S forwards the response to the SM-SR-UT Note: this step is conditional. If the SM-SR doesn't send commands at step 16, this step 19 can be skipped (see NOTE 1)	The Device-Network-S decrypts the transport message, and records the response script.	
20	Expiration of the timer related to CLEANUP_ DELAY			
21	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #EID_RPS)		
22	SM-SR-UT → SM-DP-S	Send the ES3-GetEIS Response	The response is equal to the one received in step 2 (meaning that from the outside view, the Profile has been automatically removed from the SM-SR's EIS database).	PROC_REQ1_ 1
23	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-CreateISDP, #EID_RPS, #NEW_ICCID_RPS, #MNO1_ID_RPS, # NO_REQUIRED_MEM_RPS, #MORE_TODO_RPS)		
Start loop. The SM-SR-UT SHALL exchange one command script/response script with the eUICC, and MAY exchange several other command scripts/response scripts with the eUICC. *The maximum number of iterations SHALL be given by the SM-SR-UT to the Test Tool Provider.*				
24	SM-SR-UT → Device-Network-S	The SM-SR-UT sends some command script to the Device-Network-S (optional step) over SMS, CAT-TP, or HTTP. The Device-Network-S does not transfer the OTA command to the eUICC.	The Device-Network-S decrypts the transport message, and records the command script.	

Step	Direction	Sequence / Description	Expected result	REQ
25	Device-Network-S→ eUICC	The Device-Network-S transfers the OTA command to the eUICC. Note: this step is conditional. If the SM-SR doesn't send commands at step 24, this step 25 can be skipped (see NOTE 1)		
26	eUICC→ Device-Network-S	The eUICC sends the OTA response to the Device-Network-S Note: this step is conditional. If the SM-SR doesn't send commands at step 24, this step 26 can be skipped (see NOTE 1)		
27	Device-Network-S→ SM-SR-UT	The Device-Network-S forwards the response to the SM-SR-UT Note: this step is conditional. If the SM-SR doesn't send commands at step 24, this step 27 can be skipped (see NOTE 1)	The Device-Network-S decrypts the transport message, and records the response script.	
End Loop				
28	Device-Network-S	Verify the commands optionally sent by the SM-SR-UT and recorded at steps 10, 16, and 24	Verify at least one of the commands is a DELETE command, and the ISD-P AID present in the DELETE command is the same as the one extracted in step 5. Other commands may be present before or after the DELETE command. If no command has been recorded at all, this is considered a failure	PROC_REQ1_2
29	Device-Network-S	Verify the responses optionally sent by the eUICC and recorded at steps 13, 19, and 27	Verify the response to the DELETE command indicates normal execution (SW=9000). If no response has been recorded at all, this is considered a failure	PROC_REQ1_2
30	Device-Network-S	Verify the commands sent by the SM-SR-UT and recorded at step 24	Verify at least one of the commands is a ES5.CreateISD-P command. If no command has been recorded at all, this is considered a failure	
31	Device-Network-S	Verify the response sent by the eUICC and recorded at step 27	Verify the response to the ES5.CreateISDP command indicates normal execution (SW=9000). If no response has been recorded at all, this is considered a failure	
32	SM-SR-UT → SM-DP-S	Send the `ES3-CreateISDP` Response	The status is equal to #SUCCESS	

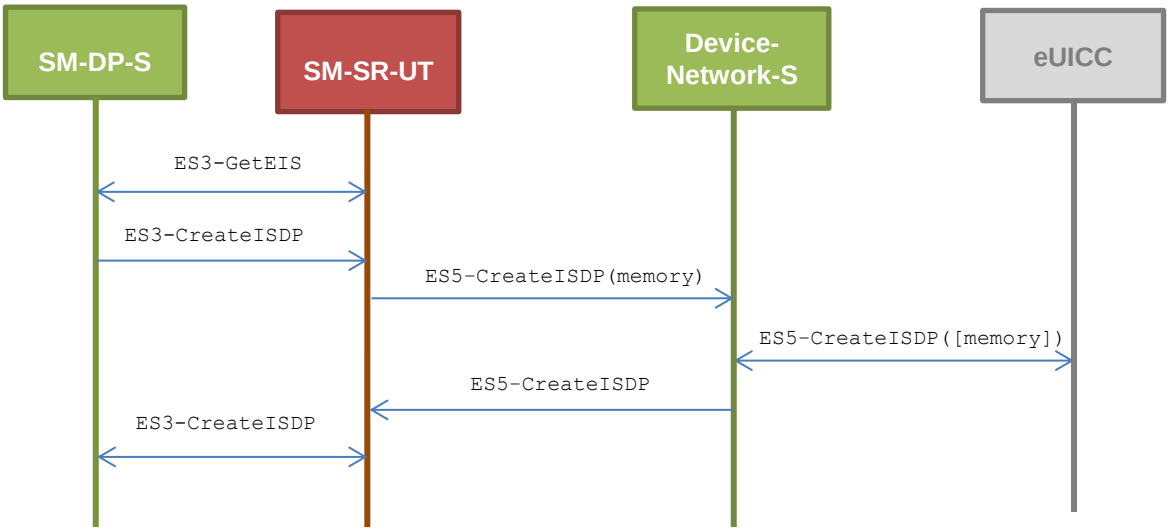
NOTE 1: the moment when the SM-SR sends the DELETE command is implementation-dependant. So steps 10, 16, and 24 are optional. In case the SM-SR does not send commands at those steps, the steps related to handling such commands and the corresponding responses do not need to be performed by the Device-Network- S. The expected commands and responses are still checked in steps 28 to 31 (causing a test failure if no command was sent at all).

4.4.10.2.2 TC.ES5.CreateISDP.2: Memory_Allocation

Test Purpose

To ensure that the memory management related to the ISD-P creation procedure is well implemented by the SM-SR. This test case proposes to verify that the "Cumulative Granted Non Volatile Memory" parameter is correctly set in the INSTALL command according to the "RequiredMemory" specified in the ES3-CreateISDP function.

Test Environment



Referenced Requirements

- PM_REQ16_1, PF_REQ3, PM_REQ16

Initial Conditions

- None

4.4.10.2.2.1 Test Sequence N°1 – Nominal Case: ISD-P Creation without required memory

Initial Conditions

- The eUICC identified by `#EID` has been provisioned on the SM-SR-UT using the `#EIS_RPS`

Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #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_RPS	
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-CreateISDP, #EID_RPS, #NEW_ICCID_RPS, #MNO1_ID_RPS, #NO_REQUIRED_MEM_RPS, #MORE_TODO_RPS)		
4	SM-SR-UT → Device-Network-S	ES5-CreateISDP function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	1- The ISD-P AID present in the INSTALL command is extracted by the test tool 2- Verify that there is no Cumulative Granted Non Volatile Memory parameter present in the INSTALL command (i.e. no tag 'EF'/'83')	PF_REQ3 PM_REQ16 PM_REQ16_1
5	Device-Network-S → eUICC	The Device-Network-S transfers the OTA command to the eUICC		
6	eUICC → Device-Network-S	The eUICC sends the OTA response to the Device-Network-S	[R_AB_009000] or [R_AF_009000] is returned by the eUICC	
7	Device-Network-S → SM-SR	The Device-Network-S transfers the OTA response to the SM-SR-UT		
8	SM-SR-UT → SM-DP-S	Send the ES3-CreateISDP Response	1- The Status is equal to #SUCCESS 2- The isd-p-aid is equal to the one extracted in step 4	

4.4.10.2.2.2 Test Sequence N°2 – Nominal Case: ISD-P Creation with required memory

Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_RPS

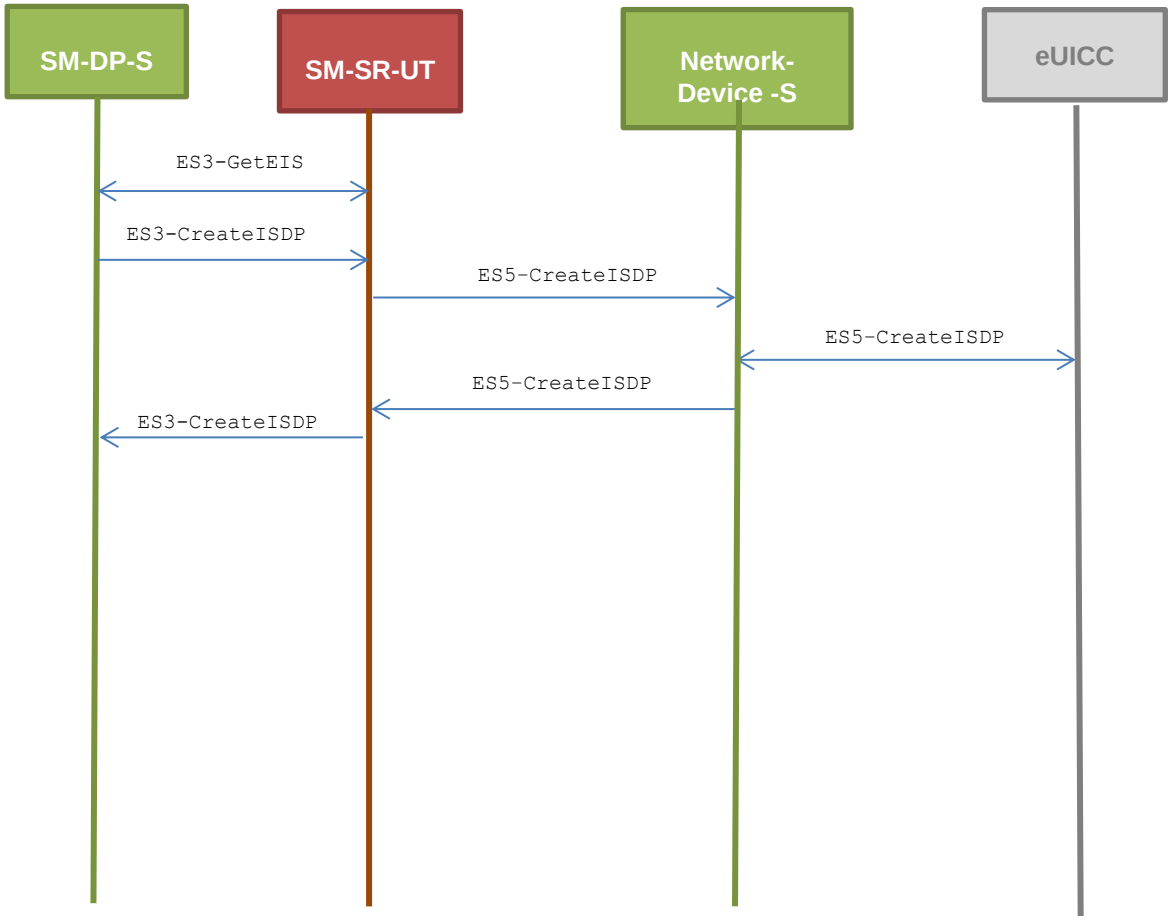
Step	Direction	Sequence / Description	Expected result	REQ
1	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-GetEIS, #EID_RPS)		
2	SM-SR-UT → SM-DP-S	Send the ES3-GetEIS response	1- The Status is equal to #SUCCESS 2- The EIS returned is equal to #EIS_RPS	
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-CreateISDP, #EID_RPS, #NEW_ICCID_RPS, #MNO1_ID_RPS, #SMALL_MEM_RPS, #MORE_TODO_RPS)		
4	SM-SR-UT → Device-Network-S	ES5-CreateISDP function is received by the Device-Network-S over SMS, CAT_TP or HTTPs	1- The ISD-P AID present in the INSTALL command is extracted by the test tool 2- Verify that the value of the Cumulative Granted Non Volatile Memory (tag 'EF'/'83') set in the INSTALL command is equal to #SMALL_MEM (encoded in hexadecimal value of 2 or 4 bytes)	PF_REQ3 PM_REQ16
5	Device-Network-S → eUICC	The Device-Network-S transfers the OTA command to the eUICC		
6	eUICC → Device-Network-S	The eUICC sends the OTA response to the Device-Network-S	[R_AB_009000] or [R_AF_009000] is returned by the eUICC	
7	Device-Network-S → SM-SR	The Device-Network-S transfers the OTA response to the SM-SR-UT		
8	SM-SR-UT → SM-DP-S	Send the ES3-CreateISDP Response	1- The Status is equal to #SUCCESS 2- The isd-p-aid is equal to the one extracted in step 4	

4.4.10.2.3 TC.ES5.CreateISDP.3: Targeted_SD

Test Purpose

To ensure that the SM-SR sends the OTA command to the proper Security Domain.

Test Environment



Referenced Requirements

- EUICC_REQ50, PROC_REQ1

Initial Conditions

- The eUICC identified by `#EID` has been provisioned on the SM-SR-UT using the `#EIS_RPS`

4.4.10.2.3.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ

1	SM-DP-S → SM-SR-UT	`SEND_REQ(`
		`ES3-GetEIS,`
		`#EID_RPS)`		

Step	Direction	Sequence / Description	Expected result	REQ
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_RPS	PROC_REQ1
3	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-CreateISDP, #EID_RPS, # NEW_ICCID_RPS, #MNO1_ID_RPS, #NO_REQUIRED_MEM_RPS, #MORE_TODO_RPS)		
4	SM-SR-UT → Device-Network-S	ES5-CreateISDP command is received by the Device-Network-S over SMS, CAT_TP or HTTPs	1. The instance AID of the ISD-P present in the INSTALL command is extracted by the test tool and denoted as {CREATED_ISD_P_AID} 2. The targeted Security Domain is the ISD-R (see Note 1)	EUICC_REQ50 PROC_REQ1 PF_REQ3 PM_REQ16
5	Device-Network-S → eUICC	The Device-Network-S transfers the OTA command to the eUICC		
6	eUICC → Device-Network-S	The eUICC sends the OTA response to the Device-Network-S	[R_AB_009000] or [R_AF_009000] is returned by the eUICC	
7	Device-Network-S → SM-SR	The Device-Network-S transfers the OTA response to the SM-SR-UT		
8	SM-SR-UT → SM-DP-S	Send the ES3-CreateISDP Response	3- The Status is equal to #SUCCESS 4- The isd-p-aid is equal to the {CREATED_ISD_P_AID} extracted in step 4	PROC_REQ1 PM_REQ16

Note 1:

To verify that a command sent in SMS, HTTPS, CAT_TP, is executed by a Security Domain

- If the command is received by SMS: the TAR of the SMS is equal to the TAR of the Security Domain (bytes 13-14-15 of the AID of the Security Domain)
- If the command is received by HTTP: if the Security Domain is the ISD-R, there is no X-Admin-Targeted-Application header in the POST-Response. If the Security Domain is an ISD-P, the X-Admin-Targeted-Application header of the POST-Response contains the RID and PIX of the AID of the Security Domain
- If the command is received by CAT-TP: the TAR of the SCP80 secured packet is equal to the TAR of the Security Domain (bytes 13-14-15 of the AID of the Security Domain)

4.4.11 ES5 (SM-SR – eUICC): Profile Download Procedure

4.4.11.1 Conformance Requirements

References

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

Requirements

- PROC_REQ1, PROC_REQ1_1, PROC_REQ1_2, PM_REQ16_1, PF_REQ3, PM_REQ16, EUICC_REQ50, EUICC_REQ51

4.4.11.2 Test Cases

General Initial Conditions

- `#MNO1_S_ID` well known to the SM-SR-UT
- `#SM_DP_S_ID` and `#SM_DP_S_ACCESSPOINT` well known to the SM-SR-UT
- The eUICC identified by `#EID` has been provisioned on the SM-SR-UT using the `#EIS_RPS`

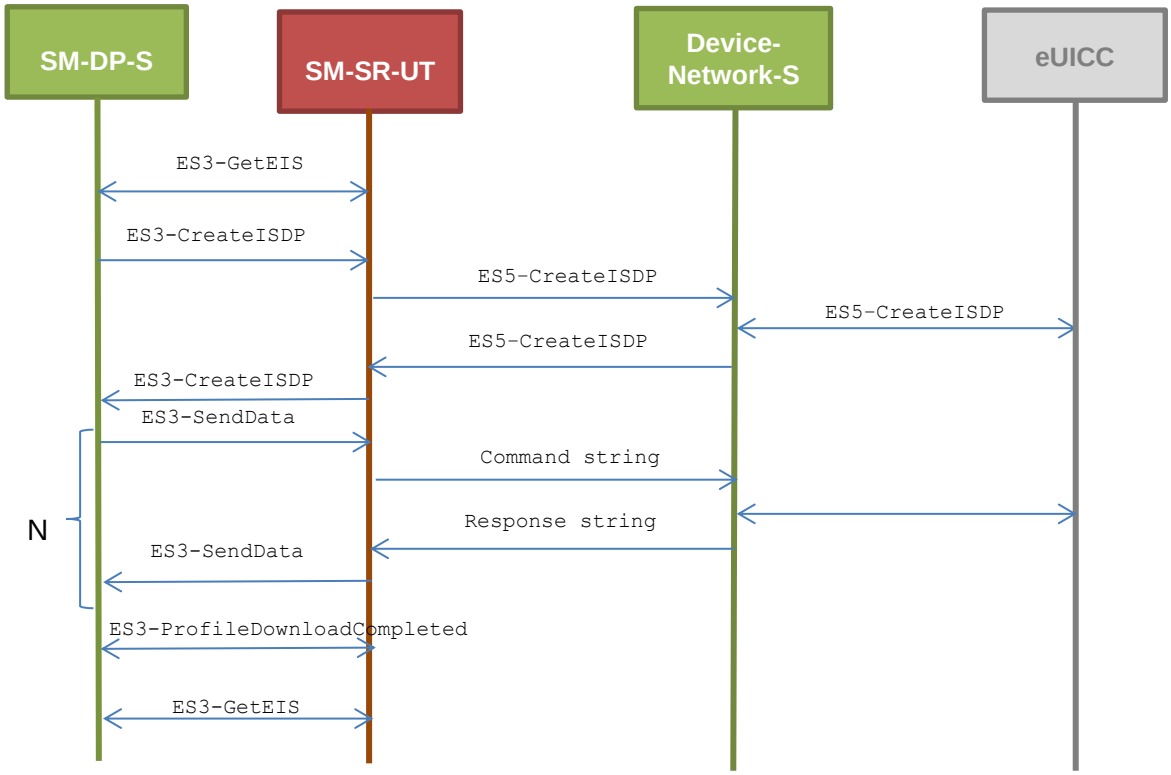
4.4.11.2.1 TC.ES5.ProfileDownload.1: Targeted Security Domains

Test Purpose

To ensure that the SM-SR sends the various commands to the correct targeted Security Domains.

An error case is also defined to ensure the SM-SR prevents the SM-DP to perform arbitrary operations in the ISD-R

Test Environment



Referenced Requirements

- PROC_REQ2, PROC_REQ3, PM_REQ17, PM_REQ18, EUICC_REQ50, EUICC_REQ51

Initial Conditions

- None

4.4.11.2.1.1 Test Sequence N°1 – Nominal Case

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1		Execute the test sequence defined in section 4.4.1.2.3.1 (TC.ES5.CreateISDP.3: Targeted_SD) in order to create an ISD-P	All steps executed successfully The AID of the created ISD-P is extracted by the SM-DP-S and denoted as {CREATED_ISD_P_AID}	

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SendData, #EID_RPS, #ISD_R_AID, EXPANDED_COMMANDS (INSTALL_FOR_PERSO({CREATED_ISD_P_AID}), #STORE_DP_CERTIF), #MORE_TODO_RPS)		
3	SM-SR-UT→ Device-Network-S	The Device-Network-S receives a command script over SMS, CAT_TP or HTTPs	The targeted Security Domain is the ISD-R (see Note 1)	PROC_REQ2 PM_REQ17 EUICC_REQ50
4	Device-Network-S → eUICC	The Device-Network-S transfers the OTA command to the eUICC		
5	eUICC → Device-Network-S	The eUICC sends the OTA response to the Device-Network-S		
6	Device-Network-S → SM-SR-UT	The Device-Network-S transfers the OTA response to the SM-SR-UT		
7	SM-SR-UT → SM-DP-S	Send the ES3-SendData response	1- The Status is equal to #SUCCESS 2- The EuiccResponseData contains the Response APDUs returned at step 5, but not the command scripting template nor the end of content indicator 3- The SM-DP-S extracts the random challenge {RC} from the STORE-DATAResponse APDU contained in the EuiccResponseData	PROC_REQ2 PM_REQ17
8	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SendData, #EID_RPS, #ISD_R_AID, EXPANDED_COMMANDS(STORE_ISDP_KEYS(#SC3_NO_DR, {RC}), #MORE_TODO_RPS)		

Step	Direction	Sequence / Description	Expected result	REQ
9	SM-SR-UT→ Device-Network-S	The Device-Network-S receives a command script over SMS, CAT_TP or HTTPs	The targeted Security Domain is the ISD-R (see Note 1)	PROC_REQ2 PM_REQ17 EUICC_REQ50
10	Device-Network-S → eUICC	The Device-Network-S transfers the OTA command to the eUICC		
11	eUICC → Device-Network-S	The eUICC sends the OTA response to the Device-Network-S		
12	Device-Network-S → SM-SR-UT	The Device-Network-S transfers the OTA response to the SM-SR-UT		
13	SM-SR-UT → SM-DP-S	Send the `ES3-SendData` response	1. The Status is equal to #SUCCESS 2. The EuiccResponseData contains the Response APDU returned at step 11, but not the command scripting template nor the end of content indicator	PROC_REQ2 PM_REQ17

14	SM-DP-S → SM-SR-UT	`SEND_REQ(`
		`  ES3-SendData,`
		`  #EID_RPS,`
		`  {CREATED_ISD_P_AID},`
		`  SCP03T_SCRIPT(`
		`  #SCP03_KVN,`
		`  #PROFILE_PACKAGE)`
		`)`		

15	SM-SR-UT→ Device-Network-S	The Device-Network-S receives a command script over SMS, CAT_TP or HTTPs	The targeted Security Domain is equal to {CREATED_ISD_P_AID} (see Note 1)	PROC_REQ3 PM_REQ17 EUICC_REQ51
16	Device-Network-S → eUICC	The Device-Network-S transfers the OTA command to the eUICC		
17	eUICC → Device-Network-S	The eUICC sends the OTA response to the Device-Network-S	The response contain SCP03t Response TLVs	
18	Device-Network-S → SM-SR-UT	The Device-Network-S transfers the OTA response to the SM-SR-UT		

Step	Direction	Sequence / Description	Expected result	REQ
19	SM-SR-UT → SM-DP-S	Send the `ES3-SendData` response	1- The `Status` is equal to `#SUCCESS` 2- `The EuiccResponseData` contains the SCP03t Response TLVs returned at step 17, but not the command scripting template nor the end of content indicator	PROC_REQ3 PM_REQ17

20	SM-DP-S → SM-SR-UT	`SEND_REQ(`
		`ES3-ProfileDownloadCompleted,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

21	SM- SR-UT → SM-DP-S	Send the `ES3-ProfileDownloadCompleted` response	The Status is equal to #SUCCESS	PROC_REQ3 PM_REQ18
22		`SEND_REQ(` ES3.GetEIS)		
23		Send the `ES3-GetEIS` response	The EIS contains a Profile with # `NEW_ICCID`_RPS and whose isd-p-aid equals {CREATED_ISD_P_AID}	PROC_REQ3

Note 1:

To verify that a command sent in SMS, HTTPS, CAT_TP, is executed by a Security Domain

- If the command is received by SMS: the TAR of the SMS is equal to the TAR of the Security Domain (bytes 13-14-15 of the AID of the Security Domain)
- If the command is received by HTTP: if the Security Domain is the ISD-R, there is no X-Admin-Targeted-Application header in the POST-Response. If the Security Domain is an ISD-P, the X-Admin-Targeted-Application header of the POST-Response contains the RID and PIX of the AID of the Security Domain
- If the command is received by CAT-TP: the TAR of the SCP80 secured packet is equal to the TAR of the Security Domain (bytes 13-14-15 of the AID of the Security Domain)

4.4.11.2.1.2 Test Sequence N°2 – Error case, APDU not allowed

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Execute the test sequence defined in section 4.4.1.2.3.1 (TC.ES5.CreateISDP.3: Targeted_SD : Targeted_SD) in order to create an ISD-P		All steps executed successfully	

Step	Direction	Sequence / Description	Expected result	REQ
2	SM-DP-S → SM-SR-UT	SEND_REQ(ES3-SendData, #EID_RPS, #ISD_R_AID, EXPANDED_COMMANDS(DELETE_ISDP1), #MORE_TODO_RPS)		
3	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_ISDR 3- The Reason code is equal to #RC_NOT_ALLOWED 4- The Device-Network-S does not receive the DELETE_ISDP1 command	PM_REQ17

4.4.12 ES7 (SM-SR – SM-SR): CreateAdditionalKeyset

4.4.12.1 Conformance Requirements

References

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

Requirements

- EUICC_REQ36, EUICC_REQ38

4.4.12.2 Test Cases

General Initial Conditions

- #MNO1_S_ID and #MNO2_S_ID well known to the SM-SR-UT
- #SM_SR_S_ID and #SM_SR_S_ACCESSPOINT well known to the SM-SR-UT

4.4.12.2.1 TC.ES7.CAK.1: CreateAdditionalKeyset with proper SIN/SDIN

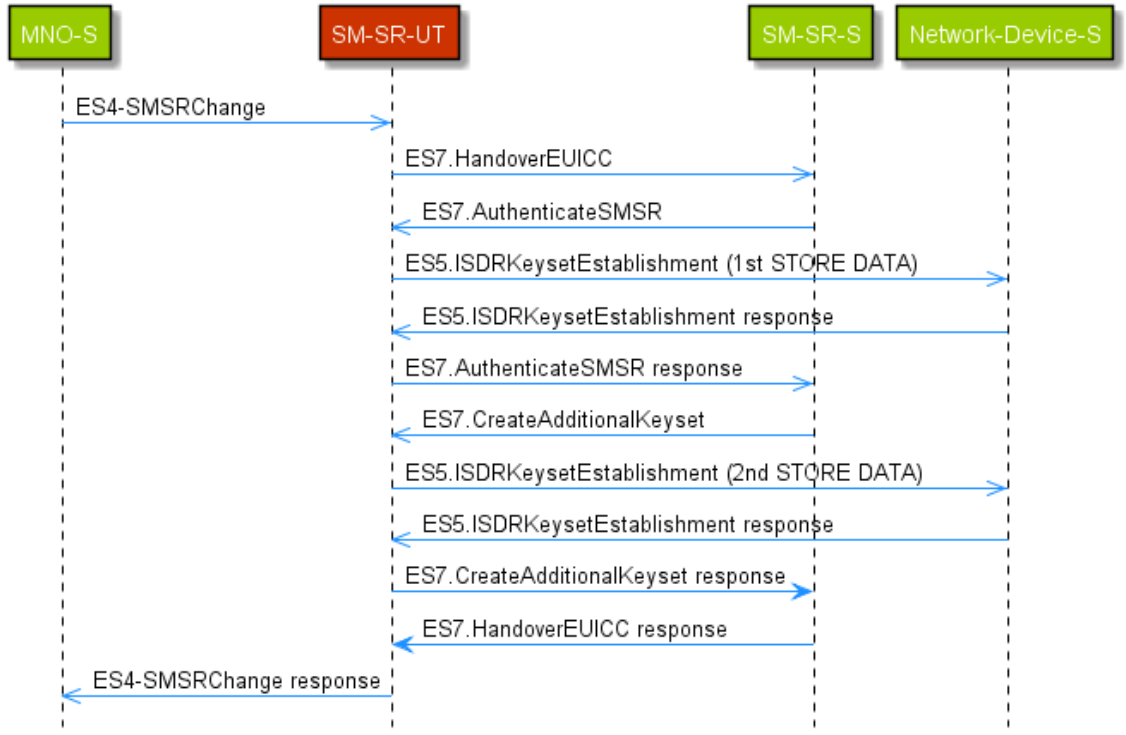
Test Purpose

To ensure that the SM-SR1 sends the correct Second STORE DATA ISD-P of ISD-R keyset establishment, in particular:

- *The SDIN (tag 45) is included if and only only if the bit b3 of the scenarioParameter byte is set to 1*

Test Environment

```
@startuml
skinparam sequence {
      ArrowColor DodgerBlue
      LifeLineBorderColor Black

      ParticipantBorderColor Black
      ParticipantFontColor White
}
hide footbox
participant OP as "MNO-S" #99CC00
participant SR1 as "SM-SR-UT" #CC3300
participant SR2 as "SM-SR-S" #99CC00
participant NDS as "Network-Device-S" #99CC00

OP->>SR1: ES4-SMSRChange
SR1->>SR2: ES7.HandoverEUICC
SR2->>SR1: ES7.AuthenticateSMSR
SR1->>NDS: ES5.ISDRKeysetEstablishment (1st STORE DATA)
NDS->>SR1: ES5.ISDRKeysetEstablishment response
SR1->>SR2: ES7.AuthenticateSMSR response
SR2->>SR1: ES7.CreateAdditionalKeyset
SR1->>NDS: ES5.ISDRKeysetEstablishment (2nd STORE DATA)
NDS->>SR1: ES5.ISDRKeysetEstablishment response
SR1->SR2: ES7.CreateAdditionalKeyset response
SR2->SR1: ES7.HandoverEUICC response
SR1->>OP: ES4-SMSRChange response
@enduml
```



Referenced Requirements

- EUICC_REQ36, EUICC_REQ38

Initial Conditions

- None

4.4.12.2.1.1 Test Sequence N°1 – ISD-R keyset Establishment without HostId

Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	MNO-S → SM-SR-UT	SEND_REQ(ES4-SMSRChange, #EID_RPS)		
2	SM-SR-UT → SM-SR-S	Send the ES7-HandoverEUICC Request	The EIS passed in parameter is equal to #EIS_ES7_RPS	
3	SM-SR-S → SM-SR-UT	SEND_REQ(ES7-AuthenticateSMSR, #EID_RPS, #VALID_SM_SR_CERTIFICATE)		
4	SM-SR-UT→ Network-Device-S	Send first STORE DATA of ES5-establishISDRKeySet function	A first STORE DATA of the ES5-establishISDRKeySet function is received by the Device-Network-S over SMS, CAT_TP or HTTPs, and contains the #VALID_SM_SR_CERTIFICATE	
5	Network-Device-S →SM-SR-UT	Send ES5- establishISDRKeySet function response, including either R_AB_03RC or R_AF_03RC depending on the transport protocol, with an RC of 16 bytes of '00'		
6	SM-SR-UT → SM-SR-S	Send the ES7- AuthenticateSMSR response	The response contains a randomChallenge equal to 16 bytes of '00'	
7	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) Note: no <HostId> is passed		

Step	Direction	Sequence / Description	Expected result	REQ
8	SM-SR-UT→ Network-Device-S	Send second STORE DATA of ES5-establishISDRKeySet function	1- A second STORE DATA of the ES5- establishISDRKeySet function is received by the Device-Network-S over SMS, CAT_TP or HTTPs. 2- The scenario filed is equal to '0309' 3- It contains no SDIN (tag 45) 4- It contains no HostId (tag 84)	EUICC_REQ38
9	Network-Device-S → SM-SR-UT	Send ES5-establishISDRKeySet function response, including either R_AB_RECEIPT or R_AF_RECEIPT depending on the transport protocol, with an Receipt of 16 bytes of '00'		
10	SM-SR-UT → SM-SR-S	Send the ES7- CreateAdditionalKeyset response	1- The response contains a Receipt equal to 16 bytes of '00' 2- The response contains no Derivationrandom	
11	SM-SR-S → SM-SR-UT	SEND_SUCCESS_RESP(ES7-HandoverEUICC)		
12	SM-SR-UT → MNO-S	Send the ES4-SMSRChange response	The Status is equal to #SUCCESS	EUICC_REQ36

4.4.12.2.1.2 Test Sequence N°2 – ISD-R keyset Establishment with HostId

Initial Conditions

- The eUICC identified by #EID has been provisioned on the SM-SR-UT using the #EIS_RPS

Step	Direction	Sequence / Description	Expected result	REQ
1	Execute steps 1 to 6 of test sequence 4.4.11.2.1.1			

Step	Direction	Sequence / Description	Expected result	REQ
2	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_HOST, #EPHEMERAL_PK_RPS, #SIGNATURE_RPS #HOST_ID)		
3	SM-SR-UT→ Network-Device-S	Send second STORE DATA of ES5-establishISDRKeySet function	1- A second STORE DATA of the ES5- establishISDRKeySet function is received by the Device-Network-S over SMS, CAT_TP or HTTPs. 2- The scenario filed is equal to '030B' 3- It contains an SDIN (tag 45) with value #VIRTUAL_SDIN 4- It contains a HostId (tag 84) with value #HOST_ID	EUICC_REQ38
4	Network-Device-S → SM-SR-UT	Send ES5- establishISDRKeySet function response, including either R_AB_RECEIPT or R_AF_RECEIPT depending on the transport protocol, with a DerivationRandom of 16 bytes of '00' and a Receipt of 16 bytes of '00'		
5	SM-SR-UT → SM-SR-S	Send the ES7- CreateAdditionalKeyset response	1- The response contains a Receipt equal to 16 bytes of '00' 2- The response contains a Derivationrandom equal to 16 bytes of '00'	
6	SM-SR-S → SM-SR-UT	SEND_SUCCESS_RESP(ES7-HandoverEUICC)		
7	SM-SR-UT → MNO-S	Send the ES4-SMSRChange response	The Status is equal to #SUCCESS	EUICC_REQ36

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]		

Step	Direction	Sequence / Description	Expected result	REQ
3	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ10, EUICC_REQ11
4	DS → eUICC-UT	`[GET_DATA_5A]`		
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] (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] (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] (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		

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- 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		
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		

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- INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed (i.e. SW='9000') 3- SW='6A88' for the GET STATUS command (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		

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_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, new ProfileS including a file and an applet or an additional SSD are dynamically downloaded: the selection of these additional 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			

Step	Direction	Sequence / Description	Expected result	REQ
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			

Step	Direction	Sequence / Description	Expected result	REQ
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]		

Step	Direction	Sequence / Description	Expected result	REQ
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	
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	

Step	Direction	Sequence / Description	Expected result	REQ
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

5.2.3.2.3.5 Test Sequence N°5 - Nominal Case using CAT_TP: SSD Selectable Only on an Enabled Profile

Initial Conditions

- SSD3 is not present on the Profile linked to the #DEFAULT_ISD_P_AID
- #PE_SSD3 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_SSD3) 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_SSD3]		
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	

Step	Direction	Sequence / Description	Expected result	REQ
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_SSD3]		
12	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ9

5.2.3.2.3.6 Test Sequence N°6 - Nominal Case using HTTPS: SSD Selectable Only on an Enabled Profile

Initial Conditions

- SSD3 is not present on the Profile linked to the #DEFAULT_ISD_P_AID
- #PE_SSD3 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_SSD3) 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_SSD3]		
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_SSD3]		
12	eUICC-UT → DS	ATS	SW='9000'	EUICC_REQ9

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 or SSD is installed through the MNO-SD on the Enabled Profile. Then, another applet or SSD 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 (applet or SSD) 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), as well as SSD1 and SSD2 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: Applet with same TAR within Two Profiles

Initial Conditions

- 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		

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- 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 4.2.3.2.2.1 (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	
9	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 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
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 (including #PE_APPLET1) under the #ISD_P_AID1		All steps successfully executed	EUICC_REQ3
11	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: Applet with same TAR within Two Profiles

Initial Conditions

- 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		

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 #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 4.2.3.2.3.1 (TC.ES5.CISDP.3:CreateISDP_HTTPS) from step 3 to step 4 in order to create the #ISD_P_AID1		All steps successfully executed	
9	Execute the test sequence defined in section 4.2.17.2.3.1 (TC.ES8.EISDPK.3:EstablishISDPkeyset_HTTPS) from step 3 to step 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
10	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
11	Close HTTPS session as described in section 4.2.1.7			

5.2.3.2.4.3 Test Sequence N°3 - Error Case: Applet with 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			
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		

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- 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'				

5.2.3.2.4.4 Test Sequence N°4 - Nominal Case using CAT_TP: SSDs with Same TAR within Two Profiles

Initial Conditions

- SSD1 and SSD2 are not present on the Profile identified by #ISD_P_AID1
- #PE_SSD1 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			

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_SSD2]) 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 4.2.3.2.2.1 (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	
9	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 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
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 (including #PE_SSD1) under the #ISD_P_AID1		All steps successfully executed	EUICC_REQ3
11	Close CAT_TP session as described in section 4.2.1.4			

5.2.3.2.4.5 Test Sequence N°5 - Nominal Case using HTTPS: SSDs with Same TAR within Two Profiles

Initial Conditions

- SSD1 and SSD2 are not present on the Profile identified by #ISD_P_AID1
- #PE_SSD1 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			

Step	Direction	Sequence / Description	Expected result	REQ
2	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #MNO_TAR, [INSTALL_SSD2]) 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 4.2.3.2.3.1 (TC.ES5.CISDP.3:CreateISDP_HTTPS) from step 3 to step 4 in order to create the #ISD_P_AID1		All steps successfully executed	
9	Execute the test sequence defined in section 4.2.17.2.3.1 (TC.ES8.EISDPK.3:EstablishISDPkeyset_HTTPS) from step 3 to step 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
10	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_SSD1) under the #ISD_P_AID1		All steps successfully executed	EUICC_REQ3
11	Close HTTPS session as described in section 4.2.1.7			

5.2.3.2.4.6 Test Sequence N°6 - Error Case: SSD with 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, [INSTALL_SSD_TAR_ISDR]) 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='6985' for the INSTALL command (see Note 1)	EUICC_REQ3
6	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 or SSD is installed through the MNO-SD on the Enabled Profile. Then, another apple or SSDt 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 (applet or SSD) 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: Applets with same AID within Two Profiles

Initial Conditions

- 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'	
7	Open CAT_TP session on ISD-R as described in section 4.2.1.2			
8	Execute the test sequence defined in section 4.2.3.2.2.1 (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	
9	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 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
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 (including #PE_APPLET3) under the #ISD_P_AID1		All steps successfully executed	EUICC_REQ2
11	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: Applets with same AID within Two Profiles

Initial Conditions

- 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'	
7	Open HTTPS session on ISD-R as described in section 4.2.1.5			
8	Execute the test sequence defined in section 4.2.3.2.3.1 (TC.ES5.CISDP.3:CreateISDP_HTTPS) from step 3 to step 4 in order to create the #ISD_P_AID1		All steps successfully executed	
9	Execute the test sequence defined in section 4.2.17.2.3.1 (TC.ES8.EISDPK.3:EstablishISDPkeyset_HTTPS) from step 3 to step 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
10	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
11	Close HTTPS session as described in section 4.2.1.7			

5.2.3.2.5.3 Test Sequence N°3 - Error Case: Applet with 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		
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

Step	Direction	Sequence / Description	Expected result	REQ
11	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note 1: The SW MAY be also '6A80'				

5.2.3.2.5.4 Test Sequence N°4 - Nominal Case using CAT_TP: SSDs with Same AID within Two Profiles

Initial Conditions

- SSD3 is not present on the Profile identified by `#ISD_P_AID1`
- `#PE_SSD3` 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, [INSTALL_SSD3]) 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 4.2.3.2.2.1 (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	
9	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 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
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 (including #PE_SSD3) under the #ISD_P_AID1		All steps successfully executed	EUICC_REQ2

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

5.2.3.2.5.5 Test Sequence N°5 - Nominal Case using HTTPS: SSDs with Same AID within Two Profiles

Initial Conditions

- SSD3 is not present on the Profile identified by `#ISD_P_AID1`
- `#PE_SSD3` 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, [INSTALL_SSD3]) 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 4.2.3.2.3.1 (TC.ES5.CISDP.3:CreateISDP_HTTPS) from step 3 to step 4 in order to create the #ISD_P_AID1		All steps successfully executed	
9	Execute the test sequence defined in section 4.2.17.2.3.1 (TC.ES8.EISDPK.3:EstablishISDPkeyset_HTTPS) from step 3 to step 6 in order to personalize the #ISD_P_AID1		All steps successfully executed	
10	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_SSD3) under the #ISD_P_AID1		All steps successfully executed	EUICC_REQ2
11	Close HTTPS session as described in section 4.2.1.7			

5.2.3.2.5.6 Test Sequence N°6 - Error Case: SSD with 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, [INSTALL_SSD_AID_ECASD]) 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='6985' for the INSTALL command (see Note 1)	EUICC_REQ2
6	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		
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, EUICC_REQ4_1_3_3_5, EUICC_REQ4_1_3_3_8

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		

Step	Direction	Sequence / Description	Expected result	REQ
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		

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- INITIALIZE UPDATE and EXTERNAL AUTHENTICATE commands are successfully executed (i.e. SW='9000') 3- The SW is '6985' for the INSTALL command (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', '6A88' or '6D00'				

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

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 `#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 `#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		
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

Step	Direction	Sequence / Description	Expected result	REQ
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 Annex H.*			
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 TLS 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	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`  #SPI_VALUE,`
		`  #ISD_R_TAR,`
		`  [OPEN_SCP81_WITH_RETRY])`		

3	Execute the generic sub-sequence “Open HTTPS session on ISD-R” defined in section 4.2.1.5 from step 2 to step 9			
4	eUICC-UT → DS	TLS_CLIENT_HELLO	The TLS_CLIENT_HELLO does not contain a SessionTicket extension (SessionTicket extension type = 0x0023)	EUICC_REQ56
5	Execute the generic sub-sequence “Open HTTPS session on ISD-R” defined in section 4.2.1.5 from step 11 to step 14			

Step	Direction	Sequence / Description	Expected result	REQ
6	DS → eUICC-UT	RESET	IP communication is broken ATR returned by eUICC	
7	DS → eUICC-UT	[TERMINAL_PROFILE]	Toolkit initialization	
8	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* OPEN CHANNEL	See Note	
9	DS → eUICC-UT	FETCH		
10	eUICC-UT → DS	*PROACTIVE COMMAND:* OPEN CHANNEL	The ISD-R makes first attempt for resuming the HTTP administration session	
11	DS → eUICC-UT	TERMINAL RESPONSE		
12	eUICC-UT → DS	TLS_CLIENT_HELLO	The TLS_CLIENT_HELLO contains an empty Session Identifier (i.e. the previous TLS session is not reused)	EUICC_REQ56
13	Execute the generic sub-sequence “Open HTTPS session on ISD-R” defined in section 4.2.1.5 from step 11 to step 14			
14	Close HTTPS session as described in section 4.2.1.7			
Note: The OPEN CHANNEL command MAY be triggered by a TIMER EXPIRATION if the eUICC supports TIMER MANAGEMENT.				

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

Step	Direction	Sequence / Description	Expected result	REQ
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, except for the last two test sequences which use HTTPS. 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, EUICC_REQ4_1_3_3_5, EUICC_REQ4_1_3_3_8

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	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`  #SPI_VALUE,`
		`  #ISD_P_TAR1,`
		`  SCP03T_SCRIPT(`
		`  #SCP03_KVN,`
		`  #PE_HEADER))`
		Use the SCP03 keys `{SCP_KENC}` and `{SCP_KMAC}`
		Change the length value of the INITIALIZE UPDATE TLV command before sending the script (e.g. with '11' instead of '0A')		

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- The response data is equal to `[R_AB_SCP03T_IU_01]` See Note	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ59
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: Instead of using the SCP03t error tag (0x9F44), the eUICC MAY return the Bad format TLV tag (i.e. 0x90) indicating "Wrong length found" (i.e. 0x02) as defined in ETSI TS 102 226 [6].				

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	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03T_SCRIPT(`
		`#BAD_SCP03_KVN,`
		`#PE_HEADER))`
		Use the SCP03 keys `{SCP_KENC}` and `{SCP_KMAC}`		

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 response data is equal to `[R_AB_SCP03T_IU_03]`	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ59
5	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	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_P_TAR1, SCP03T_SCRIPT(#SCP03_KVN, #PE_HEADER)) Use the SCP03 keys {SCP_KENC} and {SCP_KMAC} Change the length value of the EXTERNAL AUTHENTICATE TLV command (TAG '85') before sending the script (e.g. with '19' instead of '11')		
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 response data is equal to [R_AB_SCP03T_EA_01] See Note	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ60
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: Instead of using the SCP03t error tag (0x9F45), the eUICC MAY return the Bad format TLV tag (i.e. 0x90) indicating "Wrong length found" (i.e. 0x02) as defined in ETSI TS 102 226 [6].				

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	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} and {SCP_KMAC} see Note		
2	eUICC-UT → DS	*PROACTIVE COMMAND PENDING:* SEND SHORT MESSAGE		

Step	Direction	Sequence / Description	Expected result	REQ
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 response data is equal to `[R_AB_SCP03T_EA_02]`	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ60
5	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	DS → eUICC-UT	`ENVELOPE_SMS_PP(`
		`#SPI_VALUE,`
		`#ISD_P_TAR1,`
		`SCP03T_SCRIPT(`
		`#SCP03_KVN,`
		`#PE_HEADER))`
		Use the SCP03 keys `{SCP_KENC}` and `{SCP_KMAC}`
		Change the length value of the Profile data TLV command (TAG '86') before sending the script		

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 response data is equal to `[R_AB_SCP03T_01]` See Note	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ61
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	
Note: Instead of using the SCP03t error tag (0x9F46), the eUICC MAY return the Bad format TLV tag (i.e. 0x90) indicating "Wrong length found" (i.e. 0x02) ad defined in ETSI TS 102 226 [6].				

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	DS → eUICC-UT	ENVELOPE_SMS_PP(#SPI_VALUE, #ISD_P_TAR1, SCP03T_SCRIPT(#SCP03_KVN, #PE_HEADER)) Use the SCP03 keys {SCP_KENC} and, {SCP_KMAC} Corrupt a block of ciphered data in the Profile data TLV command (TAG '86') before sending the script		
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 response data is equal to [R_AB_SCP03T_02]	EUICC_REQ13, EUICC_REQ19, EUICC_REQ21, EUICC_REQ22, EUICC_REQ61
5	DS → eUICC-UT	TERMINAL RESPONSE	SW='9000'	

5.2.4.2.5.7 Test Sequence N°7 – Error Case: Incorrect length in Replace session key TLV command using HTTPs

Initial Conditions

- None

Step	Direction	Sequence / Description	Expected result	REQ
1	Open HTTPS session on ISD-R as described in section 4.2.1.5			
2	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP (#ISD_P_AID1, SCP03T_SCRIPT_INI_AUTH (#SCP03_KVN)) Use the SCP03 keys {SCP_KENC} and {SCP_KMAC}		EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58, EUICC_REQ58_1, EUICC_REQ4_1_3_3_1,

Step	Direction	Sequence / Description	Expected result	REQ
3	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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R #CONTENT_TYPE #TRANSFER_ENCODING #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]	PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61
4	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP (#ISD_P_AID1, SCP03T_REPLACE_SESSION_KEYS_BAD_LENGTH ()) Use the SCP03 keys {SCP_KENC} and {SCP_KMAC}	The response to the REPLACE_SESSION_KEYS command (i.e. TAG '87') SHALL be equal to [R_AF_SCP03T_PP_01]	EUICC_REQ4_1_3_3_2, EUICC_REQ4_1_3_3_4, PF_REQ4_1_3_3_1, EUICC_REQ4_1_3_3_5, EUICC_REQ4_1_3_3_8

5.2.4.2.5.8 Test Sequence N°8 – Error Case: Incorrect security in Replace session key TLV command using HTTPs

Initial Conditions

- None

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

2	DS → eUICC-UT	TLS_APPLICATION containing the result of
		`HTTPS_CONTENT_ISDP (`
		`#ISD_P_AID1,`
		`SCP03T_SCRIPT_INI_AUTH (`
		`#SCP03_KVN)`
		`)`
		
		Use the SCP03 keys `{SCP_KENC}` and `{SCP_KMAC}`
				EUICC_REQ17, EUICC_REQ49, EUICC_REQ51, EUICC_REQ52, EUICC_REQ57, EUICC_REQ58, EUICC_REQ58_1, EUICC_REQ4_1_3_3_1,

3	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
			`#HOST`
			`#X_ADMIN_PROTOCOL`
			`#X_ADMIN_FROM_ISD_R`
			`#CONTENT_TYPE`
			`#TRANSFER_ENCODING`
			`#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]`
				PM_REQ9, EUICC_REQ14, EUICC_REQ23, EUICC_REQ43, EUICC_REQ46, EUICC_REQ47, EUICC_REQ48, EUICC_REQ52, EUICC_REQ59, EUICC_REQ60, EUICC_REQ61

Step	Direction	Sequence / Description	Expected result	REQ
4	DS → eUICC-UT	TLS_APPLICATION containing the result of HTTPS_CONTENT_ISDP (#ISD_P_AID1, SCP03T_REPLACE_SESSION_KEYS ()) Do NOT use the SCP03 keys {SCP_KENC} and {SCP_KMAC}	The response to the REPLACE_SESSION_KEYS command (i.e. TAG '87') SHALL be equal to [R_AF_SCP03T_PP_02]	EUICC_REQ4_1_3_3_2, EUICC_REQ4_1_3_3_4, PF_REQ4_1_3_3_1, EUICC_REQ4_1_3_3_5, EUICC_REQ4_1_3_3_8

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]		
5	eUICC-UT → DS	DGI '7F21' returned	1- The returned DGI '7F21'contains the TLV certificate [R_CASD_SC2B] 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 returned DGI '7F21' contains the TLV certificate `[R_CASD_SC3]` 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		

Step	Direction	Sequence / Description	Expected result	REQ
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_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_REQ2, EUICC_REQ51_1

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			
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

Step	Direction	Sequence / Description	Expected result	REQ
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	DS → eUICC-UT	TLS_APPLICATION containing the result of `HTTPS_EMPTY_CONTENT()`		EUICC_REQ51_1

Step	Direction	Sequence / Description	Expected result	REQ
5	eUICC-UT → DS	TLS_APPLICATION with empty body	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 #HOST #X_ADMIN_PROTOCOL #X_ADMIN_FROM_ISD_R 4- The HTTP body is empty	EUICC_REQ51_1
6	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
7	Execute steps 4 and 5 of this sequence			EUICC_REQ51_1
8	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
9	Execute steps 4 and 5 of this sequence			EUICC_REQ51_1
10	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
11	Close HTTPS session as described in section 4.2.1.7			
12	Open HTTPS session on ISD-R as described in section 4.2.1.5			
13	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
14	Close HTTPS session as described in section 4.2.1.7			
15	Open HTTPS session on ISD-R as described in section 4.2.1.5			
16	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

Step	Direction	Sequence / Description	Expected result	REQ
17		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

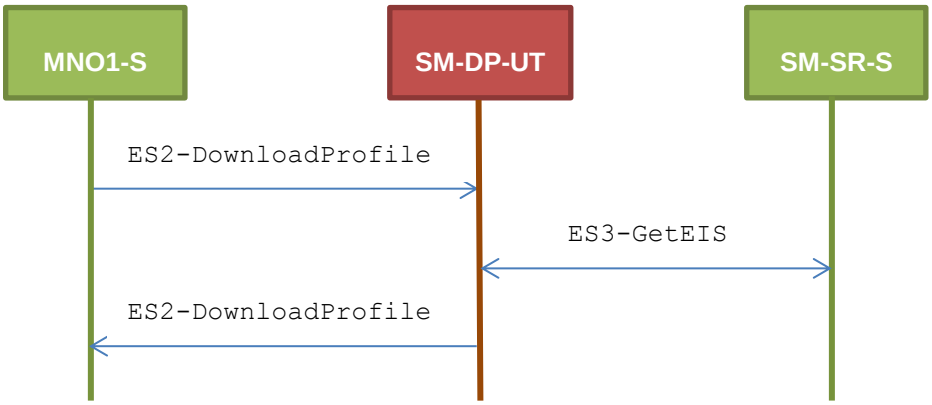
- None

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`
- The variable `{SM_DP_ID_RPS}` SHALL be set to `#SM_DP_UT_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	The Status is equal to #FAILED	PM_REQ11, SEC_REQ15

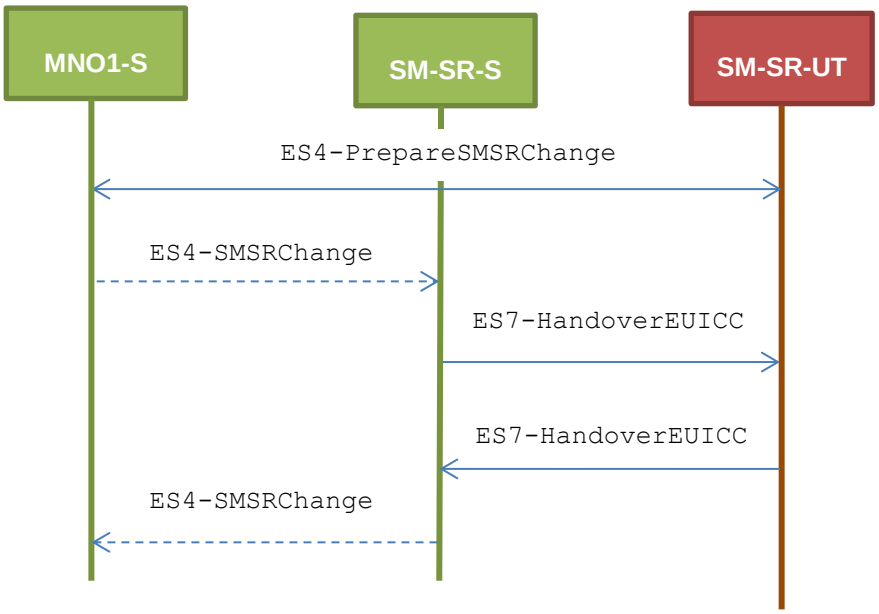
5.3.1.2.1.2 *VOID*

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
- The variable {SM_DP_ID_RPS} SHALL be set to #SM_DP_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	The Status is equal to #FAILED	EUICC_REQ39, SEC_REQ15

5.3.1.2.2.2 *VOID*

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

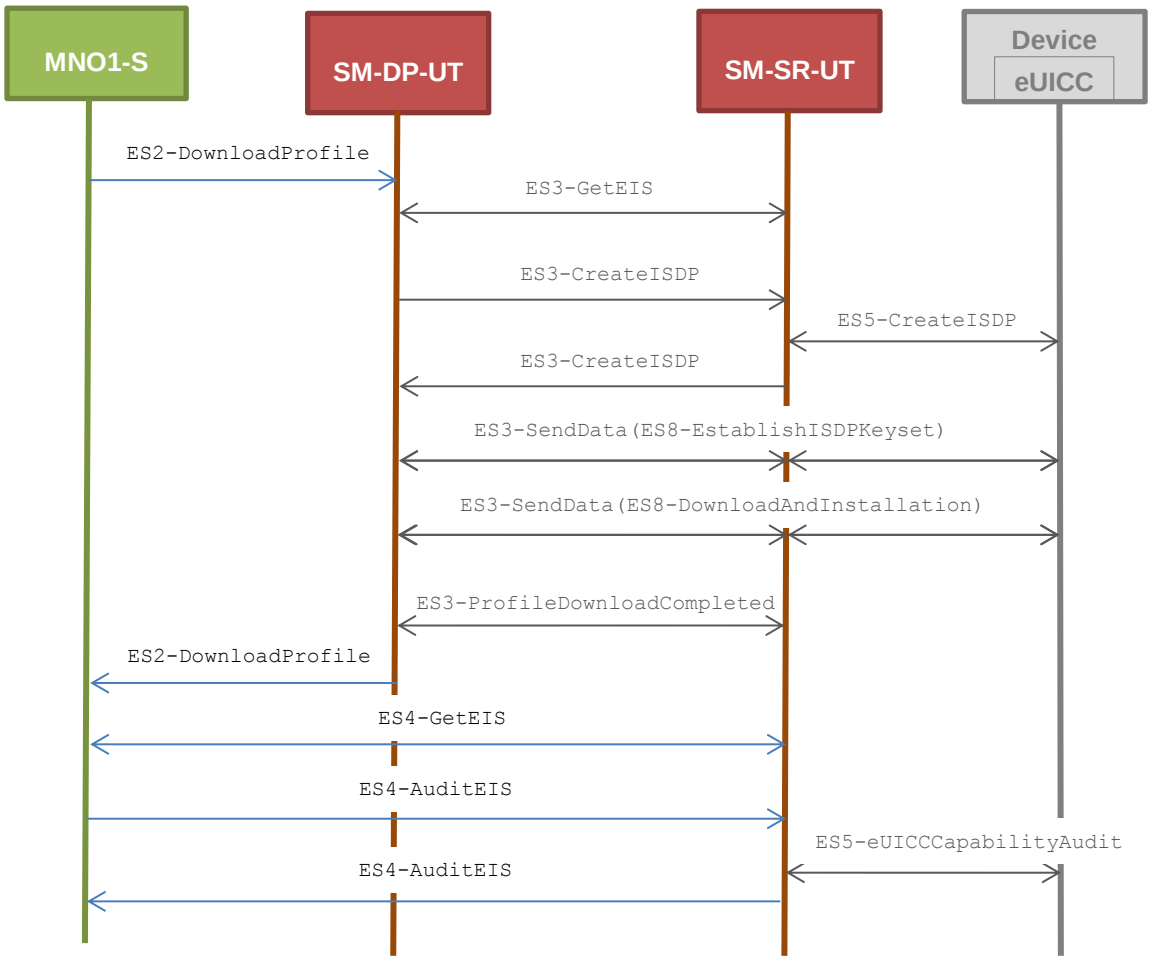
- #MNO1_S_ID and #MNO1_S_ACCESSPOINT well known to the SM-DP-UT
- #MNO1_S_ID well known to the SM-SR-UT
- #MNO1_S_ACCESSPOINT well known to the SM-SR-UT
  - A direct connection exists between the MNO1-S and the SM-SR-UT
- The variable {SM_SR_ID_RPS} SHALL be set to #SM_SR_UT_ID_RPS
- #SM_SR_ID and #SM_SR_ACCESSPOINT 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-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS, {SM_SR_ID_RPS})		
5	SM-SR-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. If the free memory of the new Profile is returned, verify that the allocated memory is different from 0 b. if no free memory is returned, verify that the allocated memory is set to 0 c. the remaining memory is 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-SR-UT	`SEND_REQ( ES4-GetEIS, #EID_RPS, {SM_SR_ID_RPS})`		
5	SM-SR-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. if the free memory of the new Profile is returned, verify that the allocated memory is different from 0 b. if no free memory is returned, verify that the allocated memory is set to 0 c. the remaining memory is 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

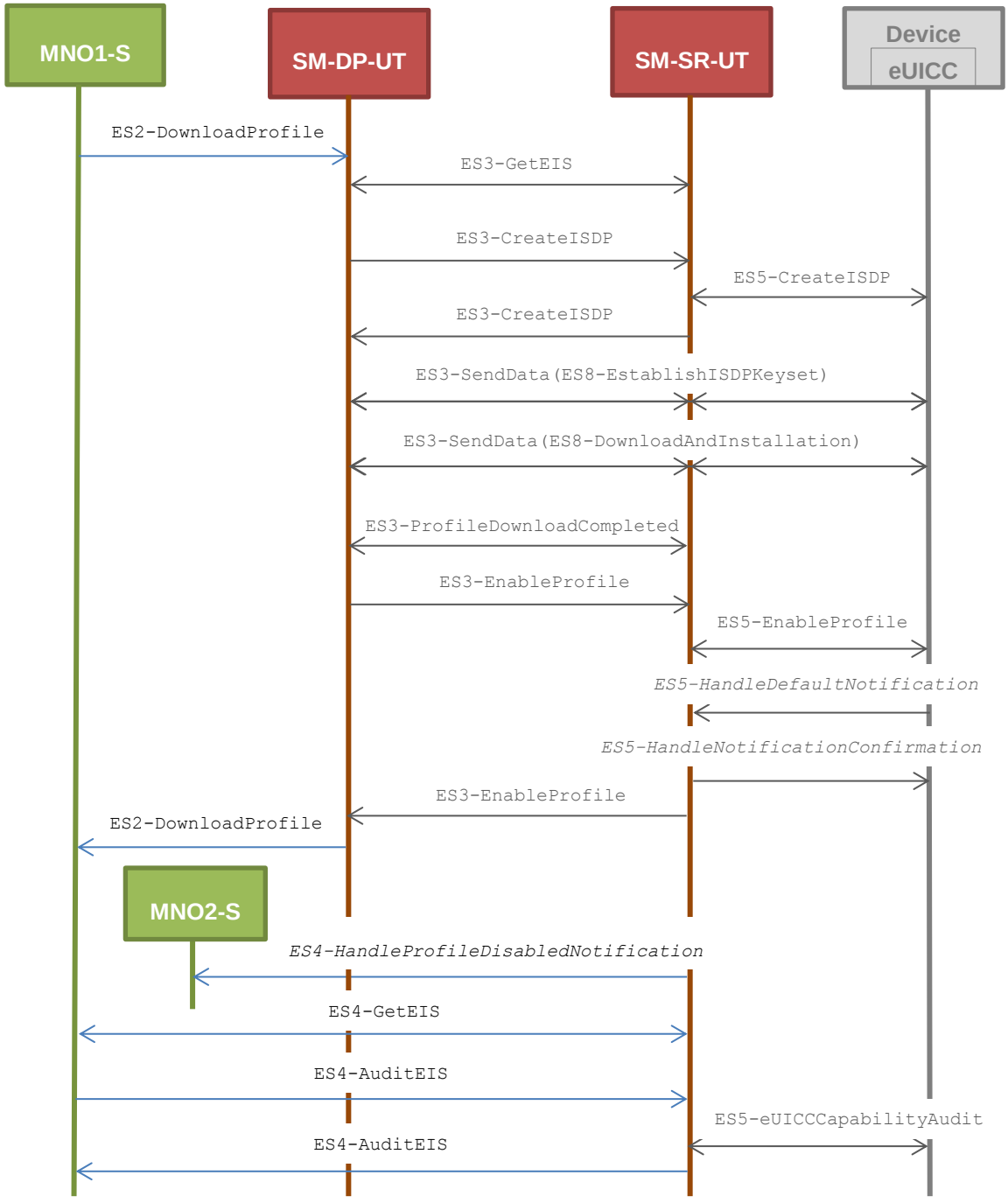
- PROC_REQ1, PROC_REQ2, PROC_REQ3, PROC_REQ7, PROC_REQ20
- PM_REQ4, PM_REQ8, PM_REQ9, PM_REQ11, PM_REQ14, PM_REQ16, PM_REQ17, PM_REQ18, PM_REQ22, PM_REQ25
- PF_REQ2, PF_REQ3, PF_REQ4, PF_REQ7, PF_REQ18, PF_REQ27
- EUICC_REQ27, EUICC_REQ29

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)
  - 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
  - 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`

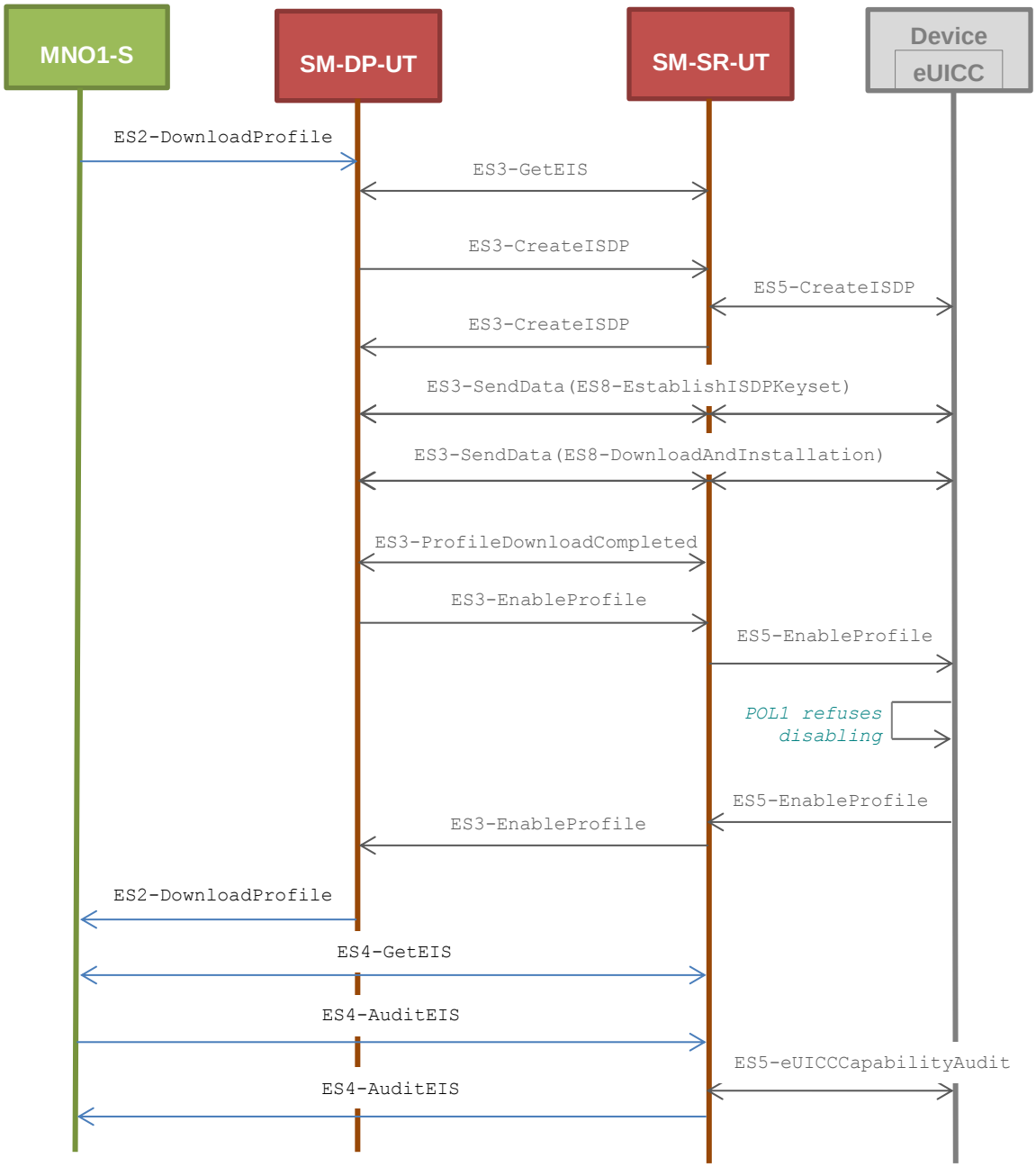
- POL1 MAY be adapted in the eUICC

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-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS, {SM_SR_ID_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 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. if the free memory of the new Profile is returned, verify that the allocated memory is different from 0 b. if no free memory is returned, verify that the allocated memory is set to 0 c. the remaining memory is 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 #WARNING 2- The couple (Subject code and Reason code) is equal to #SC_POL1, #RC_REFUSED 3- 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-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS, {SM_SR_ID_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 information is 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. if the `free memory` of the new Profile is returned, verify that the `allocated memory` is different from 0 b. if no `free memory` is returned, verify that the `allocated memory` is set to 0 c. the `remaining memory` is 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

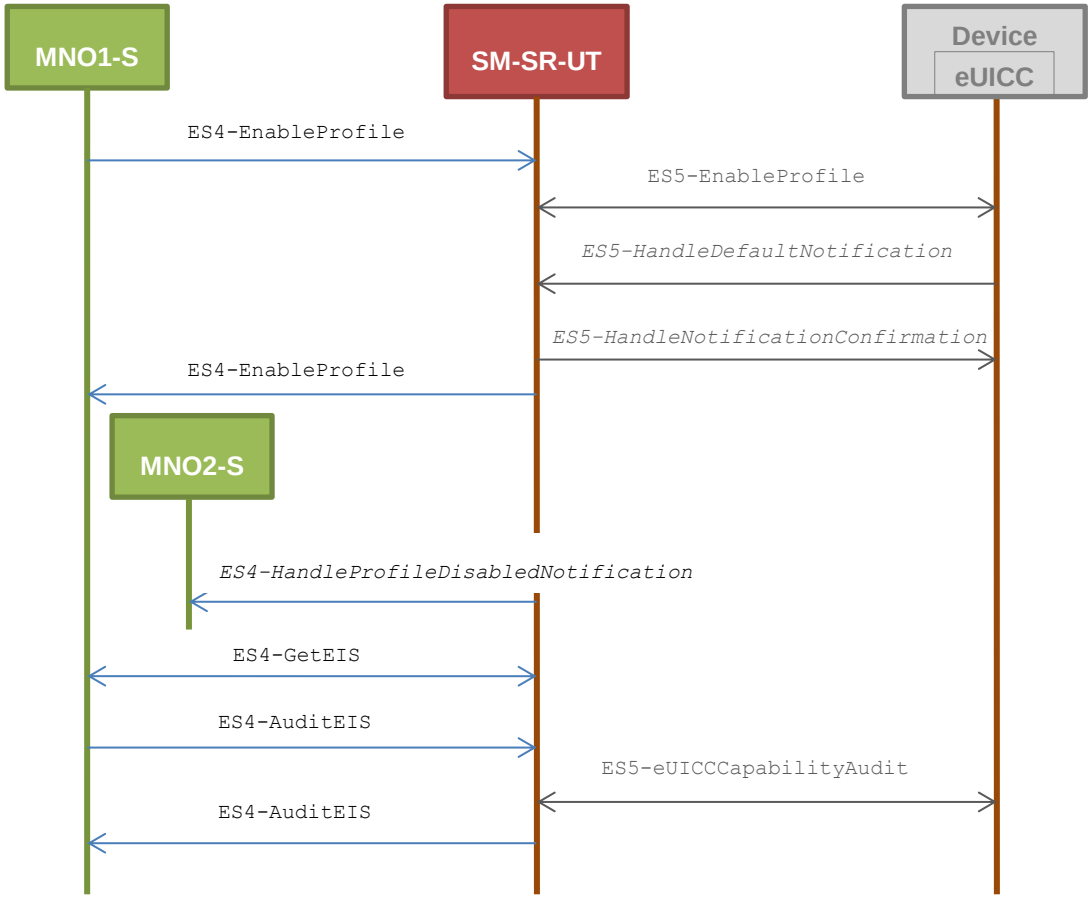
- PF_REQ2, PF_REQ4, PF_REQ6, PF_REQ7, PF_REQ24, PF_REQ27, PF_REQ29
- PROC_REQ5, PROC_REQ6, PROC_REQ20
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

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

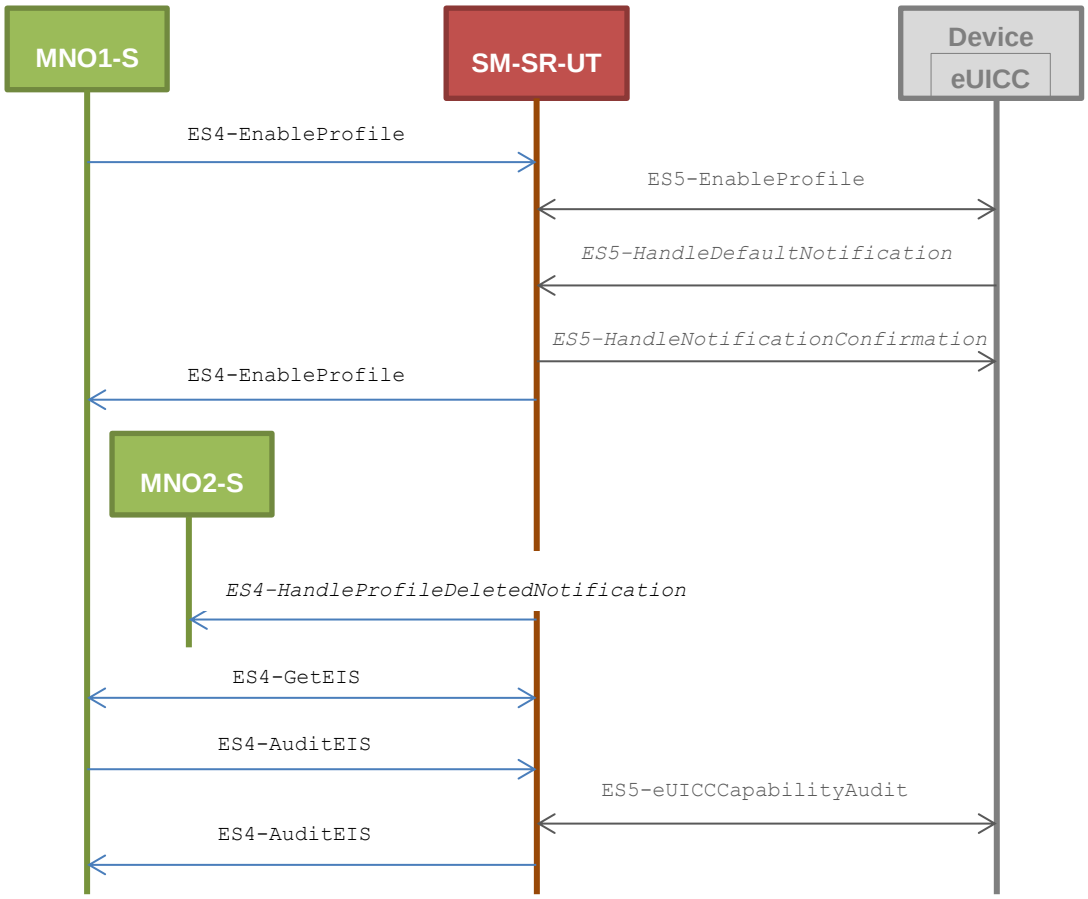
1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-EnableProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment MAY take several minutes)*			

Step	Direction	Sequence / Description	Expected result	REQ
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 information is 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	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)		

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 → 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-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)`		

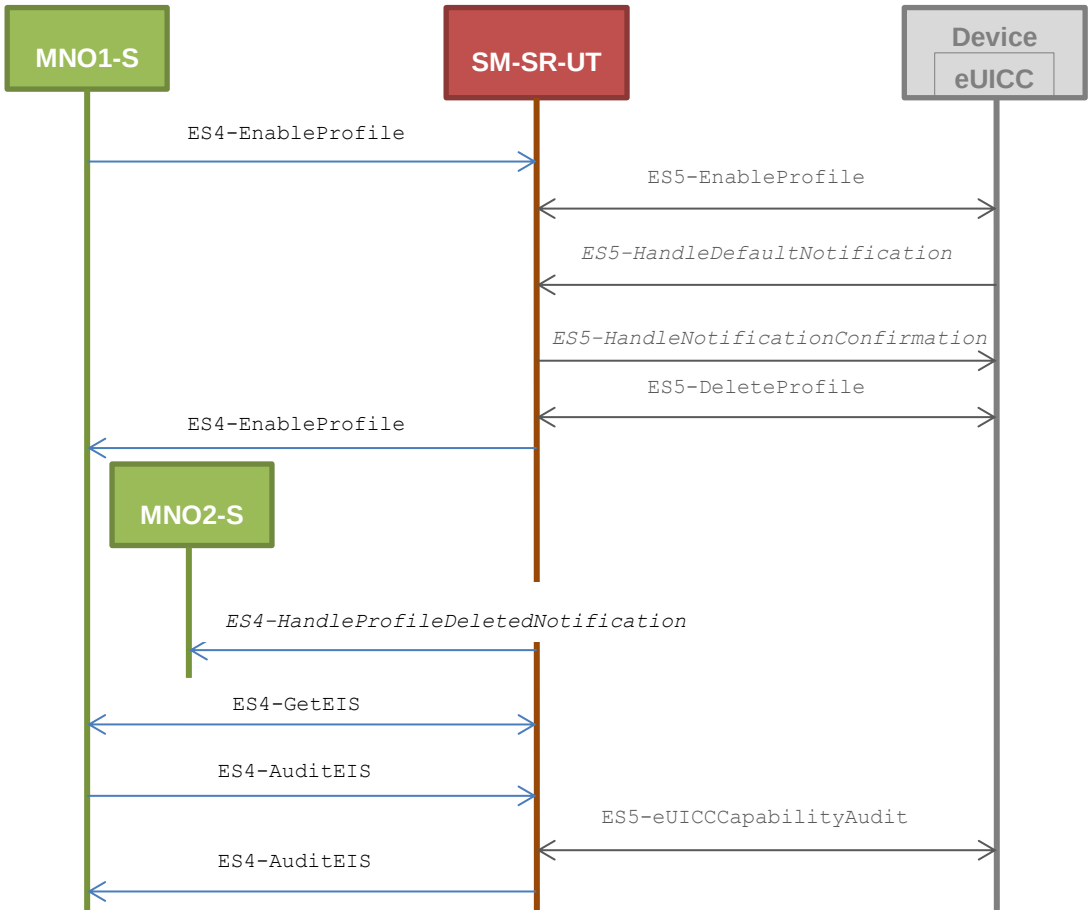
6	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R information is not present b. only Profiles related to the MNO1-S are present among which that identified by `#NEW_ICCID` c. the Profile identified by `#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` is 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

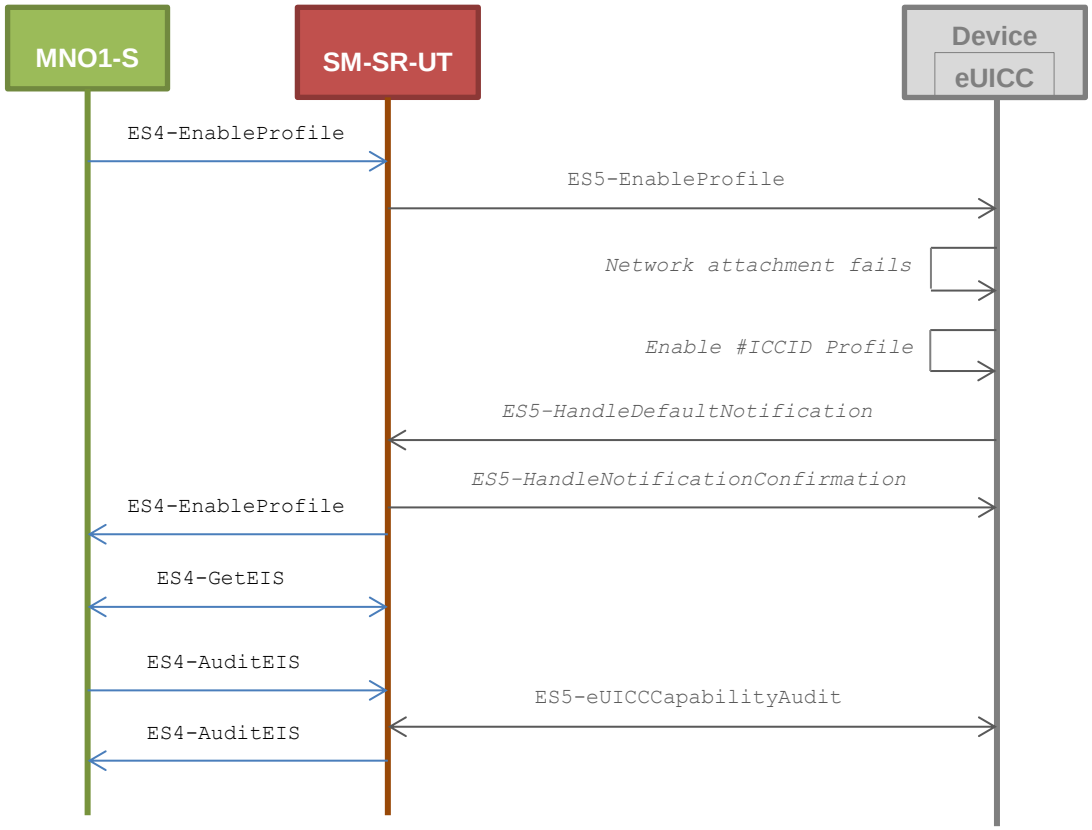
- The Profile downloaded, identified by `#NEW_ICCID`, SHALL be adapted to contain correct Connectivity Parameters (i.e. `#MNO1_CON_NAN`, `#MNO1_CON_LOGIN`, `#MNO1_CON_PWD`)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by `#NEW_ICCID`)
  - It 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	The Status is equal to #SUCCESS	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)		
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 information is 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` is 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)`		

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 → MNO1-S	Send the `ES4-EnableProfile` 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_INACCESSIBLE`	PF_REQ2, PF_REQ4, PF_REQ24, PROC_REQ6, 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 information is 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

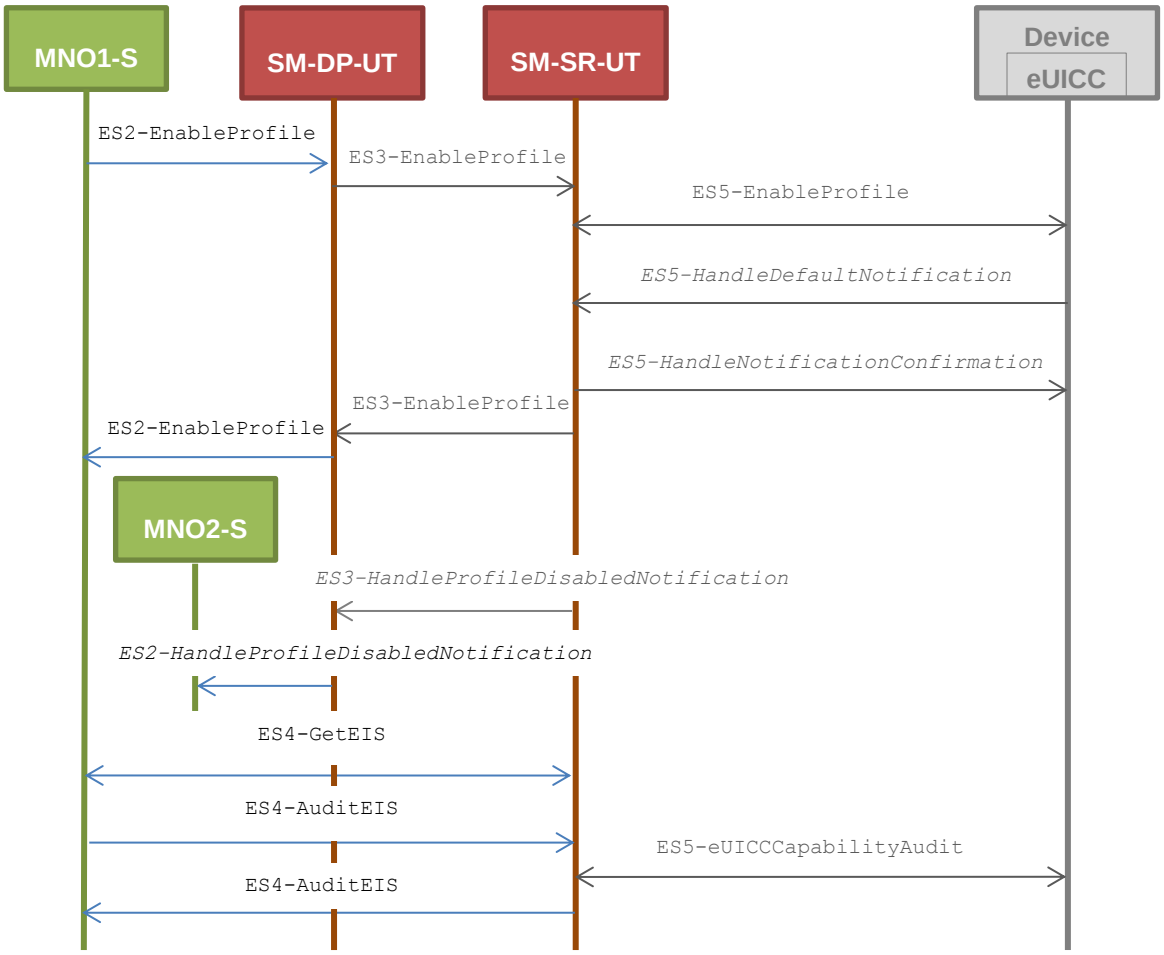
- PF_REQ2, PF_REQ4, PF_REQ6, PF_REQ7, PF_REQ12, PF_REQ15, PF_REQ17, PF_REQ18, PF_REQ21, PF_REQ23
- PROC_REQ7, PROC_REQ8, PROC_REQ20
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

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

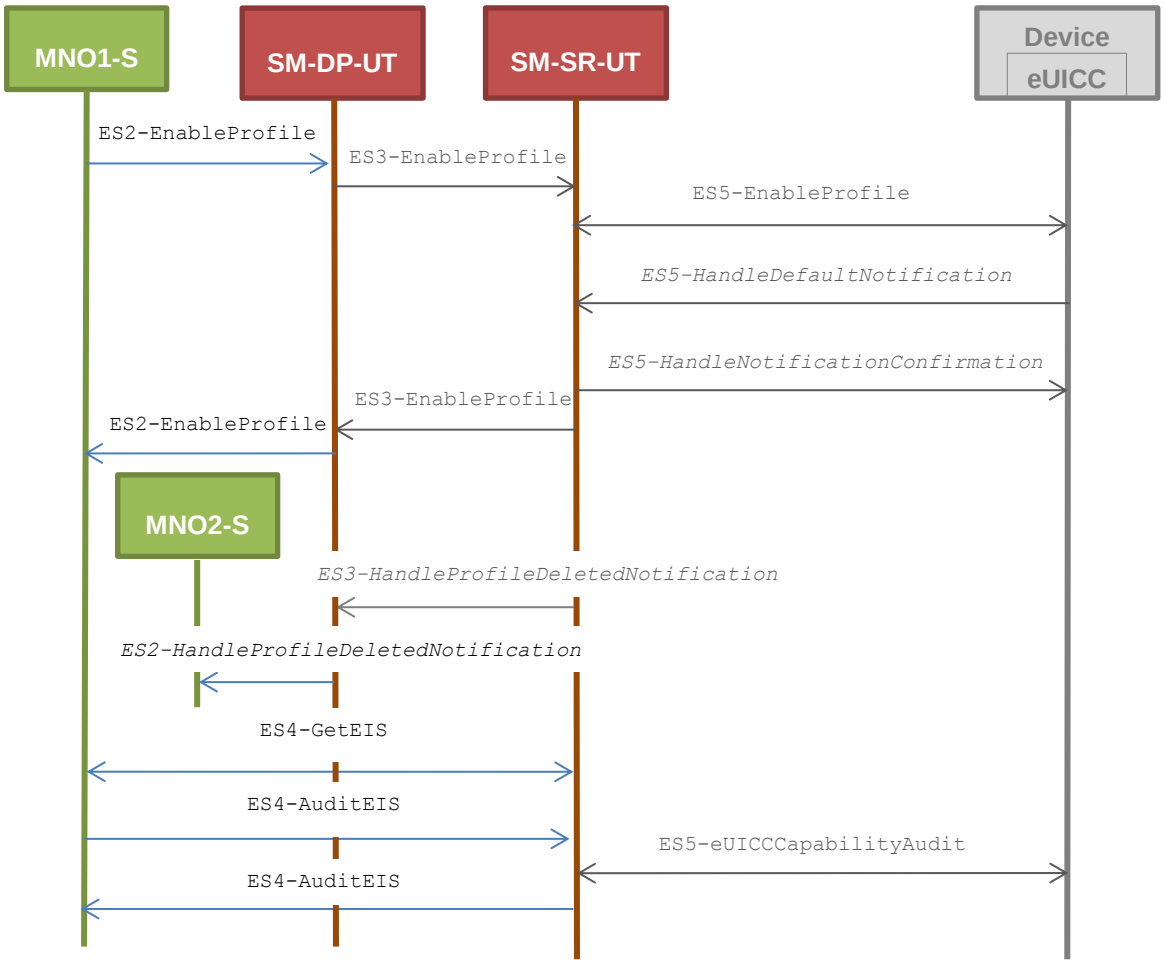
- The Profile downloaded, identified by `#NEW_ICCID`, SHALL be adapted to contain correct Connectivity Parameters (i.e. `#MNO1_CON_NAN`, `#MNO1_CON_LOGIN`, `#MNO1_CON_PWD`)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by `#NEW_ICCID`)
  - It 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 information is 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

5.3.3.2.2.2 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	The `Status` is equal to `#SUCCESS`	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)`		

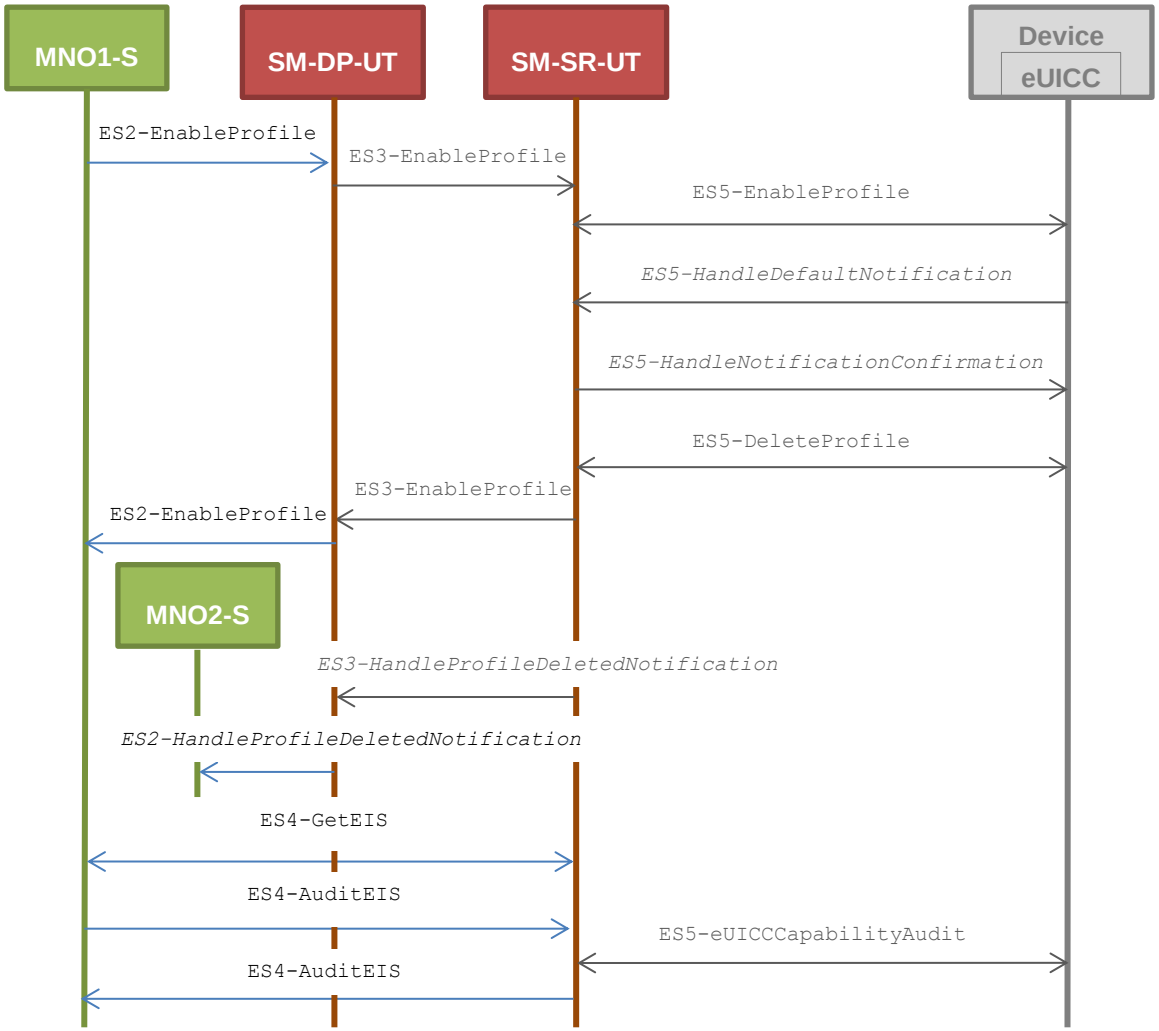
6	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R information is not present b. only Profiles related to the MNO1-S are present among which that identified by `#NEW_ICCID` c. the Profile identified by `#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 except that: a. the remaining memory is 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



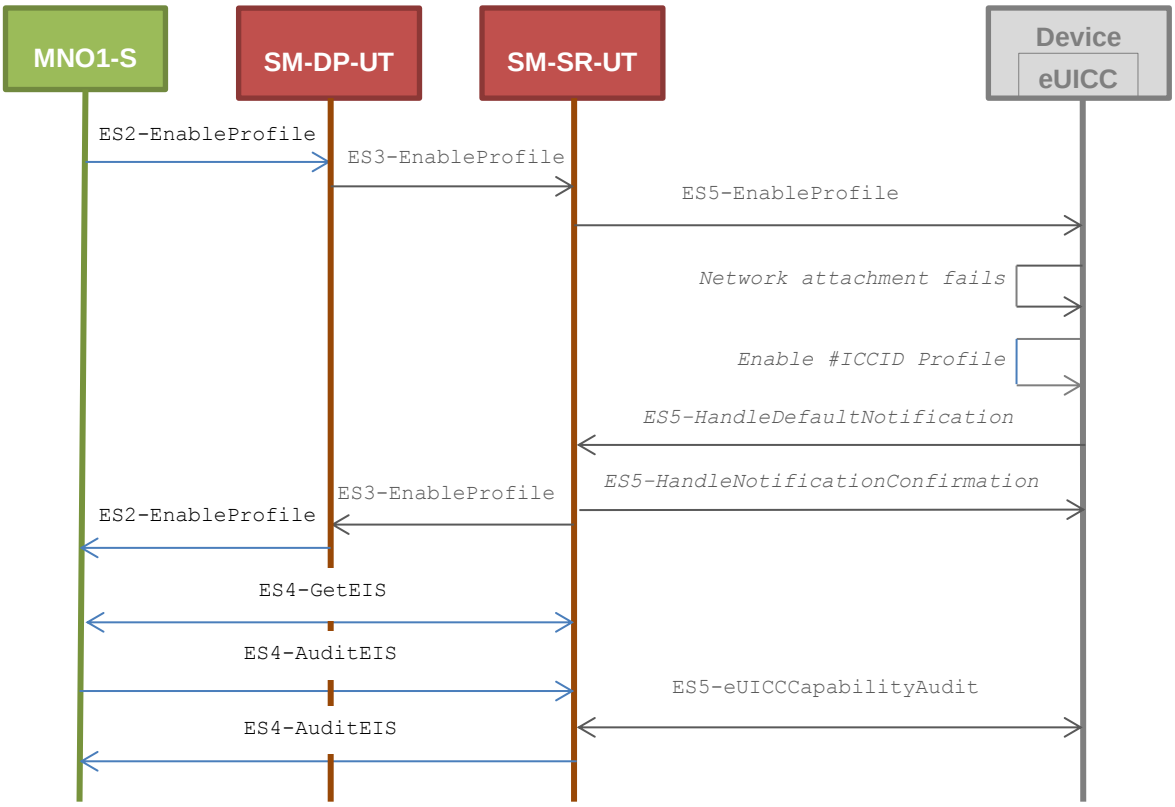
- The Profile downloaded, identified by `#NEW_ICCID`, SHALL be adapted to contain correct Connectivity Parameters (i.e. `#MNO1_CON_NAN`, `#MNO1_CON_LOGIN`, `#MNO1_CON_PWD`)
- The SM-SR-UT is able to communicate with the network linked to the new Profile of the eUICC (identified by `#NEW_ICCID`)
  - It 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	The Status is equal to #SUCCESS	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 information is 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` is 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_PROFILE 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 information is 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
- 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
- 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 contains bad Connectivity Parameters*
- *the Profile to be Disabled contains the POL1 "Disabling not Allowed"*

Referenced Requirements

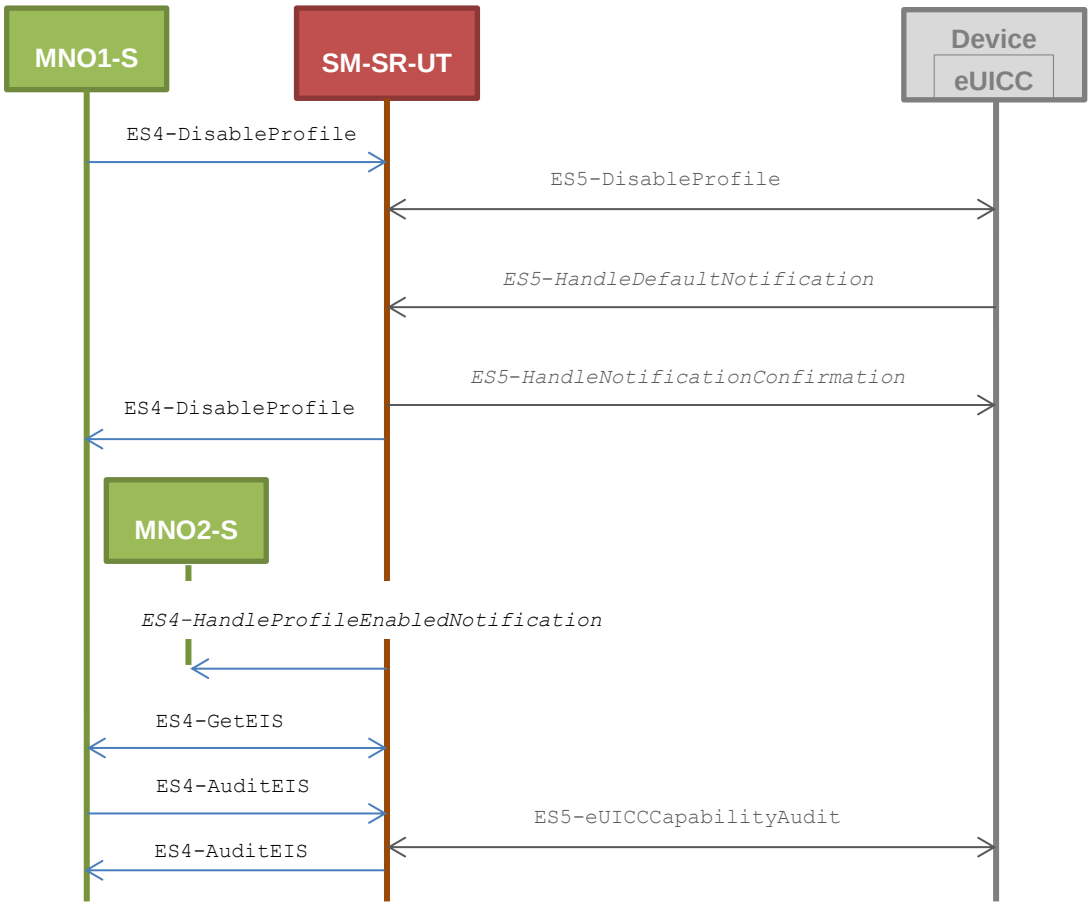
- PF_REQ2, PF_REQ5, PF_REQ6, PF_REQ7, PF_REQ25, PF_REQ28
- PROC_REQ9, PROC_REQ20
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

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



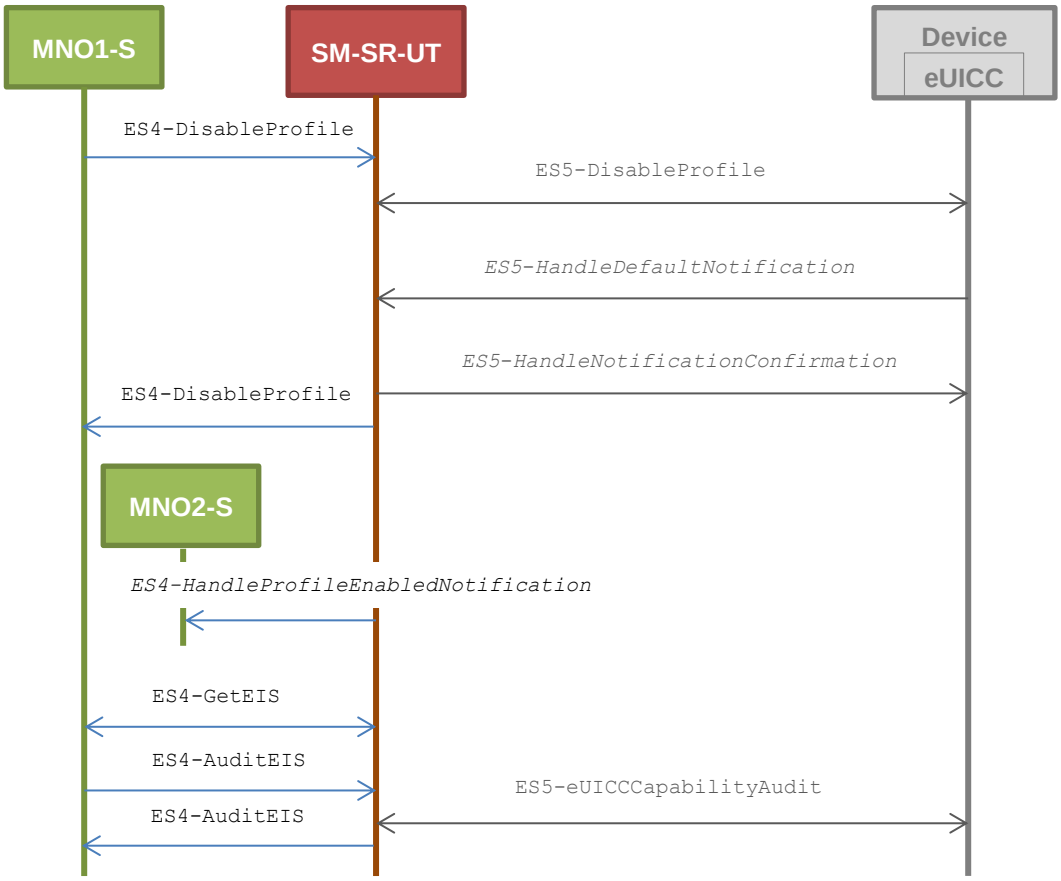
Initial Conditions

- 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
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 information is 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

5.3.4.2.1.2 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

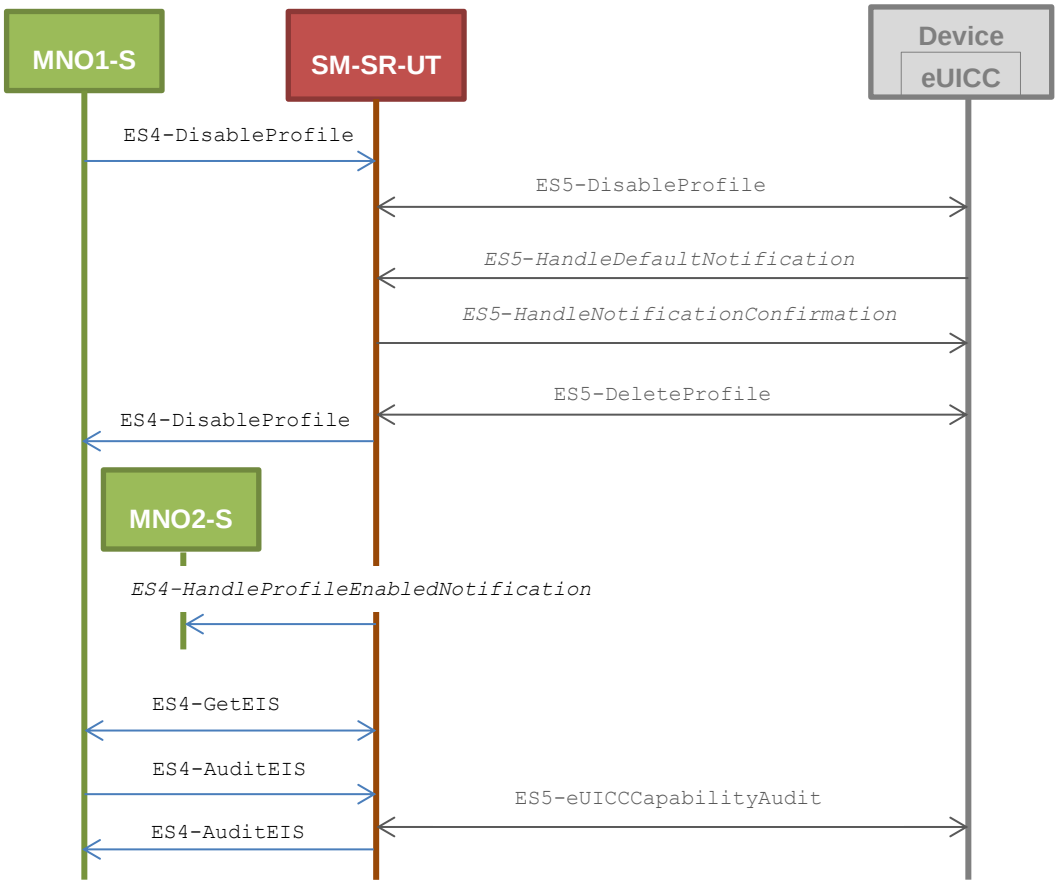
1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

2	*Wait until a response is received (the SM-SR-UT treatment MAY take several minutes)*			
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` 3- The `Reason code` is equal to `#RC_OBJ_EXIST`	PF_REQ2, PF_REQ5, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9, PROC_REQ20

Step	Direction	Sequence / Description	Expected result	REQ
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 information is 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 is 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

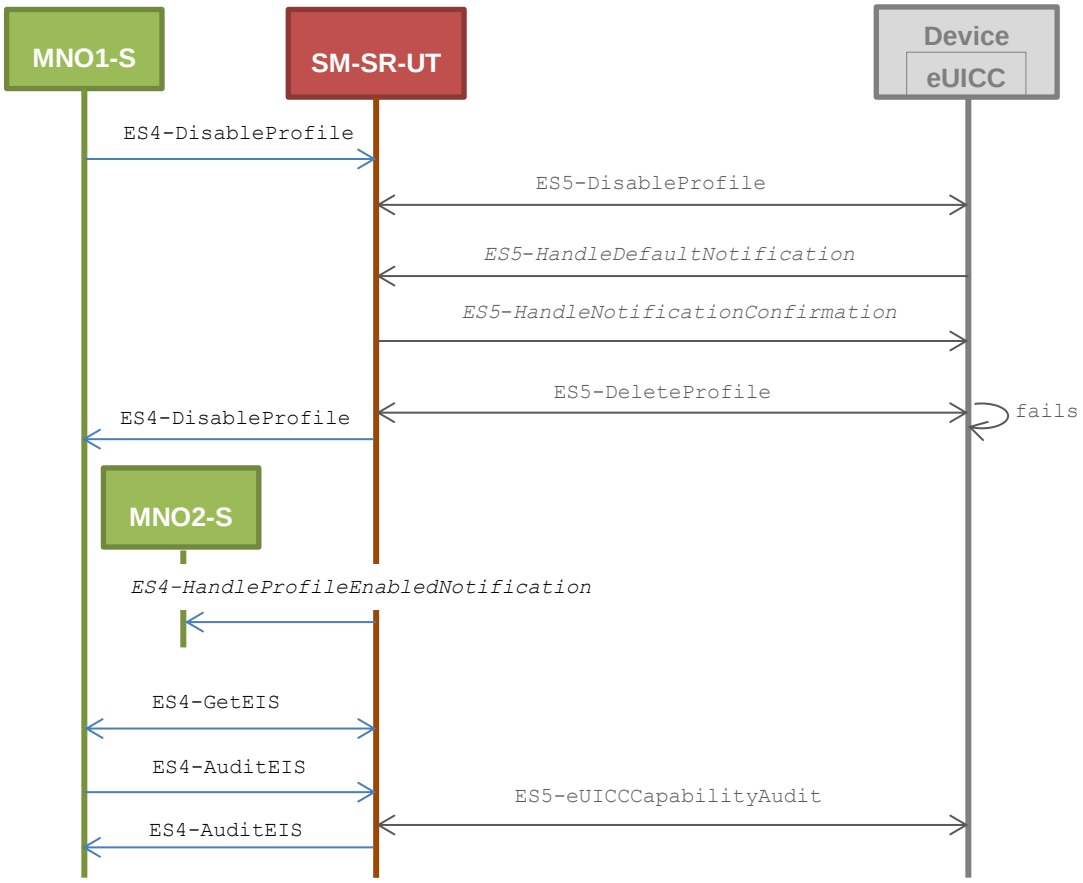
1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

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` 3- The `Reason code` is equal to `#RC_OBJ_EXIST`	PF_REQ2, PF_REQ5, PF_REQ6, PF_REQ25, PF_REQ28, EUICC_REQ27, EUICC_REQ29, PROC_REQ9, PROC_REQ20
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 information is 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` is 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

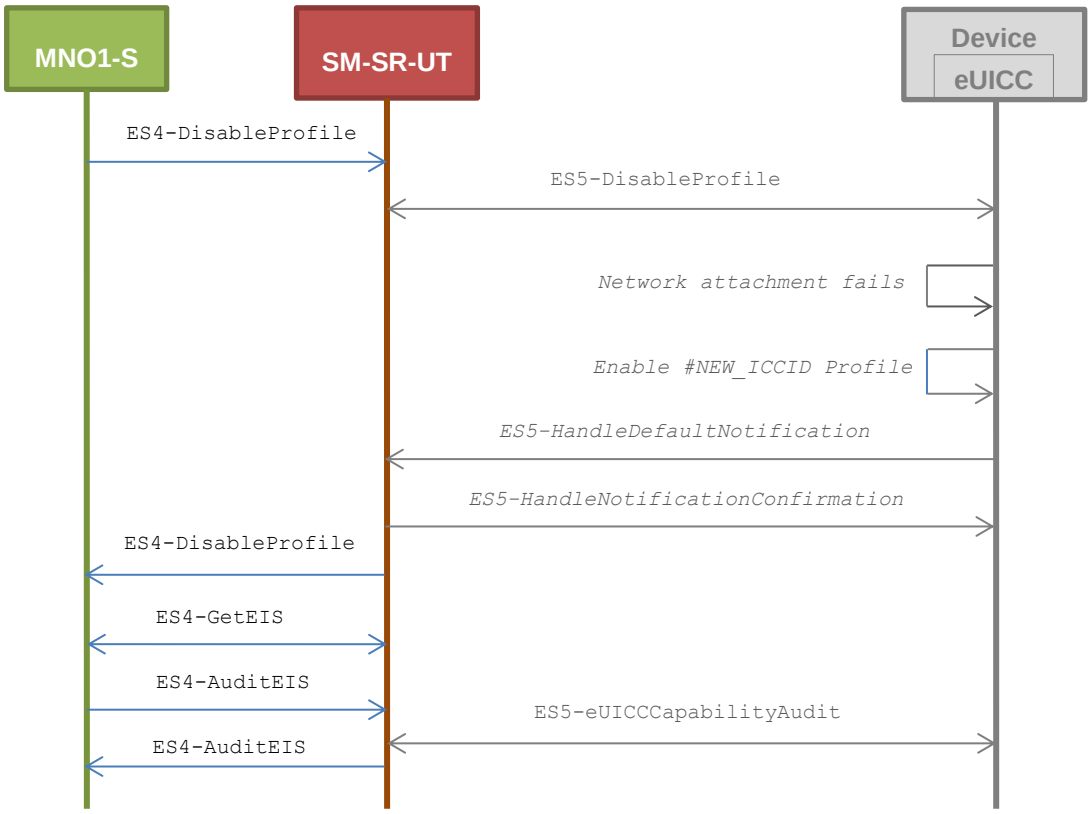
1	MNO1-S → SM-SR-UT	`SEND_REQ(`
		`ES4-DisableProfile,`
		`#EID_RPS,`
		`#NEW_ICCID_RPS)`		

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
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 information is 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_PROFILE` 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
4	MNO1-S → SM-SR-UT	`SEND_REQ( ES4-GetEIS, #EID_RPS)`		

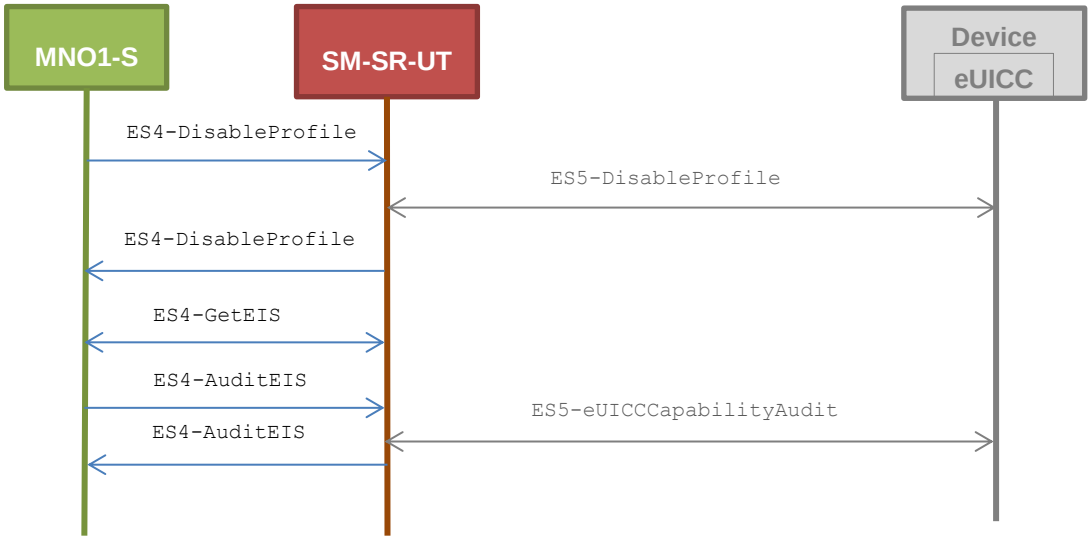
Step	Direction	Sequence / Description	Expected result	REQ
5	SM-SR-UT → MNO1-S	Send the `ES4-GetEIS` response	1- The `Status` is equal to `#SUCCESS` 2- The `EIS` is equal to `#EIS_RPS` except that: a. the ISD-R information is not present b. the Profile identified by `#NEW_ICCID` is Enabled	PM_REQ22

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 information is 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 contains bad Connectivity Parameters*

Referenced Requirements

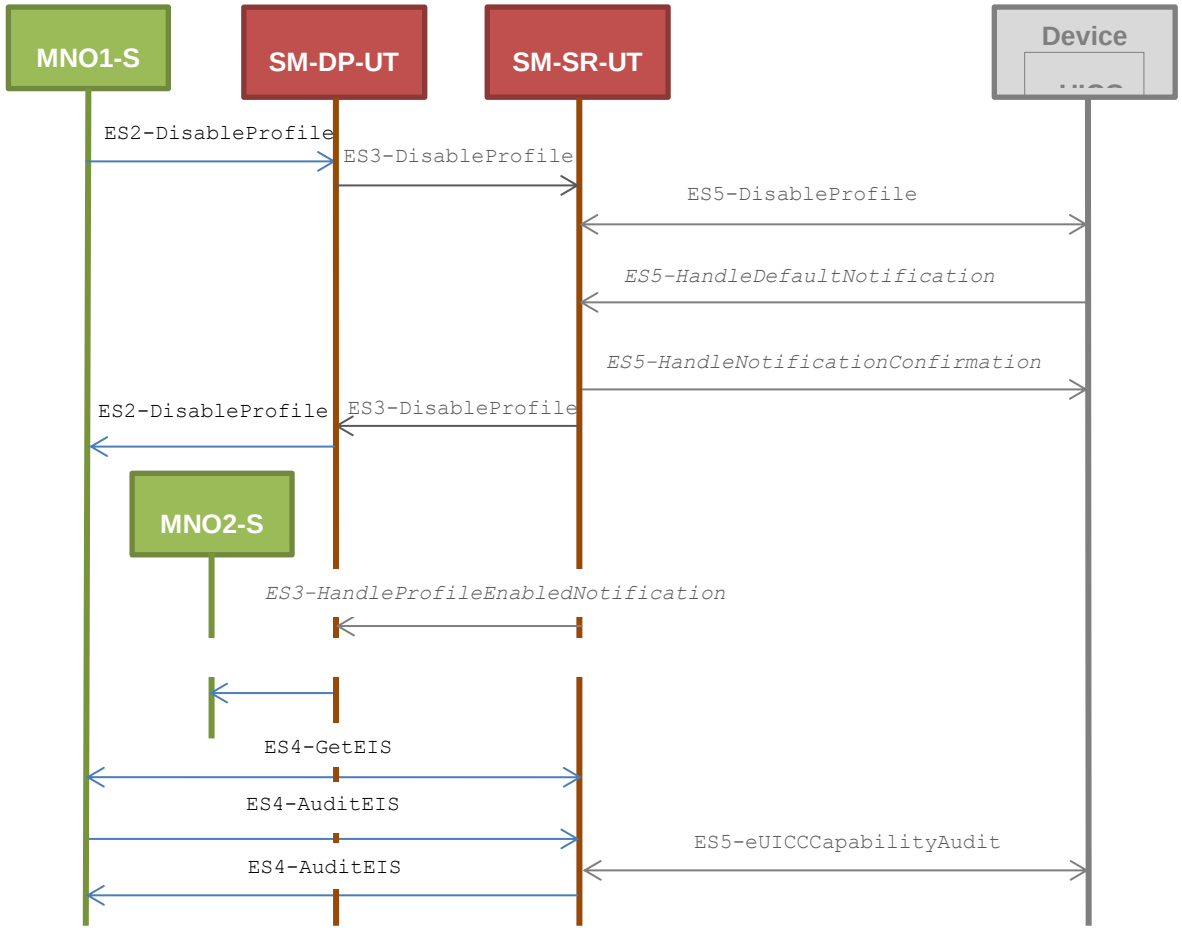
- PF_REQ2, PF_REQ5, PF_REQ7, PF_REQ13, PF_REQ16, PF_REQ19, PF_REQ22
- PROC_REQ10, PROC_REQ20
- PM_REQ22, PM_REQ26
- EUICC_REQ27, EUICC_REQ29

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)

5.3.4.2.2.1 Test Sequence N°1 – Nominal Case: Empty POL1 and POL2

Test Environment



Initial Conditions

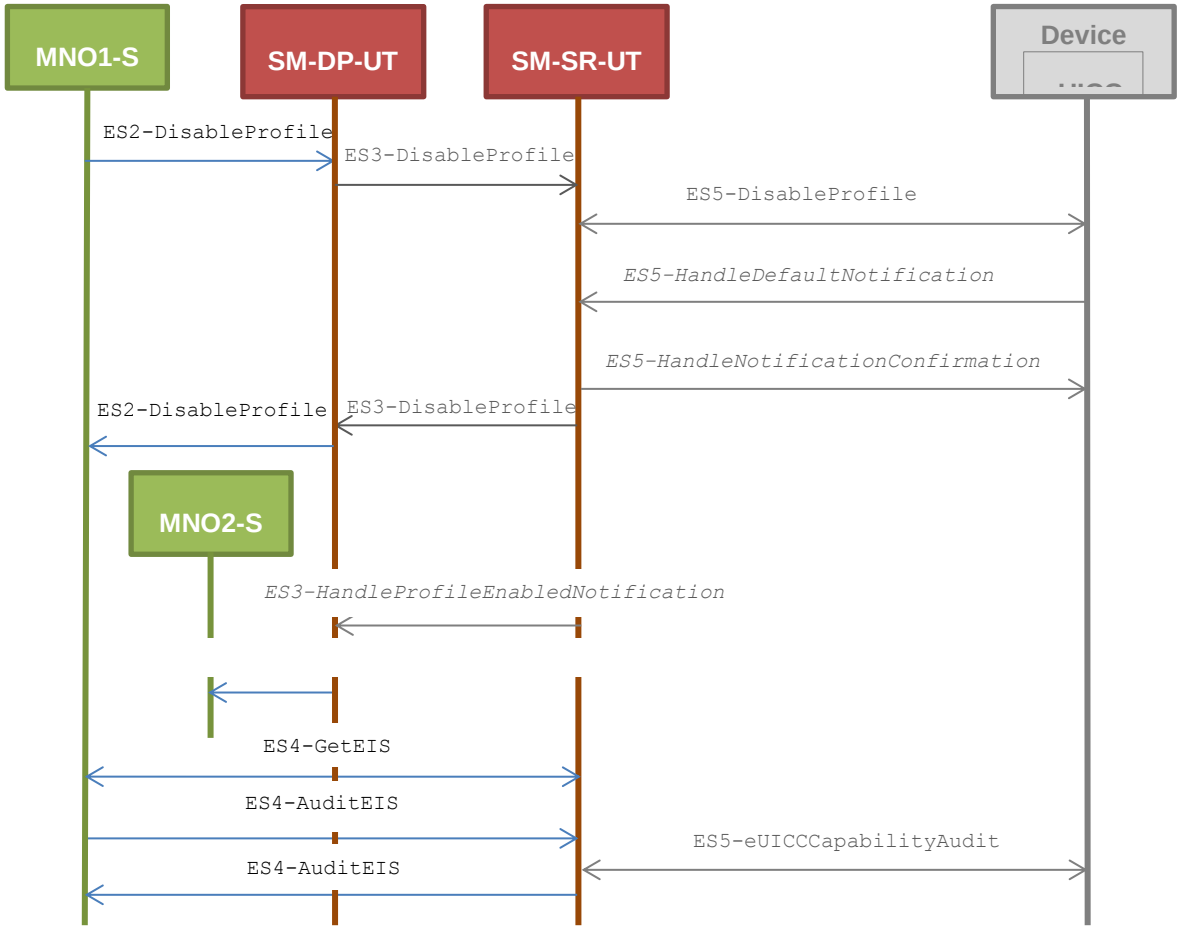
- POL1 and POL2 of the Profile identified by `#NEW_ICCID` do not contain any rules

- 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, 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 information is 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

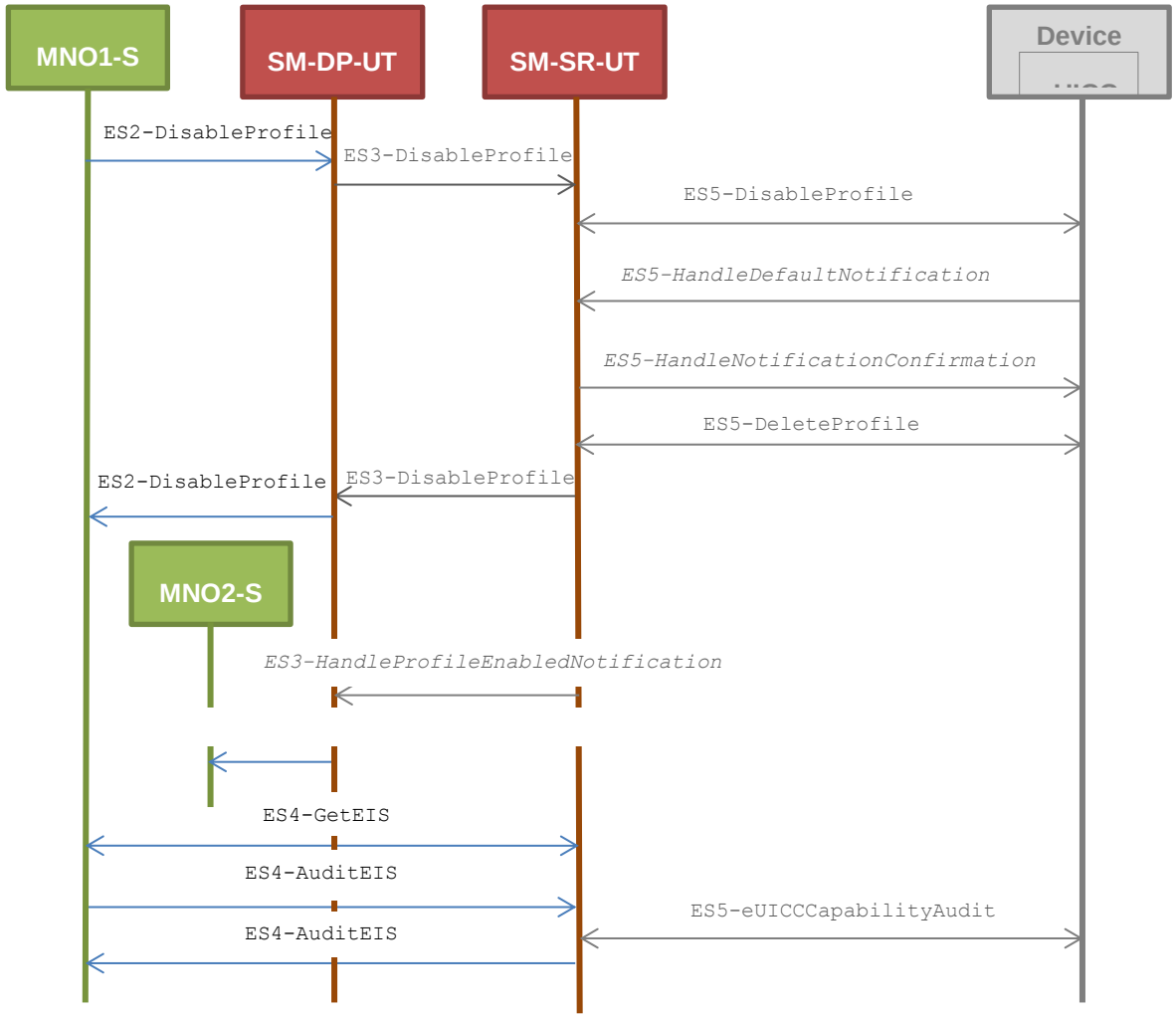
- POL1 of the Profile identified by `#NEW_ICCID` contains the rule "Profile deletion is mandatory when it is disabled"
- POL2 of the Profile identified by `#NEW_ICCID` allows disabling

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	1- The `Status` is equal to `#WARNING` 2- The `Subject code` is equal to `#SC_POL1` 3- The `Reason code` is equal to `#RC_OBJ_EXIST`	PF_REQ5, PF_REQ13, PF_REQ19, PF_REQ22, PROC_REQ10, PROC_REQ20, 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 information is 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` is 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

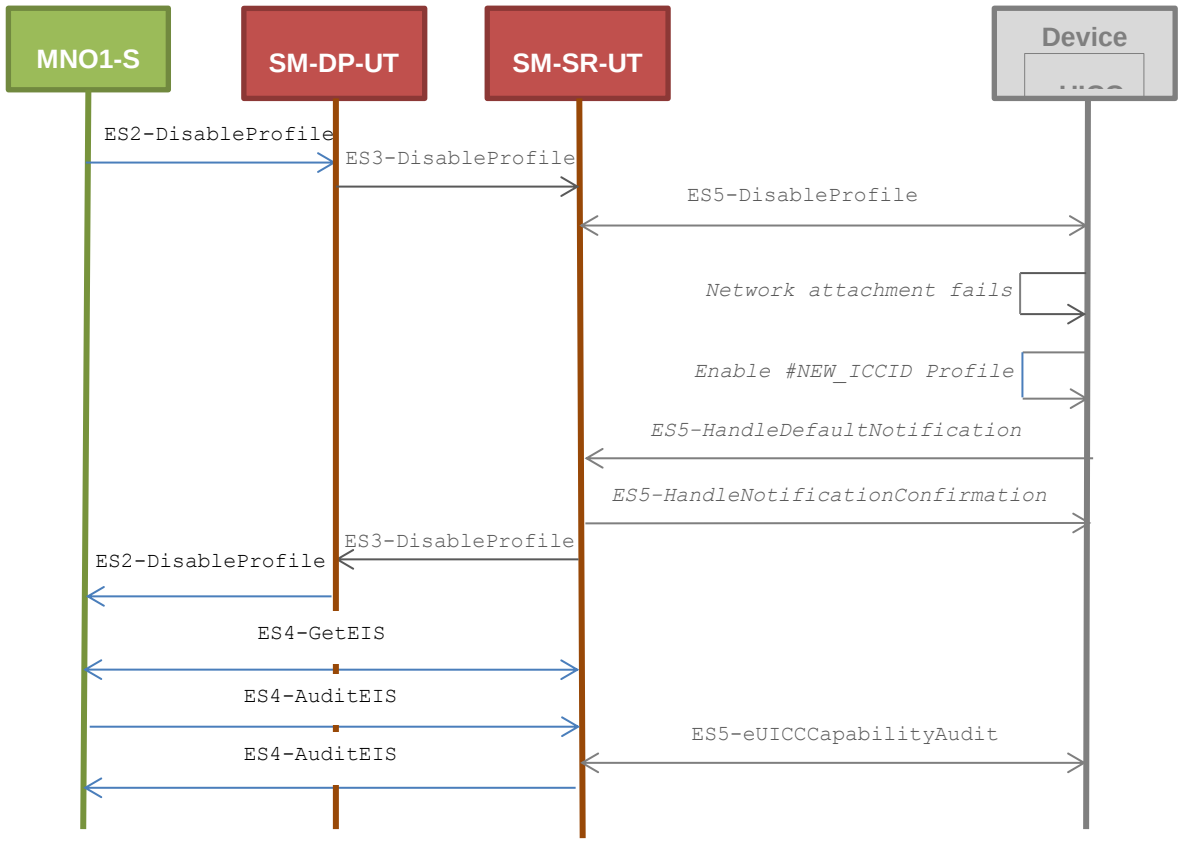
- POL1 of the Profile identified by `#NEW_ICCID` does not contain any rules
  - Disabling of the Profile is allowed
  - "Profile deletion is mandatory when it is disabled" is not set
- POL2 of the Profile identified by `#NEW_ICCID` contains the rule "Profile deletion is mandatory when it is disabled"

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	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_POL2 3- The Reason code is equal to #RC_OBJ_EXIST	PF_REQ5, PF_REQ13, PF_REQ19, PF_REQ22, PROC_REQ10, PROC_REQ20, 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 information is 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 is 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_PROFILE` 3- The `Reason code` is equal to `#RC_INACCESSIBLE`	PF_REQ5, PF_REQ13, PF_REQ19, PF_REQ22, PROC_REQ10, 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 information is 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

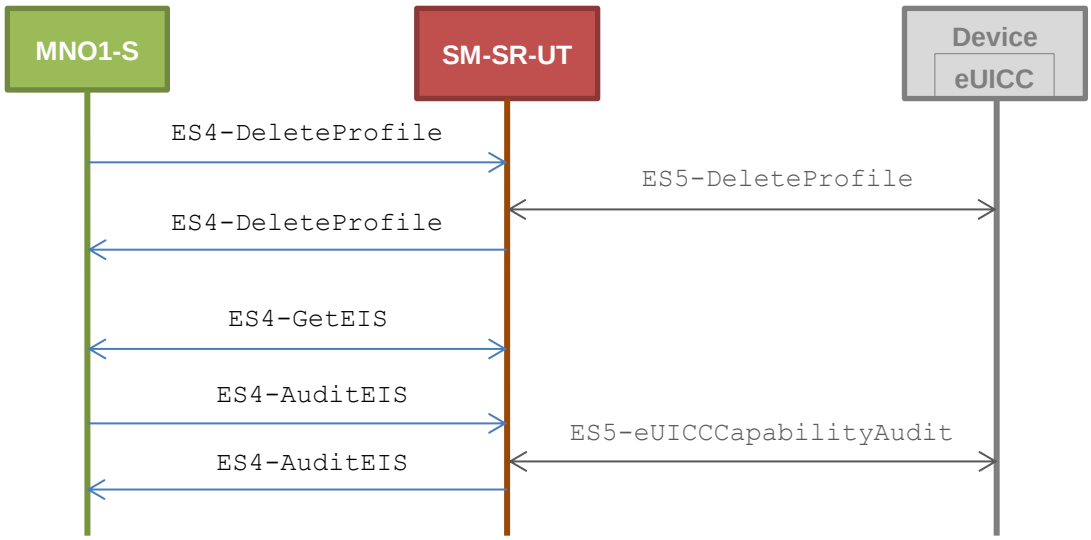
  - To download the new Profile (e.g. #PROFILE_PACKAGE), the test sequence defined in section 5.3.2.2.1.1 MAY be used
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by #ICCID)
  - It means that the SM-SR-UT knows the Connectivity Parameters 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

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 information is 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 is 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 information is 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

5.3.5.2.1.3 Test Sequence N°3 - Error Case: ISD-P not present on the eUICC

Initial Conditions

- The Profile identified by #NEW_ICCID is no more present in the eUICC (even though it is present in the EIS known to the SM-SR-UT)
- POL2 of the Profile identified by #NEW_ICCID do not contain any rules in the EIS
  - Deletion of the Profile is allowed

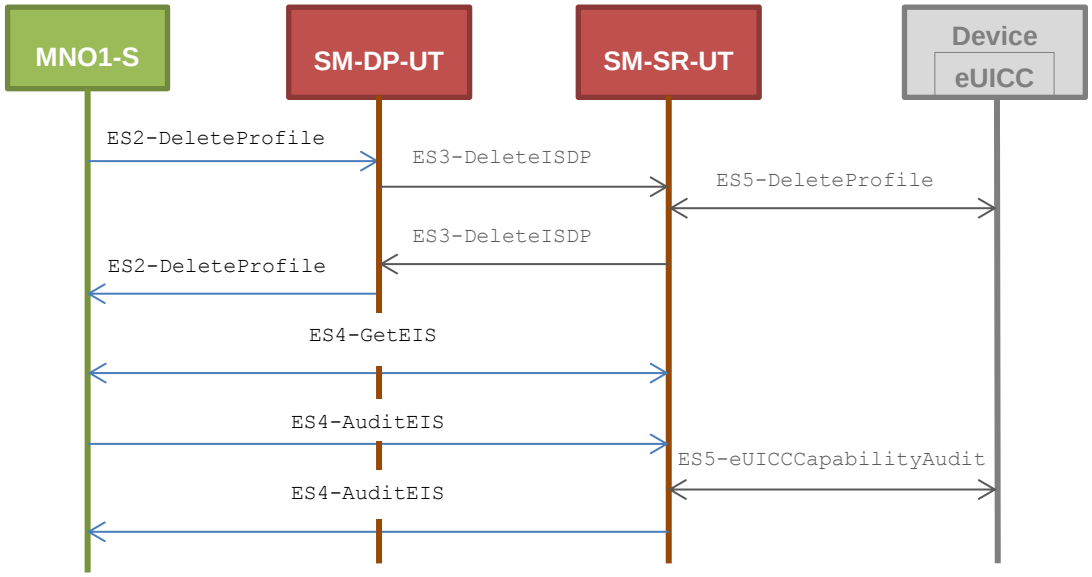
Step	Direction	Sequence / Description	Expected result	REQ
1	MNO1-S → SM-SR-UT	SEND_REQ(ES4-DeleteProfile, #EID_RPS, #NEW_ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment MAY take several minutes)*			
3	SM-SR-UT → MNO1-S	Send the ES4-DeleteProfile response	1- The Status is equal to #WARNING 2- The Subject code is equal to #SC_ISDP 3- The Reason code is equal to #RC_NOT_PRESENT 4- The euiccResponseData MAY be present. If any, it SHALL contain the POR generated by the eUICC (i.e. SW='6A88' or SW='6A82')	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 information is not present b. the Profile identified by #NEW_ICCID is no more present	PM_REQ22

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 information is 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 is 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)		

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-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 not present	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 information is 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

5.3.5.2.2.3 Test Sequence N°3 - Error Case: ISD-P not present on the eUICC

Initial Conditions

- The Profile identified by `#NEW_ICCID` is no more present in the eUICC (even though it is present in the EIS known to the SM-SR-UT)
- POL2 of the Profile identified by `#NEW_ICCID` do not contain any rules in the EIS
  - 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 #WARNING 2- The Subject code is equal to #SC_ISDP 3- The Reason code is equal to #RC_NOT_PRESENT 4- The euiccResponseData is not present	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 information is not present b. the Profile identified by #NEW_ICCID is no more present	PM_REQ22

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
- PM_REQ22, PM_REQ25
- PROC_REQ13
- SEC_REQ19

5.3.7.2 Test Cases

General Initial Conditions

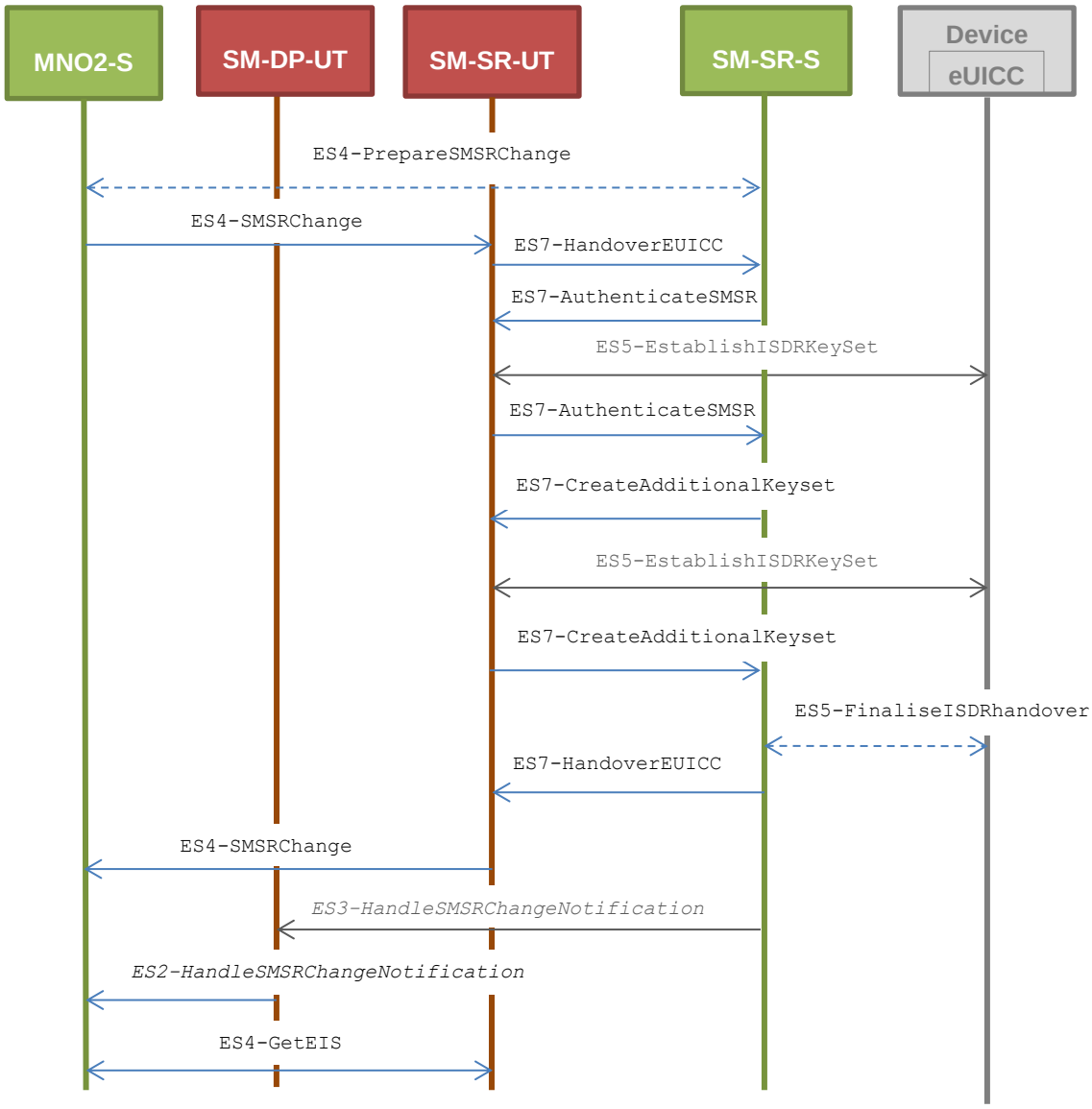
- `#MNO1_S_ID` well known to the SM-SR-UT
- `#MNO2_S_ID` well known to the SM-SR-UT
- The Profile identified by `#ICCID` is owned by MNO2-S and is in Enabled state
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by `#ICCID`)
  - It 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
- 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 Profiles present in the #EIS_RPS SHALL contain an smdp-id equal to #SM_DP_ID
- 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 except that the ISD-R keys values are empty	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) The "HostId" parameter SHALL be set to an empty value.		

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 → SM-SR-S	Send the `ES7-CreateAdditionalKeyset` response	1- The `Status` is equal to `#SUCCESS` 2- The `derivation random` is empty 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 → MNO2-S	Send the `ES4-SMSRChange` response	The `Status` is equal to `#SUCCESS`	EUICC_REQ36, PROC_REQ13
11	SM-SR-S → SM-DP-UT	`SEND_NOTIF( ES3-HandleSMSRChangeNotification, #EIS_RPS, #TIMESTAMP_RPS)` Note: The `#EIS_RPS` shall: - not contain the ISD-R keysets - At most contain Profiles related to the `#SM_DP_ID`		
12	SM-DP-UT → MNO2-S	Send the `ES2-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` except that: a. The ISD-R information is not provided b. At most Profiles owned by the MNO2-Sare present 2- The completion timestamp is equal to `#TIMESTAMP_RPS`	EUICC_REQ33, EUICC_REQ34, PROC_REQ13

Step	Direction	Sequence / Description	Expected result	REQ
13	MNO2-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
14	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_ID_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 except that the ISD-R keys values are empty	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

Step	Direction	Sequence / Description	Expected result	REQ
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) The "HostId" parameter SHALL be set to an empty value.		
7	*Wait until a response is received (the SM-SR-UT treatment MAY take several minutes)*			
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 and different from an empty value (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 → MNO2-S	Send the ES4-SMSRChange response	The Status is equal to #SUCCESS	EUICC_REQ36, PROC_REQ13

Step	Direction	Sequence / Description	Expected result	REQ
11	SM-SR-S → SM-DP-UT	SEND_NOTIF(ES3-HandleSMSRChangeNotification, #EIS_RPS, #TIMESTAMP_RPS) Note: The #EIS_RPS shall: - Not contain the ISD-R keysets - At most contain Profiles related to #SM_DP_ID		
12	SM-DP-UT → MNO2-S	Send the ES2-HandleSMSRChangeNotification notification	1- The EIS parameter is equal to #EIS_RPS except that: a. The ISD-R information is not provided b. At most Profiles owned by the MNO2-S are present 2- The completion timestamp is equal to #TIMESTAMP_RPS	EUICC_REQ33, EUICC_REQ34, PROC_REQ13
13	MNO2-S → SM-SR-UT	SEND_REQ(ES4-GetEIS, #EID_RPS)		
14	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_ID_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 except that the ISD-R keys values are empty	EUICC_REQ36, EUICC_REQ39, PROC_REQ13

Step	Direction	Sequence / Description	Expected result	REQ
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)*			
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 and different form an empty value (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)		

Step	Direction	Sequence / Description	Expected result	REQ
10	SM-SR-UT → MNO2-S	Send the `ES4-SMSRChange` response	The `Status` is equal to `#SUCCESS`	EUICC_REQ36, PROC_REQ13

11	SM-SR-S → SM-DP-UT	`SEND_NOTIF(`
		`ES3-HandleSMSRChangeNotification,`
		`#EIS_RPS,`
		`#TIMESTAMP_RPS)`
		Note: The `#EIS_RPS` shall:
		- Not contain the ISD-R keysets
		- At most contain Profiles related to `#SM_DP_ID`		

12	SM-DP-UT → MNO2-S	Send the `ES2-HandleSMSRChangeNotification` notification	1- The `EIS` parameter is equal to `#EIS_RPS` except that: a. The ISD-R information is not provided b. At most Profiles owned by the MNO2-S are present 2- The completion timestamp is equal to `#TIMESTAMP_RPS`	EUICC_REQ33, EUICC_REQ34, PROC_REQ13

13	MNO2-S → SM-SR-UT	`SEND_REQ(`
		`ES4-GetEIS,`
		`#EID_RPS)`		

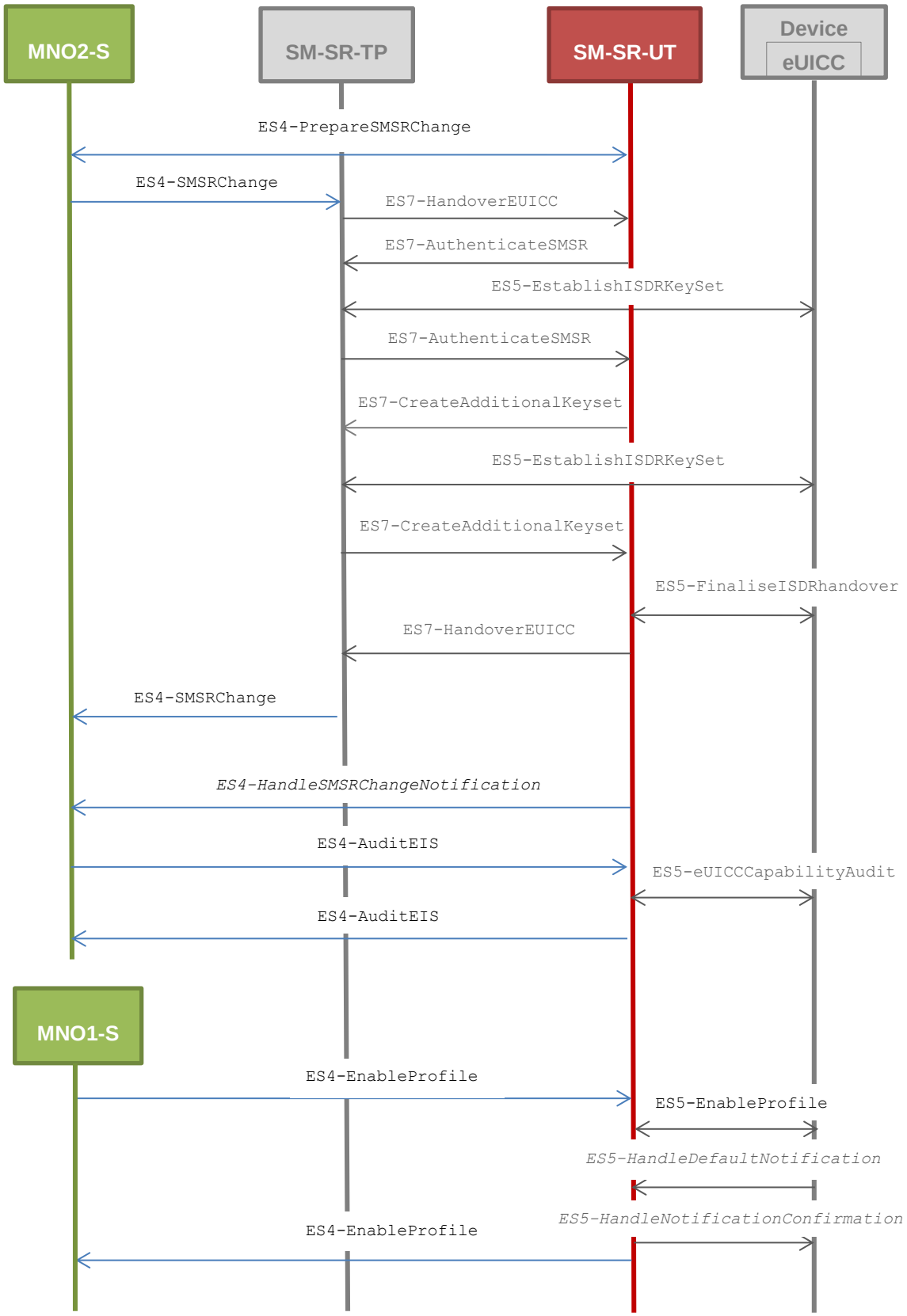
14	SM-SR-UT → MNO2-S	Send the `ES4-GetEIS` response	1- The Status is equal to `#FAILED` 2- The Subject code is equal to `#SC_EID` 3- The Reason code is equal to `#RC_ID_UNKNOWN`	PM_REQ22, SEC_REQ19

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


MNO2-S
SM-SR-TP
SM-SR-UT
Device
eUICC
ES4-PrepareSMSRChange
ES4-SMSRChange
ES7-HandoverEUICC
ES7-AuthenticateSMSR
ES5-EstablishISDRKeySet
ES7-AuthenticateSMSR
ES7-CreateAdditionalKeyset
ES5-EstablishISDRKeySet
ES7-CreateAdditionalKeyset
ES5-FinaliseISDRhandover
ES7-HandoverEUICC
ES4-SMSRChange
ES4-HandleSMSRChangeNotification
ES4-AuditEIS
ES5-eUICCCapabilityAudit
ES4-AuditEIS
MNO1-S
ES4-EnableProfile
ES5-EnableProfile
ES5-HandleDefaultNotification
ES5-HandleNotificationConfirmation
ES4-EnableProfile

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.

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).

Referenced Requirements

- PF_REQ2, PF_REQ7
- EUICC_REQ25, EUICC_REQ35, EUICC_REQ36, EUICC_REQ37, EUICC_REQ38, EUICC_REQ39, EUICC_REQ40
- 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`
- All Profiles present in the `#EIS_RPS` SHALL NOT contain any smdp-id
- 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…)
- 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
- 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
- The SM-SR-UT is able to communicate with the network linked to the Profile identified by `#NEW_ICCID`
  - It means that the SM-SR-TP knows the Connectivity Parameters of the MNO's network related to the Disabled Profile (i.e. `#MNO1_CON_NAN`, `#MNO1_CON_LOGIN`, `#MNO1_CON_PWD`)

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)*			
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, PROC_REQ13, PF_REQ2
6	SM-SR-UT → MNO2-S	SEND_NOTIF(ES4-HandleSMSRChangeNotification, #EIS_RPS, #TIMESTAMP_RPS) Note: The #EIS_RPS shall: - Not contain the ISD-R information - Only contain Profiles owned by the MNO2-S		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	1- The Status is equal to #SUCCESS 2- The EIS is equal to #EIS_RPS except that: a. the ISD-R information is not present b. only Profiles related to the MNO2-S are present	PM_REQ25, PROC_REQ13, PF_REQ7, PF_REQ2

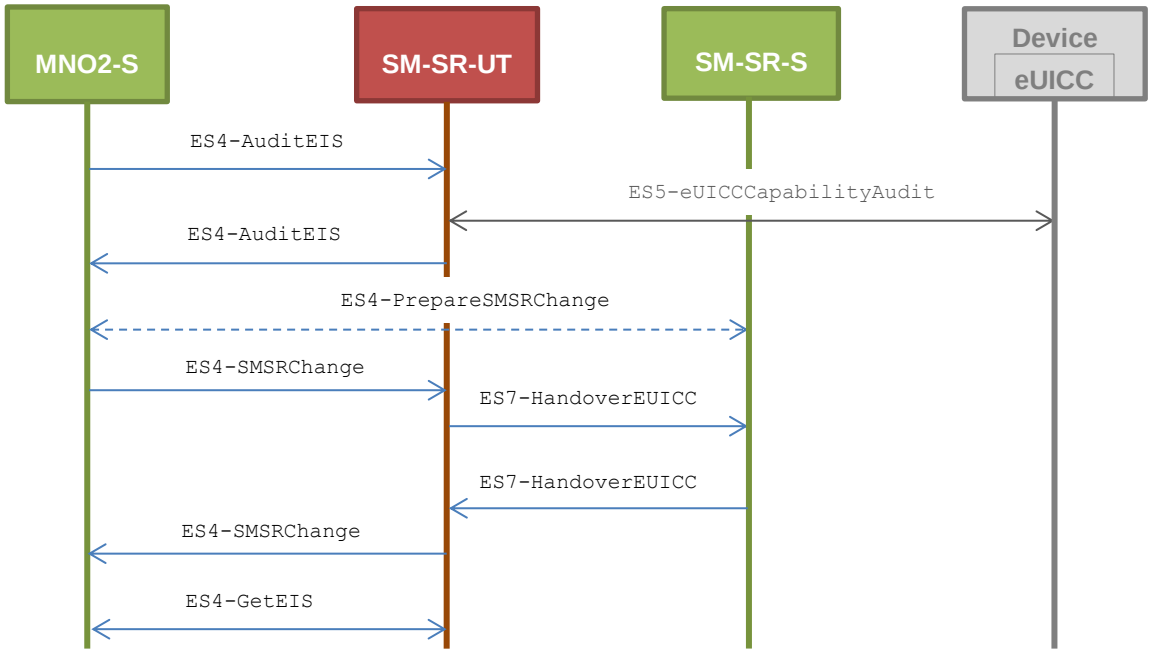
Step	Direction	Sequence / Description	Expected result	REQ
10	MNO1-S → SM-SR-UT	SEND_REQ(ES4-EnableProfile, #EID_RPS, #NEW_ICCID_RPS) See Note 2		
11	*Wait until a response is received (the SM-SR-UT treatment MAY take several minutes)*			
12	SM-SR-UT → MNO1-S	Send the ES4-EnableProfile response	The Status is equal to #SUCCESS	
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 Profiles present in the `#EIS_RPS` SHALL NOT contain any smdp-id
- 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

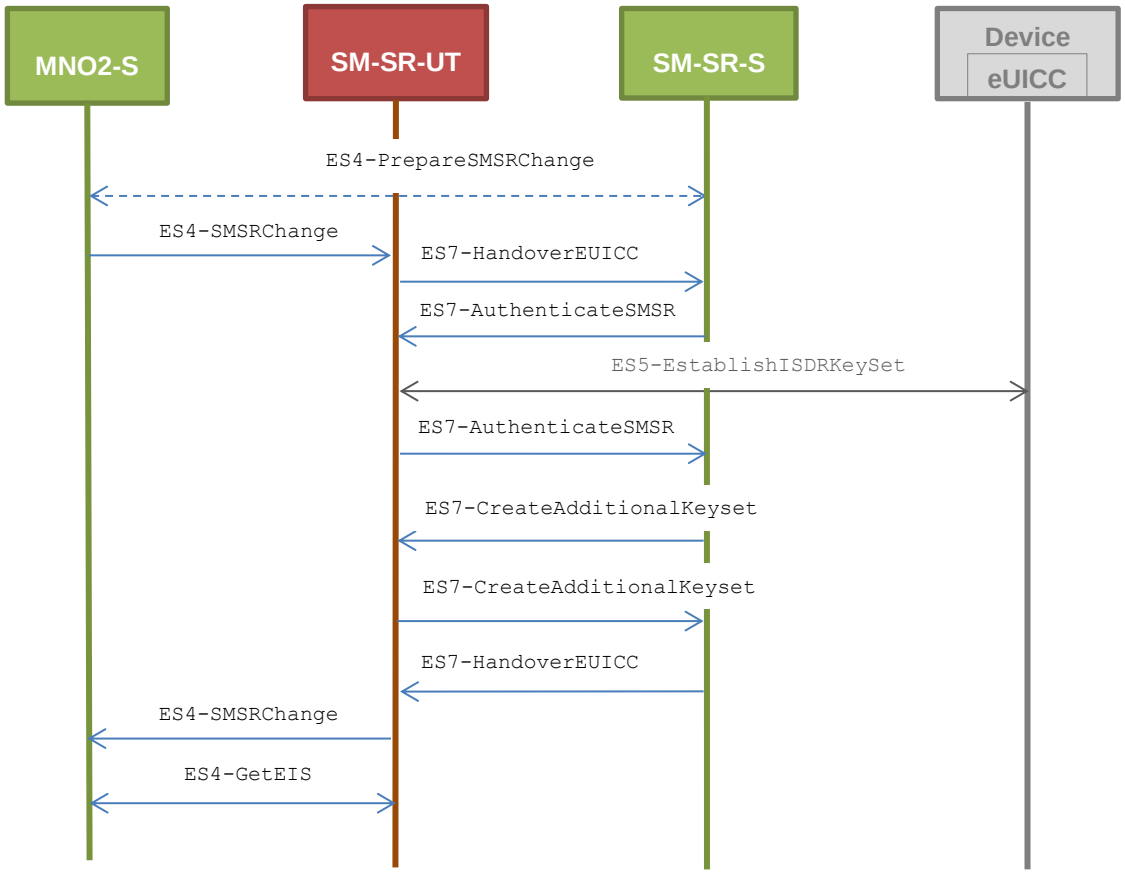
Step	Direction	Sequence / Description	Expected result	REQ
1	MNO2-S → SM-SR-UT	SEND_REQ(ES4-AuditEIS, #EID_RPS, #ICCID_RPS)		
2	*Wait until a response is received (the SM-SR-UT treatment MAY take several minutes)*			
3	SM-SR-UT → MNO2-S	Send the ES4-AuditEIS response	The Status is equal to #SUCCESS	PF_REQ2, 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} c. the ISD-R keys values are empty	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 Profiles present in the #EIS_RPS SHALL NOT contain any smdp-id
- 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 except that the ISD-R keys values are empty	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) The "HostId" parameter SHALL be set to an empty value.		

Step	Direction	Sequence / Description	Expected result	REQ
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 4- derivationRandom is empty 5- The receipt is empty	EUICC_REQ36, EUICC_REQ38, EUICC_REQ39, PROC_REQ13
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

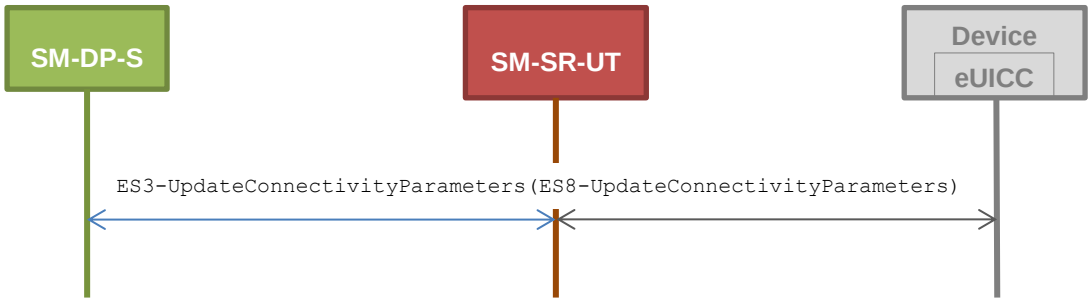
- #MNO1_S_ID well known to the SM-SR-UT
- #MNO2_S_ID well known to the SM-SR-UT
- The Profile identified by #ICCID is owned by MNO2-S and is in Enabled state
- The SM-SR-UT is able to communicate with the network linked to the default Enabled Profile of the eUICC (identified by #ICCID)
  - It means that 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



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
Note 1: The C-APDUs generated by the method SCP03_SCRIPT SHALL be set into the RPS element <connectivityParameters>				

6 Test Specifications

Some test specifications related to the eUICC ecosystem have been developed by external organisations (e.g. TCA (former SIMAlliance)). These organisations defined their own requirements for test benches, test applicability and pass criteria.

This section lists the test specifications that relate to the GSMA Remote Provisioning Architecture for Embedded UICC - Technical Specification [2].

6.1 TCA eUICC Profile Package Test Specification

The eUICC SHALL take test cases defined in the TCA eUICC Profile Package: Interoperable Format Test Specification [17] in order to check its compliance with the SIMAlliance eUICC Profile Package: Interoperable Format Technical Specification [16].

All the mandatory test cases are applicable according to the applicability of the referred TCA test specification.

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)

```
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
DATA	22 0E 80 50 30 00 08 01 02 03 04 01 02 03 04 00
DCS	F6
DEST_ADDR	05 85 02 82 F2
DEST_ADDR2	05 85 03 83 F3
DEST_ADDR3	05 85 03 83 F4
DIALING_NUMBER	33 86 99 42 11 F0
DIALING_NUMBER_INITIAL	33 86 99 00 00 F0
DNS_IP	21 01 02 03 04
DNS_PORT	00 35
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
ICCID_UNKNOWN	89 01 99 99 00 00 55 77 78 75
INIT_MAC	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
INIT_MAC_32	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
IP_VALUE	7F 00 00 01
IP_VALUE2	7F 00 00 02
ISD_P_AID1	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00 see Note 1
ISD_P_ID1	00 00 10

Constant name	Value in hexadecimal string
	see Note 3
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_AID_UNKNOWN	A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 99 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
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
NEW_SCP81_PSK	18 94 D8 3C 1F BF 38 27 92 76 B7 0F 8F 02 61 16
NAN_VALUE	09 47 53 4D 41 65 55 49 43 43
PID	11
PPK-ENC	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
PPK-ENC_32	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
PPK-MAC	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
PPK-MAC_32	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
PPK-RMAC	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
PPK-RMAC_32	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
PSK_DEK	01 02 03 04 05 06 07 08 01 02 03 04 05 06 07 08
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
SC3_NO_DR_HOST	0D

Constant name	Value in hexadecimal string
SCP03_KVN	30
SCP80_NEW_KVN	0E see Note 2
SM-SR_FQDN	73 6D 73 72 2E 65 78 61 6D 70 6C 65 2E 63 6F 6D Note: meaning 'smsr.example.com'
SPI_VALUE	16 39
SPI_VALUE_NO_POR	16 00
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)* *Note 3: SHALL correspond to the identifier of #ISD_P_AID1 (i.e. digits 15 to 20 of PIX of ISD-P)*	

Table 8: Hexadecimal Constants

B.2 ASCII Constants

Here are the ASCII constants values used in this document:

Constant name	Value in ASCII
BIG_MEM	9999999
CONTENT_TYPE	Content-Type: application/vnd.globalplatform.card-content-mgt-response;version=1.0
EUM_S_ID	1.3.6.1.4.1.46304.992.1.1
EXPIRED	Expired
FAILED	Failed
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	1.3.6.1.4.1.46304.992.1.2

Constant name	Value in ASCII
MNO2_S_ID	1.3.6.1.4.1.46304.992.1.3
MSISDN1	447112233445
MSISDN2	447112233446
MSISDN3	447112233447
M2MSP1_S_ID	1.3.6.1.4.1.46304.992.1.4
M2MSP2_S_ID	1.3.6.1.4.1.46304.992.1.5
POST_URI	POST /gsma/adminagent HTTP/1.1
POST_URI_NOTIF	POST /gsma/adminagent?msg=#NOTIF_PROFILE_CHANGE HTTP/1.1
POST_URI_NOTIF_DEFAULT	POST /gsma/adminagent?msg=#NOTIF_PROFILE_CHANGE_DEFAULT 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_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	2.1
RC_INVALID_DEST	3.4
RC_MEMORY	4.8
RC_NOT_ALLOWED	1.2
RC_OBJ_EXIST	3.6
RC_REFUSED	3.8
RC_UNKNOWN	3.9
RC_NOT_PRESENT	4.6
RC_TTL_EXPIRED	5.3
RC_VERIFICATION_FAILED	6.1
RPS_CONTEXT_ID	Context-XYZ- R2xvcmlh
RPS_MESSAGE_ID	http://example.com/uniqueMessageId- WW9sYW5kYQ
RPS_TRANSACTION_ID	tx5347502e3131
SC_CERT_REQ	8.5.1
SC_ECASD	8.5.2
SC_EID	8.1.1
SC_EIS	8.6
SC_EUICC	8.1
SC_FUN_PROV	1.2

Constant name	Value in ASCII
SC_EXT_RES	1.4
SC_FUNCTION	1.6
SC_FUN_REQ	1.1
SC_PLMA	8.2.7
SC_SD_AID	8.3.1
SC_ISDP	8.3
SC_ISDR	8.4
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_SM_SR_CERT	8.7.1
SC_SR_CERTIF	8.5.3
SC_SUB_ADDR	8.2.6
SHORT_VALIDITY_PERIOD	30
SMALL_MEM	999
SM_DP_S_ID	1.3.6.1.4.1.46304.992.1.6
SM_SR_S_ID	1.3.6.1.4.1.46304.992.1.7
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_ID	The Identifier of the default ISD-P (digits 15 to 20 of PIX of ISD-P) pre-installed on the eUICC (this corresponds to the #DEFAULT_ISD_P_AID).
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_SC2B	Value of the TLV '7F21' - CASD certificate (of the default Enabled Profile) allowing to confidentialy setup keys using scenario #2.B.
CASD_CERTIFICATE_SC3	Value of the TLV '7F21' - CASD certificate (of the default Enabled Profile) allowing to confidentialy setup keys using scenario #3.
EID	Content of the TLV '5A' available on the ECASD.
EUM_OID	EUM_OID (i.e. value of the tag '42' – CA Identifier of the ECASD certificate) *Note: When present in the ECASD, this value SHALL be encoded as a value part of the DER_TLV_OID (e.g. 0x2B….).* *When present in the EIS, this value SHALL be encoded as a dotted number notation (e.g. "1.3.6….".).*
EUM_SUBJECT_KEY_ID	Subject Key Identifier of the EUM Certificate (i.e. value of the tag 'C9' of the ECASD certificate).
EUM_PK_ECDSA	Public key of the EUM used for ECDSA.
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).

eUICC setting name	Description
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
CLEANUP_DELAY	A delay within which an SM-SR platform may delete an ISD-P whose creation was not confirmed by the eUICC. See Note 2.

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 {L} #EUM_OID`
	`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 {L}`
	`C0 01 01`
	`C1 01 01`
	`C2 01 01`
	`C9 14 #EUM_SUBJECT_KEY_ID`
	`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

Platform setting name	Description
EUM_S_CERT_ID_ECDSA	The certificate subject name of the EUM-S used for ECDSA. The use of the certificate subject name in the EIS implicitly means that all platforms under test (i.e. SM-DP-UT and SM-SR-UT) know the #EUM_S_PK_ECDSA (this public key is part of the #EUM_S_CERT_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
EUM_S_CERT_ECDSA	X.509 Certificate of the EUM-S used for ECDSA. Subject name of this certificate is set to #EUM_S_CERT_ID_ECDSA.
EXPIRED_ECASD_CERT	An expired certificate CERT.ECASD.ECKA of a simulated eUICC. The TLV '7F21' SHALL contain: 93 01 09 42 {L} #EUM_OID 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 {L} C0 01 01 C1 01 01 C2 01 01 C9 14 #EUM_SUBJECT_KEY_ID 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 {L} #CI_OID 5F 20 01 01 95 01 82 5F 24 04 20 00 01 01 73 {L} C8 01 02 C9 14 #CI_SUBJECT_KEY_ID 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
KEY_SECURED	The #KEY encrypted with a transport key (as defined in GSMA Remote Provisioning Architecture for Embedded UICC-Technical Specification [2]). The transport key value and the related algorithm can be freely chosen by the SM-SR-UT. see Note 2
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

Platform setting name	Description
	signature. The content of the TLV is the same as #VALID_SM_DP_CERTIFICATE. 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
PF_ICCID_TO_DOWNLOAD	The ICCID of a single profile of type PF_PROFILE_TYPE_TO_DOWNLOAD, for which the SM-DP-UT can deliver a Profile Package
PF_PROFILE_TYPE_TO_DOWNLOAD	A profile type that is known by the SM-DP-UT; the SM-DP can provide one and only one profile package for this profile type, and the ICCID of the corresponding profile would be PF_ICCID_TO_DOWNLOAD.
PF_SM_DP_UT_ES2_URI	The URL of the WebService endpoint on which the SM-DP accepts ES2 requests. See Note 2
PF_ SM_SR_UT_ES3_URI	The URL of the WebService endpoint on which the SM-SR accepts ES3 requests. See Note 2
PF_ SM_SR_UT_ES4_URI	The URL of the WebService endpoint on which the SM-SR accepts ES4 requests. See Note 2
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.

Platform setting name	Description
	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:
	`93 01 02`
	`42 {L} #CI_OID`
	`5F 20 01 02`
	`95 01 82`
	`5F 24 04 21 45 01 01`
	`73 {L}`
	`C8 01 01`
	`C9 14 #CI_SUBJECT_KEY_ID`
	`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 {L} #CI_OID`
	`5F 20 01 01`
	`95 01 82`
	`5F 24 04 21 45 01 01`
	`73 {L}`
	`C8 01 02`
	`C9 14 #CI_SUBJECT_KEY_ID`
	`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 {L} #EUM_OID`
	`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 {L}`
	`C0 01 01`
	`C1 01 01`
	`C2 01 01`
	`C9 #EUM_SUBJECT_KEY_ID`
	`7F 49 {L} #PK_ECASD_S_ECKA`
	`5F 37 {L} {SIGNATURE}`
	This signature SHALL be generated using the `#EUM_S_SK_ECDSA`.

Platform setting name	Description
	see Note 1
CI_SUBJECT_KEY_ID	Subject Key Identifier of the CI GSMA CI Certificate (20 bytes long). see Note 3
CI_OID	OID of the root CI see Note 3
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.

Note that section 3.4 describes exceptions to the structure of some RPS elements described below.

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> <OperationExecutionStatus> #SUCCESS </OperationExecutionStatus> <Isd-p-aid>#DEFAULT_ISD_P_AID</Isd-p-aid> #ICCID_RPS </Record>
BIG_MEM_RPS	<RequiredMemory>#BIG_MEM</RequiredMemory>
CATTP_CAP_RPS	<CattpSupport>TRUE</CattpSupport> <CattpVersion>6.13.0</CattpVersion> <HttpSupport>FALSE</HttpSupport> <SecurePacketVersion>12.1.0</SecurePacketVersion> <RemoteProvisioningVersion>3.2.0</RemoteProvisioningVersion>
CON_PARAM_RPS	<connectivityParameters> 222F80E288002A3A0727A1253507#BEARER_DESCRIPTION4709#NAN_VALUE0D05#LOGIN0D08#PWD </connectivityParameters> see Note 6
CUR_SR_ID_RPS	<CurrentSmSrid>#SM_SR_ID</CurrentSmSrid>
CUR_SR_S_ID_RPS	<CurrentSmSrid>#SM_SR_S_ID</CurrentSmSrid>
DATA_RPS	<Data>#DATA</Data> see Note 6
DEFAULT_ISDP_RPS	<Isd-p-aid>#DEFAULT_ISD_P_AID</Isd-p-aid>
ECASD_BADSIGN_RPS	<Aid>#ECASD_AID</Aid>

RPS element name	Value
	<Tar>#ECASD_TAR</Tar> <Sin>#VIRTUAL_SIN</Sin> <Sdin>#VIRTUAL_SDIN</Sdin> <Role>ECASD</Role> <Keyset> <Version>116</Version> <Type>CA</Type> <Certificate> <Index>4</Index> <CAId>#EUM_OID</CAId> <Value>#ECASD_BAD_SIGN_CERT</Value> </Certificate> </Keyset>
ECASD_RPS	<Aid>#ECASD_AID</Aid> <Tar>#ECASD_TAR</Tar> <Sin>#VIRTUAL_SIN</Sin> <Sdin>#VIRTUAL_SDIN</Sdin> <Role>ECASD</Role> <Keyset> <Version>116</Version> <Type>CA</Type> <Certificate> <Index>4</Index> <CAId>#EUM_OID</CAId> <Value>#VIRTUAL_ECASD_CERT</Value> </Certificate> </Keyset>
ECC_KEY_LENGTH_RPS	<ECCKeyLength>ECC-256</ECCKeyLength>
EID_RPS	<Eid>#EID</Eid>
EIS_BADCASDSIGN_RPS (ES3 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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_BADSIGN_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles>

RPS element name	Value
	800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS -- *Optional* <Isdr-r>#ISD_R_ES3_RPS</Isdr-r> </Eis> see Note 1
EIS_BADEUMSIGN_RPS (ES1 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> </Eis> see Note 2
EIS_ES1_RPS (ES1 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</PlatformVersion> <Isd-p-loadfile-aid> #ISD_P_PKG_AID </Isd-p-loadfile-aid> <Isd-p-module-aid>

RPS element name	Value
	#ISD_P_MOD_AID </Isd-p-module-aid> <Ecasd>#ECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> <AdditionalProperties> <Property key=”a key” value=”a value”/> <Property key=”gsma.ESIM.DNSResolverClientSupport” value=”true”/> </AdditionalProperties> </Eis> see Note 1
EIS_ES2_RPS (ES2 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000

RPS element name	Value
	</AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS -- *Optional* <AdditionalProperties> <Property key="a key" value="a value"/> <Property key="gsma.ESIM.DNSResolverClientSupport" value="true"/> </AdditionalProperties> </Eis> see Note 1
EIS_ES3_RPS (ES3 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS – *Optional, see Note 9* #PROFILE2_RPS – *Optional, see Note 9* <Isdr-r>#ISD_R_ES3_RPS</Isdr-r> <AdditionalProperties> <Property key="a key" value="a value"/> <Property key="gsma.ESIM.DNSResolverClientSupport" value="true"/> </AdditionalProperties> </Eis> see Note 1
EIS_ES4_RPS (ES4 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType>

RPS element name	Value
	<PlatformVersion>1.0.0</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> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS <AdditionalProperties> <Property key=”a key” value=”a value”/> <Property key=”gsma.ESIM.DNSResolverClientSupport” value=”true”/> </AdditionalProperties> </Eis> see Note 1
EIS_ES7_RPS (ES7 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig">

RPS element name	Value
	#SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_ES7_RPS</Isdr-r> <AdditionalProperties> <Property key=”a key” value=”a value”/> <Property key=”gsma.ESIM.DNSResolverClientSupport” value=”true”/> </AdditionalProperties> </Eis> see Note 1
EIS_EXPIREDCASD_RPS (ES7 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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>#EXPIREDECASD_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_ES7_RPS</Isdr-r> </Eis>

RPS element name	Value
	see Note 1
EIS2_BADCASDSIGN_RPS (ES7 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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_BADSIGN_RPS</Ecasd> <EuiccCapabilities> #FULL_CAP_RPS </EuiccCapabilities> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EnumSignature> <RemainingMemory>800000</RemainingMemory> <AvailableMemoryForProfiles> 750000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS}#PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_ES7_RPS</Isdr-r></Eis> see Note 1
EIS2_ES1_RPS (ES1 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue>

RPS element name	Value
	#KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE3_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> </Eis> see Note 3
EIS3_ES1_RPS (ES1 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID2_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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> </EumSignedInfo> <EumSignature xmlns:ds="http://www.w3.org/2000/09/xmldsig"> #SIGNED_INFO_RPS <ds:SignatureValue> {SIGNATURE} </ds:SignatureValue> #KEY_INFO_RPS </EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 800000 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> </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>

RPS element name	Value
	<Sin>#VIRTUAL_SIN</Sin> <Sdin>#VIRTUAL_SDIN</Sdin> <Role>ECASD</Role> <Keyset> <Version>116</Version> <Type>CA</Type> <Certificate> <Index>4</Index> <CAId>#EUM_OID</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> <RemoteProvisioningVersion>3.2.0</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>3.2.0</RemoteProvisioningVersion>
ICCID_RPS	<Iccid>#ICCID</Iccid>
ICCID_UNKNOWN_RPS	<Iccid>#ICCID_UNKNOWN</Iccid>
ICCID1_RPS	<Iccid>#ICCID1</Iccid>
ICCID2_RPS	<Iccid>#ICCID2</Iccid>
INIT_SEQ_COUNTER_RPS	<InitialSequenceCounter>1</InitialSequenceCounter>
INVALID_EIS_RPS (ES1 interface)	<Eis> <EumSignedInfo> #VIRTUAL_EID_RPS <Eum-Id>#EUM_S_ID</Eum-Id> <ProductionDate>2014-01-01T09:30:47Z</ProductionDate> <PlatformType>eUICC Operating System</PlatformType> <PlatformVersion>1.0.0</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> </EumSignedInfo> <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
	</EumSignature> <RemainingMemory>750000</RemainingMemory> <AvailableMemoryForProfiles> 500 </AvailableMemoryForProfiles> {SM_SR_ID_RPS} #PROFILE1_RPS #PROFILE2_RPS <Isdr-r>#ISD_R_RPS</Isdr-r> </Eis> see Note 1
ISD_R_ES3_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>1</version> </Keyset>
ISD_R_ES7_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>1</version> <Type>SCP80</Type> <Cntr>1</Cntr> <Key kcv=””> <Index>1</Index> <KeyComponent type=”88” value=””> </KeyComponent> </Key> <Key kcv=””> <Index>2</Index> <KeyComponent type=”88” value=””> </KeyComponent> </Key> <Key kcv=””> <Index>3</Index> <KeyComponent type=”88” value=””> </KeyComponent> </Key> </Keyset>
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>1</version> <Type>SCP80</Type> <Cntr>1</Cntr> <Key kcv=”{KEY_KCV}”>

RPS element name	Value
	<Index>1</Index> <KeyComponent type=”88” value=”#KEY_SECURED”> </KeyComponent> </Key> <Key kcv=”{KEY_KCV}”> <Index>2</Index> <KeyComponent type=”88” value=”#KEY_SECURED”> </KeyComponent> </Key> <Key kcv=”{KEY_KCV}”> <Index>3</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>
NO_REQUIRED_MEM_RPS	<RequiredMemory>0</RequiredMemory>
PF_ICCID_TO_DOWNLOAD_RPS	<Iccid>#PF_ICCID_TO_DOWNLOAD</Iccid>
PF_PROFILE_TYPE_TO_DOWNLOAD_RPS	<ProfileType>#PF_PROFILE_TYPE_TO_DOWNLOAD</ProfileType>
PLMA_MNO1_FOR_MNO2_RPS	<Plma> <Mno-id>#MNO1_S_ID</Mno-id> <ProfileType>#PROFILE_TYPE1</ProfileType> <M2m-sp-id>#MNO2-S-ID</M2m-sp-id> <authorisedOperation>UnsetFallBackAttribute</authorisedOperation>

RPS element name	Value
	</Plma>
PLMA_MNO1_FOR_M2MSP1_RPS	<Plma> <Mno-id>#MNO1_S_ID</Mno-id> <ProfileType>#PROFILE_TYPE1</ProfileType> <M2m-sp-id>#M2MSP1_S_ID</M2m-sp-id> <authorisedOperation>GetEIS</authorisedOperation> <authorisedOperation>EnableProfile</authorisedOperation> <authorisedOperation>HandleProfileEnabledNotification</authorisedOperation> <authorisedOperation>UnsetFallBackAttribute</authorisedOperation> <authorisedOperation>HandleProfileFallBackAttributeUnsetNotification</authorisedOperation> </Plma>
PLMA_MNO1_FOR_M2MSP2_RPS	<Plma> <Mno-id># MNO1_ID_RPS</Mno-id> <ProfileType>#PROF_TYPE1_RPS</ProfileType> <M2m-sp-id>#M2MSP2_ID_RPS</M2m-sp-id> <authorisedOperation>HandleProfileEnabledNotification</authorisedOperation> </Plma>
PLMA_MNO2_FOR_MNO1_RPS	<Plma> <Mno-id>#MNO2_ID_RPS</Mno-id> <ProfileType>#PROF_TYPE2_RPS</ProfileType> <M2m-sp-id>#MNO1_ID_RPS</M2m-sp-id> <authorisedOperation>GetEIS</authorisedOperation> <authorisedOperation>DisableProfile</authorisedOperation> <authorisedOperation>HandleProfileDisabledNotification</authorisedOperation> </Plma>
PLMA_MNO2_FOR_M2MSP1_RPS	<Plma> <Mno-id>#MNO2_S_ID</Mno-id> <ProfileType>#PROFILE_TYPE2</ProfileType> <M2m-sp-id>#M2MSP1_S_ID</M2m-sp-id> <authorisedOperation>EnableProfile</authorisedOperation> <authorisedOperation>HandleEmergencyProfileAttributeSetNotification</authorisedOperation> <authorisedOperation>HandleProfileEnabledNotification</authorisedOperation> <authorisedOperation>SetEmergencyProfileAttribute</authorisedOperation> <authorisedOperation>SetFallBackAttribute</authorisedOperation> </Plma>
ONC_MNO1_RPS	<Onc> <Mno-id># MNO1_ID_RPS</Mno-id> <ProfileType>#PROF_TYPE1_RPS</ProfileType> <discardedNotifications>HandleProfileEnabledNotification</discardedNotifications> </Onc>
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> <Qualification>Not allowed</Qualification>

RPS element name	Value
	</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> {SM_DP_ID_RPS} #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> {SM_DP_ID_RPS} #PROF_TYPE2_RPS <AllocatedMemory>100000</AllocatedMemory> <FreeMemory>50000</FreeMemory> #POL2_DEL_RPS </ProfileInfo>
PROFILE3_RPS	<ProfileInfo> #ICCID2_RPS #ISDP3_RPS #MNO2_ID_RPS <FallbackAttribute>TRUE</FallbackAttribute> #SUB_ADDR2_RPS <State>Enabled</State> {SM_DP_ID_RPS} #PROF_TYPE2_RPS <AllocatedMemory>100000</AllocatedMemory> <FreeMemory>50000</FreeMemory> #POL2_DEL_RPS </ProfileInfo>
SC3_DR_HOST_RPS	<ScenarioParameter>#SC3_DR_HOST</ScenarioParameter>
SC3_DR_RPS	<ScenarioParameter>#SC3_DR</ScenarioParameter>
SC3_NO_DR_RPS	<ScenarioParameter>#SC3_NO_DR</ScenarioParameter>
SD_ISDP2_RPS	<sd-aid>#ISD_P_AID2</sd-aid>
SHORT_VP_RPS	< ValidityPeriod>#SHORT_VALIDITY_PERIOD < /ValidityPeriod>
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”/>

RPS element name	Value
	<ds:SignatureMethod Algorithm=”http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256”/> <ds:Reference> <ds:Transforms> <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /> </ds:Transforms> <ds:DigestMethod Algorithm=”http://www.w3.org/2001/04/xmlenc#sha256”/> <ds:DigestValue>{DIGEST}</ds:DigestValue> </ds:Reference> </ds:SignedInfo>
SM_DP_S_ID_RPS	<Smdp-id>#SM_DP_S_ID</Smdp-id>
SM_DP_UT_ID_RPS	<Smdp-id>#SM_DP_ID</Smdp-id>
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>#SMALL_MEM</RequiredMemory>
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_S_UNK_ID_RPS	<Target-SmSr-id>#UNKNOWN_SM_SR_ID</Target-SmSr-id>
TGT_SR_UT_ID_RPS	<Target-SmSr-id>#SM_SR_ID</Target-SmSr-id>
TGT_UK_SR_S_ID_RPS	<Target-SmSr-id>#UNKNOWN_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 generated with the #EUM_S_SK_ECDSA *Note 2: The {SIGNATURE} SHALL NOT be generated with the #EUM_S_SK_ECDSA* *Note 3: The {SIGNATURE} SHALL be 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: Void*	

RPS element name	Value
Note 8: Void	
Note 9: each test sequence using this structure EIS_ES3_RPS specifies which of the profiles appear in the structure. In case the test sequence does not specify it, the presence or absence of either of the profiles is irrelevant to the purpose of the test sequence and does not justify failing the test sequence.	

Table 12: RPS Elements

B.6 Profiles Information

Here is the different Profiles information used to execute the test cases defined in the section 5.3 or 4.4 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)
	  ○ ICCID (i.e. `#ICCID` if the Profile is Enabled)
	  ○ ISD-P AID (i.e. `#DEFAULT_ISD_P_AID` if the Profile is Enabled)
	  ○ MSISDN (i.e. `#MSISDN` if the Profile is Enabled)
	  ○ Fall-back Attribute
	  ○ state
	  ○ Profile type
	  ○ allocated memory
	  ○ POL2
	• ISD-R information with
	  ○ AID (i.e. `#ISD_R_AID`)
	  ○ SIN
	  ○ SDIN
	  ○ SCP80 and/or SCP81 keysets information
	• ECASD information with
	  ○ AID (i.e. `#ECASD_AID`)
	  ○ SIN
	  ○ SDIN
	  ○ certificate (i.e. `#ECASD_CERTIFICATE`)

Profile information	Description

	• eUICC capabilities
	  ○ supported CAT_TP version and/or supported HTTPS version
	  ▪ depends if O_HTTPS and O_CAT_TP are supported
	  ○ supported secured packet version
	  ○ supported remote provisioning version
	• The EUM X.509 certificate containg the #EUM_PK_ECDSA
	
	The tool provider SHALL format the data (i.e. RPS) and add:
	• the SM-SR-UT Identifier (i.e. #SM_SR_ID)
	• the SM-DP-UT Identifier (i.e. #SM_DP_ID) if required
	• 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

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.
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.

Profile information	Description
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 : {
  major-version 2,
  minor-version 3,
  profileType "GSMA Profile Package",
  iccid '89019990001234567893'H,
  eUICC-Mandatory-services {
  usim NULL,
  milenage NULL
  },
  eUICC-Mandatory-GFSTEList {
  -- see Note 1
  id-MF,
  id-USIM
  },
  -- These SMS Connectivity Parameters MAY be freely changed
  connectivityParameters 'A0090607#TON_NPI#DIALING_NUMBER'H
}
	A0 4D
	  80 01 02
	  81 01 03
	  82 14 47534D412050726F66696C65205061636B616765
	  83 0A 89019990001234567893
	  A5 04
	  81 00
	  84 00
	
	  A6 10
	
	  06 06 67810F010201
	  06 06 67810F010204
	
	  87 0B A0090607913386994211F0

PE_MF	

mfValue ProfileElement ::= mf : {
  mf-header {
  mandated NULL,
  identification 1
  },
  templateID id-MF,
  mf {
  fileDescriptor : {
  pinStatusTemplateDO '01020A'H
  }
	B0 8201F8
	  A0 05
	  80 00
	  81 01 01
	
	  81 06 67810F010201
	  A2 07
	  A1 05
	  C6 03 01020A

},
ef-pl {
  fileDescriptor : {
  -- EF PL modified to use Access Rule 15 within EF ARR
  securityAttributesReferenced '0F'H
  }
},
ef-iccid {
  -- swapped ICCID: 98109909002143658739
  fillFileContent '98109909002143658739'H
},
ef-dir {
  fileDescriptor {
  -- Shareable Linear Fixed File
  -- 4 records, record length: 38 bytes
  fileDescriptor '42210026'H,
  efFileSize '98'H
  },
  -- USIM AID: A0000000871002FF33FF018900000100
  fillFileContent
  '61184F10A0000000871002FF33FF0189000001005004555349 4D'H
},
ef-arr {
  fileDescriptor {
  -- 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,
  efFileSize '022B'H
  },
  -- see Table 15 to see the access rules definitions
  fillFileContent '#ACCESS_RULE1'H,
  fillFileOffset 10,
  fillFileContent '#ACCESS_RULE2'H,
  fillFileOffset 15,
  fillFileContent '#ACCESS_RULE3'H,
  fillFileOffset 26,
	A3 05
	  A1 03
	  8B 01 0F
	
	A4 0C
	  83 0A 98109909002143658739
	
	A5 27
	  A1 09
	  82 04 42210026
	  80 01 98
	
	  83 1A
	  61184F10A0000000871002FF33FF0189000001005004555349 4D
	
	A6 82019E
	  A1 0A
	  82 04 42210025
	  80 02 022B
	
	  83 1B #ACCESS_RULE1
	  82 01 0A
	  83 16 #ACCESS_RULE2
	  82 01 0F
	  83 0B #ACCESS_RULE3
	  82 01 1A

```
    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
                      0108A4068301029501088001 58A40683
                      010A950108'H

  }
}
```

```
    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
          0108A4068301029501088001 58A40683
          010A950108
```

PE_PUK

```
pukVal ProfileElement ::= pukCodes : {
  puk-Header {
    mandated NULL,
    identification 2
  },
  pukCodes {
   {
```

```
A3 3F
  A0 05
    80 00
    81 01 02

  A1 36
   30 11
```

```
      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
    }
  }
}
```

```
    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 3
  },
  pinCodes pinconfig : {
    {
      keyReference pinAppl1,
      pinValue '31323334FFFFFFFF'H,
      unblockingPINReference pukAppl1
    },
    {
      keyReference pinAppl2,
      pinValue '30303030FFFFFFFF'H,
      unblockingPINReference pukAppl2
    },
    {
      keyReference adm1,
```

```
A2 41
  A0 05
    80 00
    81 01 03

  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
      80 01 0A
```

pinValue '35363738FFFFFFFF'H, pinAttributes 1 } } }	81 08 35363738FFFFFFFF 83 01 01
PE_USIM	

usimValue ProfileElement ::= usim : {
usim-header {
mandated NULL,
identification 4
},
templateID id-USIM,
adf-usim {
fileDescriptor : {
fileID '7FF1'H,
dfName 'A0000000871002FF33FF018900000100'H,
pinStatusTemplateDO '01810A'H
}
},
ef-imsi {
-- numerical format: 234101943787656
fillFileContent '08294301913487 6765'H
},
ef-arr {
fileDescriptor {
linkPath '2F06'H
}
},
ef-ust {
-- Service Dialling Numbers, Short Message Storage…
fillFileContent '0A2E178CE73204000000000000'H
},
ef-spn {
-- ASCII format: "GSMA eUICC"
fillFileContent '0247534D41206555494343FFFFFFFFFFFF'H
},
	B3 7C
	A0 05
	80 00
	81 01 04
	
	81 06 67810F010204
	A2 1D
	A1 1B
	83 02 7FF1
	84 10 A0000000871002FF33FF018900000100
	C6 03 01810A
	
	A3 0B
	
	83 09 08294301913487 6765
	
	A4 06
	A1 04
	C7 02 2F06
	
	A8 0F
	
	83 0D 0A2E178CE73204000000000000
	
	AD 13
	
	83 11 0247534D41206555494343FFFFFFFFFFFF

ef-est {
-- *Services deactivated*
fillFileContent '00'H
},
ef-acc {
-- *Access class 4*
fillFileContent '0040'H
},
ef-ecc {
-- *Emergency Call Code 911*
fillFileContent '19F1FF01'H
}
}
	AE 03
	
	83 01 00
	
	B2 04
	
	83 02 0040
	
	B6 06
	
	83 04 19F1FF01

PE_USIM_PIN	

usimPin ProfileElement ::= pinCodes : {
pin-Header {
mandated NULL,
identification 5
},
pinCodes pinconfig : {
{
keyReference secondPINAppl1,
pinValue '39323338FFFFFFFF'H
unblockingPINReference secondPUKAppl1,
-- *PIN is Enabled*
pinAttributes 1,
-- *maxNumOfAttemps:2, retryNumLeft:2*
maxNumOfAttemps-retryNumLeft 34
}
}
}
	A2 25
	A0 05
	80 00
	81 01 05
	
	A1 1C A0 1A
	30 18
	80 02 0081
	81 08 39323338FFFFFFFF
	82 02 0081
	
	83 01 01
	
	84 01 22

PE_NAA	
akaParamValue ProfileElement ::= akaParameter : { aka-header {	A4 3A A0 05

mandated NULL,
identification 6
},
algoConfiguration algoParameter : {
algorithmID milenage,
-- RES and MAC 64 bits, CK and IK 128 bits
algorithmOptions '01'H,
key '000102030405060708090A0B0C0D0E0F'H,
opc '0102030405060708090A0B0C0D0E0F00'H,
-- rotationConstants uses default: '4000204060'H
-- xoringConstants uses default value
authCounterMax '010203'H
}
-- sqnOptions uses default: '02'H
-- sqnDelta uses default: '000010000000'H
-- sqnAgeLimit uses default: '000010000000'H
-- sqnInit uses default: all bytes zero
}
	80 00
	81 01 06
	
	A1 31 A1 2F
	80 01 01
	
	81 01 01
	82 10 000102030405060708090A0B0C0D0E0F
	83 10 0102030405060708090A0B0C0D0E0F00
	
	86 03 010203

PE_MNO_SD	

mnoSdValue ProfileElement ::= securityDomain : {
sd-Header {
mandated NULL,
identification 7
},
instance {
applicationLoadPackageAID 'A0000001515350'H,
classAID 'A000000151535041'H,
instanceAID 'A000000151000000'H,
applicationPrivileges '82FC80'H,
-- Secured
lifeCycleState '0F'H,
-- SCP80 supported
applicationSpecificParametersC9 '810280008201F08701F0'H,
-- other parameters MAY be necessary
applicationParameters {
-- TAR: B20100, MSL: 12
	A6 82010A
	A0 05
	80 00
	81 01 07
	
	A1 44
	4F 07 A0000001515350
	4F 08 A000000151535041
	4F 08 A000000151000000
	82 03 82FC80
	
	83 01 0F
	
	C9 0A 810280008201F08701F0
	
	EA 11

`      uiccToolkitApplicationSpecificParametersField`	`    80 0F`
`         '01000001000000002011203B2010000'H`	`      01000001000000002011203B2010000`
`    }`	
`  },`	
`  keyList {`	`A2 81BA`
`    {`	`  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,
      keyVersionNumber '01'H,
      keyCompontents {
        {
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value MAY be freely changed
          keyData '112233445566778899AABBCCDDEEFF10'H
        }
      }
    },
       -- AES Token Key (as an example)
       -- This value MAY be freely changed
      keyUsageQualifier  '81'H,
      -- MAY be used by SD
      keyAccess  '01'H,
      -- Key Id 01
      keyIdentifier  '01'H,
      keyVersionNumber '70'H,
      keyCompontents  {
       {
         -- AES (16 bytes key length)
         -- This value MAY be freely changed
          keyType  '88'H,
          -- This value MAY be freely changed
          keyData  'CDFE56B7B72FAE6A047341F003D7A48D'H
        }
      }
    },
    {
       -- Receipt (the AES scheme SHALL be supported)
      keyUsageQualifier '44'H,
      -- MAY be used by SD
      keyAccess '01'H,
```

```
      82 01 03
      83 01 01
      30 17
        30 15
          80 01 80
          86 10 112233445566778899AABBCCDDEEFF10
    30 25
      95 01 81
      96 01 01
      82 01 01
      83 01 70
      30 17
        30 15
          80 01 88
          86 10 CDFE56B7B72FAE6A047341F003D7A48D
    30 25
      95 01 44
      96 01 01
```

```
       -- Key Id 01
       keyIdentifier '01'H,
       keyVersionNumber '71'H,
       keyCompontents  {
        {
         -- AES (16 bytes key length)
         keyType  '88'H,
         -- This value MAY be freely changed
         keyData  '11121314212223243132333441424344'H
        }
       }
    }
  }
}
```

```
      82 01 01
      83 01 71
      30 17
        30 15


          80 01 88

          86 10 11121314212223243132333441424344
```

PE_SSD

```
ssdValue ProfileElement ::= securityDomain : {
  sd-Header {
    mandated NULL,
    identification 8
  },
  instance {
    applicationLoadPackageAID 'A0000001515350'H,
    classAID 'A000000151535041'H,
    instanceAID 'A00000055910100102736456616C7565'H,
    -- by default extradited under MNO-SD
    -- Privileges: Security Domain + Trusted Path
    applicationPrivileges '808000'H,
    -- Personalized
    lifeCycleState '0F'H,
    -- SCP80 supported, extradiction supported
    applicationSpecificParametersC9 '810280008201F0'H,
    applicationParameters {
      -- TAR: 6C7565, MSL: 12
      uiccToolkitApplicationSpecificParametersField
```

```
A6 81C0
  A0 05
    80 00
    81 01 08

  A1 49
    4F 07 A0000001515350
    4F 08 A000000151535041
    4F 10 A00000055910100102736456616C7565


    82 03 808000

    83 01 0F

    C9 07 810280008201F0
    EA 11

      80 0F
        0100000100000020112036C756500
```

'0100000100000020112036C756500'H	
}	
},	A2 6C
keyList {	30 22
{	
-- *C-ENC + R-ENC*	95 01 38
keyUsageQualifier '38'H,	82 01 01
keyIdentifier '01'H,	83 01 01
keyVersionNumber '01'H,	30 17
keyCompontents {	30 15
{	
-- *DES mode implicitly known (as an example)*	80 01 80
keyType '80'H,	
-- *This value MAY be freely changed*	86 10 11223344556677881122334455667788
keyData '11223344556677881122334455667788'H	
}	
}	
},	30 22
{	
-- *C-MAC + R-MAC*	95 01 34
keyUsageQualifier '34'H,	
-- *MAC key*	82 01 02
keyIdentifier '02'H,	83 01 01
keyVersionNumber '01'H,	30 17
keyCompontents {	30 15
{	
-- *DES mode implicitly known (as an example)*	80 01 80
keyType '80'H,	
-- *This value MAY be freely changed*	86 10 11223344556677881122334455667788
keyData '11223344556677881122334455667788'H	
}	
}	
},	30 22
{	
-- *C-DEK + R-DEK*	95 01 C8
keyUsageQualifier 'C8'H,	
-- *data ENC key*	82 01 03

keyIdentifier '03'H,
keyVersionNumber '01'H,
keyCompontents {
{
-- *DES mode implicitly known (as an example)*
keyType '80'H,
-- *This value MAY be freely changed*
keyData '112233445566778811223344556677 88'H
}
}
}
}
}
	83 01 01
	30 17
	30 15
	
	80 01 80
	
	86 10 112233445566778811223344556677 88

PE_RFM_UICC	

rfmUicc ProfileElement ::= rfm : {
rfm-header {
identification 11
},
-- *Instance AID*
instanceAID ' A00000055910100001'H,
tarList {
'B00000'H
},
-- *cryptographic checksum + counter higher*
minimumSecurityLevel '12'H,
-- *full access*
uiccAccessDomain '00'H,
-- *full access*
uiccAdminAccessDomain '00'H
}
	A7 20
	A0 03
	81 01 0B
	
	4F 09 A00000055910100001
	A0 05 04 03
	B00000
	
	81 01 12
	
	04 01 00
	
	04 01 00

PE_RFM_USIM	

rfmUsim ProfileElement ::= rfm : {
rfm-header {
identification 12
	A7 40
	A0 03
	81 01 0C

},
-- *Instance AID*
instanceAID 'A00000055910100002'H,
tarList {
'B00020'H
},
-- *cryptographic checksum + counter higher*
minimumSecurityLevel '12'H,
-- *full access*
uiccAccessDomain '00'H,
-- *full access*
uiccAdminAccessDomain '00'H,
adfRFMAccess {
adfAID 'A0000000871002FF33FF018900000100'H,
-- *UICC access condition: ADM1*
adfAccessDomain '02000100'H,
-- *UICC access condition: ADM1*
adfAdminAccessDomain '02000100'H
}
}
	
	4F 09 A00000055910100002
	A0 05
	04 03 B00020
	
	81 01 12
	
	04 01 00
	
	04 01 00
	30 1E
	80 10 A0000000871002FF33FF018900000100
	
	81 04 02000100
	
	82 04 02000100

PE_END	

endValue ProfileElement ::= end : {
end-header {
mandated NULL,
identification 99
}
}
	AA 07
	A0 05
	80 00
	81 01 63

Note: The rule related to the usage of curly brackets defined in section 2.2.3 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’
- `#PE_SSD1`: This PE installs SSD1.
- `#PE_SSD3`: This PE installs SSD3.

ASN.1 format	DER TLV format
PE_APPLET1	

applet1 ProfileElement ::= application : {
  app-Header {
  mandated NULL,
  identification 9
  },
  loadBlock {
  loadPackageAID 'A000000559101001'H,
  loadBlockObject '{LFDB_APPLET1}'H
  },
  instanceList {
  {
  applicationLoadPackageAID 'A000000559101001'H,
  classAID 'A000000559101001112233'H,
  instanceAID 'A00000055910100111223301'H,
  applicationPrivileges '000000'H,
  -- *Selectable by default*
  applicationSpecificParametersC9 '00'H,
  applicationParameters {
  uiccToolkitApplicationSpecificParametersField
  -- *TAR: 112233*
  '010000000000311223300'H
  }
  }
  }
}
see Note 1
	A8 {L}
	  A0 05
	  80 00
	  81 01 09
	
	  A1 {L}
	  4F 08 A000000559101001
	  C4 {L} {LFDB_APPLET1}
	
	  A2 3E
	  30 3C
	  4F 08 A000000559101001
	  4F 0B A000000559101001112233
	  4F 0C A00000055910100111223301
	  82 03 000000
	  C9 01 00
	  EA 0D
	  80 0B
	  010000000000311223300

PE_APPLET3	

applet3 ProfileElement ::= application : {
  app-Header {
  mandated NULL,
  identification 10
  },
  loadBlock {
  loadPackageAID 'A000000559101003'H,
  loadBlockObject '{LFDB_APPLET3}'H
  },
  instanceList {
  {
  applicationLoadPackageAID 'A000000559101003'H,
  classAID 'A0000005591010034455 66'H,
  instanceAID 'A00000055910100344556601'H,
  applicationPrivileges '000000'H,
  applicationSpecificParametersC9 '00'H
  }
  }
}
see Note 1
	A8 {L}
	  A0 05
	  80 00
	  81 01 0A
	
	  A1 {L}
	  4F 08 A000000559101003
	  C4 {L} {LFDB_APPLET3}
	
	  A2 2F
	  30 2D
	  4F 08 A000000559101003
	  4F 0B A0000005591010034455 66
	  4F 0C A00000055910100344556601
	  82 03 000000
	  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 '10'H,
  shortEFID ''H
  },
  fillFileContent '1122334455'H
  }
  }
}
see Note 2
	A1 26
	  A0 05
	  80 00
	  81 01 16
	
	  A1 1D
	  30 1B
	  62 12
	
	  82 02 0121
	  83 02 1122
	
	  8B 03 2F0601
	  80 01 10
	  88 00
	
	  81 05 1122334455

```
ssdValue ProfileElement ::= securityDomain : {
  sd-Header {
    mandated NULL,
    identification 9
  },
  instance {
    applicationLoadPackageAID 'A0000001515350'H,
    classAID 'A000000151535041'H,
    instanceAID 'A00000055910100173736431112233 01'H,
    -- by default extradited under MNO-SD
    -- Privileges: Security Domain + Trusted Path
    applicationPrivileges '808000'H,
    -- Personalized
    lifeCycleState '0F'H,
    -- SCP80 supported, extradiction supported
    applicationSpecificParametersC9 '810280008201F0'H,
    applicationParameters {
      -- TAR: 112233, MSL: 12
      uiccToolkitApplicationSpecificParametersField
         '01000001000000020112031122330 0'H
    }
  },
  keyList {
    {
      -- C-ENC + R-ENC
      keyUsageQualifier '38'H,
      keyIdentifier '01'H,
      keyVersionNumber '01'H,
      keyCompontents {
        {
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value MAY be freely changed
          keyData '11223344556677881122334455667788'H
        }
      }
    },
```

```
A6 81C0
  A0 05
    80 00
    81 01 09

  A1 49
    4F 07 A0000001515350
    4F 08 A000000151535041
    4F 10 A00000055910100173736431112233 01

    82 03 808000

    83 01 0F

    C9 07 810280008201F0
    EA 11

      80 0F
        01000001000000020112031122330 0


  A2 6C
    30 22

      95 01 38
      82 01 01
      83 01 01
      30 17
        30 15

          80 01 80

          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
        }
      }
    }
  }
}
```

```
ssdValue ProfileElement ::= securityDomain : {
  sd-Header {
    mandated NULL,
    identification 10
  },
  instance {
    applicationLoadPackageAID 'A0000001515350'H,
    classAID 'A000000151535041'H,
    instanceAID 'A00000055910100173736433445566 01'H,
    -- by default extradited under MNO-SD
    -- Privileges: Security Domain + Trusted Path
    applicationPrivileges '808000'H,
    -- Personalized
    lifeCycleState '0F'H,
    -- SCP80 supported, extradiction supported
    applicationSpecificParametersC9 '810280008201F0'H,
    applicationParameters {
      -- TAR: 445566, MSL: 12
      uiccToolkitApplicationSpecificParametersField
         '01000001000000020 1120344556600'H
    }
  },
  keyList {
    {
      -- C-ENC + R-ENC
      keyUsageQualifier '38'H,
      keyIdentifier '01'H,
      keyVersionNumber '01'H,
      keyCompontents {
        {
          -- DES mode implicitly known (as an example)
          keyType '80'H,
          -- This value MAY be freely changed
          keyData '11223344556677881122334455667788'H
        }
      }
    },
```

```
A6 81C0
  A0 05
    80 00
    81 01 0A

  A1 49
    4F 07 A0000001515350
    4F 08 A000000151535041
    4F 10 A00000055910100173736433445566 01

    82 03 808000

    83 01 0F

    C9 07 810280008201F0
    EA 11

      80 0F
        01000001000000020 1120344556600

  A2 6C
    30 22

      95 01 38
      82 01 01
      83 01 01
      30 17
        30 15

          80 01 80

          86 10 11223344556677881122334455667788
```

{
-- *C-MAC + R-MAC*
keyUsageQualifier '34'H,
-- *MAC key*
keyIdentifier '02'H,
keyVersionNumber '01'H,
keyCompontents {
{
-- *DES mode implicitly known (as an example)*
keyType '80'H,
-- *This value MAY be freely changed*
keyData '11223344556677881122334455667788'H
}
}
},
{
-- *C-DEK + R-DEK*
keyUsageQualifier 'C8'H,
-- *data ENC key*
keyIdentifier '03'H,
keyVersionNumber '01'H,
keyCompontents {
{
-- *DES mode implicitly known (as an example)*
keyType '80'H,
-- *This value MAY be freely changed*
keyData '11223344556677881122334455667788'H
}
}
}
}
}
See Note 1
	30 22
	95 01 34
	82 01 02
	83 01 01
	30 17
	30 15
	80 01 80
	86 10 11223344556677881122334455667788
	30 22
	95 01 C8
	82 01 03
	83 01 01
	30 17
	30 15
	80 01 80
	86 10 11223344556677881122334455667788

Note: The rule related to the usage of curly brackets defined in section 2.2.3 SHALL NOT apply for the elements described in the column "ASN.1 format".

Note 1: This PE SHALL be added just after the `#PE_SSD`.

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].
CREATED_ISD_P_AID	The instance AID of an ISD-P created by the SM-SR-UT or SM-SR-S.
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).

Variable name	Description
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].
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.
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.
NEW_SCP81_PSK KCV	Key check value of the #NEW_SCP81_PSK.
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.
PSK_DEK KCV	Key check value of the #PSK_DEK.
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).

Variable name	Description
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).
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`
SM_DP_ID_RPS	The SM-DP identifier structure used in off-card interfaces. Depending of the test, this value SHALL be either: • `#SM_DP_UT_ID_RPS` or • `#SM_DP_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:

Method name	Explanation

ENVELOPE_SMS_PP	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.

EXPANDED_COMMANDS	Wraps command APDUs within Expanded Remote Application data format as defined in ETSI TS 102 226 [6], without Command Scripting template tag nor End of content indicator.
	
	Parameters:
	• *APDU1; APDU2…*
	
	The result of applying this method to these parameters SHALL be:
	
	`'22 {L}' + `*APDU1*` +`
	`'22 {L}' + `*APDU2*` + … +`

Method name	Explanation

EXPANDED_RESPONSES	Wraps response APDUs within Expanded Remote Application data format as defined in ETSI TS 102 226 [6], without Command Scripting template tag nor End of content indicator.
	
	Parameters:
	• *R-APDU1; R-APDU2…*
	
	The result of applying this method to these parameters SHALL be:
	
	`'23 {L}' + R-APDU1 +`
	`'23 {L}' + R-APDU2 + … +`

HTTPS_CONTENT	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`
	`Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0`
	`#X_ADMIN_NEXT_URI`
	
	`{COMMAND_SCRIPT}`
	
	`{COMMAND_SCRIPT}` SHALL be:
	`'AE 80' +`
	`EXPANDED_COMMANDS(APDU1, APDU2,...) '00 00'`

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`
	`Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0`
	`#X_ADMIN_NEXT_URI`

Method name	Explanation

	`X-Admin-Targeted-Application: ISD_P_TARGETED_AID`
	
	`{COMMAND_SCRIPT}`
	
	• If the commands list is composed of APDUs:
	
	`{COMMAND_SCRIPT}` SHALL contain the list of APDUs 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' +`
	`EXPANDED_COMMANDS( COMMAND1, COMMAND2,...)`
	`'00 00'`
	
	If *CHAINING_OPT* is set, the `{COMMAND_SCRIPT}` SHALL be:
	`'AE 80' +`
	`'83 01' + CHAINING_OPT +`
	`EXPANDED_COMMANDS( COMMAND1, COMMAND2,...)`
	`'00 00'`
	
	• 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 + … +
	`'00 00'`

HTTPS_EMPTY_CONTENT	Generate an HTTPS POST message sent by the SM-SR containing no command but instructing to not close the HTTP session.
	
	`#HTTP/1.1 204`
	`#X_ADMIN_PROTOCOL`
	`#X_ADMIN_NEXT_URI`

INSTALL_FOR_PERSO	Generates the APDU INSTALL (for personalization) allowing to target a specific Security Domain identified by its instance AID

Method name	Explanation
	Parameters: • *AID* Result: - CLA = 80 - INS = E6 - P1 = 20 - P2 = 00 - LC = 16 - Data = 00 00 10 *AID* 00 00 00 - LE = 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*

Method name	Explanation
	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.
SCP03T_REPLACE_SESSION_KEYS	Parameters: • None Here is the SCP03t script to generate: '87 {L}' + '80 {L} #*INIT_MAC*' '81 {L} #*PPK-ENC*' '82 {L} #*PPK-MAC*' '83 {L} #*PPK-RMAC*' The TLV starting with Tag '87'is 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.
SCP03T_REPLACE_SESSION_KEYS_BAD_LENGTH	Parameters: • None Here is the SCP03t script to generate: '87 {L}' + '80 {L} #*INIT_MAC_32*' '81 {L} #*PPK-ENC_32*' '82 {L} #*PPK-MAC_32*' '83 {L} #*PPK-RMAC_32*' The TLV starting with Tag '87'is 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 to cipher and sign this TLV 87.
SCP03T_SCRIPT_INI_AUTH	Generate an SCP03t script Parameters: • *KVN* Here is the SCP03t script to generate:

Method name	Explanation
	'84 0A' + *KVN* + '00 {HOST_CHALLENGE}' '85 11 33 {HOST_CRYPTOGRAM} {MAC}' See Annex C for the definition of {HOST_CHALLENGE}, {HOST_CRYPTOGRAM} and {MAC}. 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_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 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:

Method name	Explanation
	• *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: '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, or the random keys if the previous script was a SCP03T_REPLACE_SESSION_KEYS, SHALL be used to cipher and sign each TLV.
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). If it is not specified differently in the test step, the KVN used SHALL be either • #SCP80_KVN if the TAR indicates that the targeted SD is the ISD-R or an ISD-P or • #MNO_SCP80_KVN if the TAR indicates (#MNO_TAR) that the targeted SD is the MNO_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).

Method name	Explanation
	{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}' + EXPANDED_COMMANDS(*COMMAND1*, *COMMAND2*,...) If *CHAINING_OPT* is set, the {COMMAND_SCRIPT} SHALL be: 'AA {L}' + '83 01' + *CHAINING_OPT* + EXPANDED_COMMANDS(*COMMAND1*, *COMMAND2*,...) • 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 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

Method name	Explanation

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>`
	`<ProcessingEnd>{CURRENT_DATE}</ProcessingEnd>`
	`<FunctionExecutionStatus>`
	`<Status>`*`STATUS`*`</Status>`
	`<StatusCodeData>`
	`<Subject>`*`SUBJECT_CODE`*`</Subject>`
	`<Reason>`*`REASON_CODE`*`</Reason>`

Method name	Explanation
	</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.
SEND_NOTIF	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>

Method name	Explanation
	</ReceiverEntity> <MessageId>{MESSAGE_ID}</MessageId> <MessageType>*NOTIF_NAME*</MessageType> <MessageDate>{CURRENT_DATE}</MessageDate> </RPSHeader> <RPSBody> <*NOTIF_NAME*> <FunctionCallIdentifier> {FUNC_CALL_ID} </FunctionCallIdentifier> *IN_DATA1* *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.
SEND_REQ	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>

Method name	Explanation
	<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}. 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_SOAP_REQ	Send a secured request message using the SOAP protocol to an off-card entity. Parameters: • *FUNCTION_NAME* • *IN_DATA1; IN_DATA2…*

Method name	Explanation

	The request is built this way:
	• The template below is used
	• The FUNCTION_NAME identifies the XML type that represents the request, as defined in the euicc.request.ESx.xsd
	• The IN_DATA that are <wsa:Xxx> fields replace the corresponding <wsa:Xxx> in the <s:Header> of the template below
	• The other IN_DATA are RPS elements that shall be placed in the XML structure following the type identified by FUNTION_NAME.
	
	`<?xml version="1.0" encoding="UTF-8"?>`
	`<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`
	`xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"`
	`xmlns:rps3="http://namespaces.gsma.org/esim-messaging/3">`
	`<s:Header>`
	`<wsa:From>…</wsa:From>`
	`<wsa:To>…</wsa:To>`
	`<wsa:MessageID>…</wsa:MessageID>`
	`<wsa:Action>…</wsa:Action>`
	`</s:Header>`
	`<s:Body rps3:MessageVersion="1.0.0">`
	`<{FUNCTION_NAME}>`
	`<rps3:FunctionCallIdentifier> callID:1 </rps3:FunctionCallIdentifier>`
	`<rps3:ValidityPeriod>3600</rps3:ValidityPeriod>`
	`{IN_DATA1}`
	`{IN_DATA2}`
	`…`
	`</{FUNCTION_NAME}>`
	`</s:Body>`
	`</s:Envelope>`

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"`

Method name	Explanation
	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…). Depending of the receiver of this message, the endpoint SHALL be either the #SM_DP_ACCESSPOINT or the #SM_SR_ACCESSPOINT.

Method name	Explanation
STORE_ISDP_KEYS	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: - 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 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 #SCP03_KVN 91 00 84 {L} #HOST_ID (present only if *SC3_PARAM*=#SC3_DR_HOST) 7F 49 {L} #SM_EPK_ECKA

Method name	Explanation
	`00 85 {L}' + RANDOM_CHALLENGE`
STORE_ISDR_KEYS	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: - CLA = 80 - INS = E2 - P1 = 89 - P2 = 01 - LC = {LC} - Data = '3A 02 {L} A6 {L} 90 02 03' + *SC3_PARAM* + '95 01 10 -- *Key Usage* 80 01 88 -- *Key Type* 81 01 10 -- *Key Length* 82 01 01 -- *Key Identifier* 83 01 #SCP80_NEW_KVN -- *Key Version Number* 91 00 -- *Initial Sequence Counter* 45 {L} #ISD_R_SDIN (present only if *SC3_PARAM*=#SC3 DR HOST) 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_NEW_KVN 91 00

Method name	Explanation
	45 {L} #ISD_R_SDIN (present only if *SC3_PARAM* = #SC3_DR_HOST) 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} 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

Method name	Explanation
	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_DNS_PARAM	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A5 {L} 81 {L} #SM-SR_FQDN A2 {L} 3E {L} #DNS_IP 82 02 #DNS_PORT 82 02 #DNS_PORT – redundant TLV
BAD_STORE_POL1	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 06 - Data = 3A 06 03 81 01 07

Name	Content in hexadecimal string
DELETE_ISDP1	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = 12 - Data = 4F 10 #ISD_P_AID1 - LE = 00
DELETE_ISDP_UNKNOWN	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = 12 - Data = 4F 10 #ISD_P_AID_UNKNOWN - 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
ENVELOPE_LOCAL_DISABLE	- CLA = 80 - INS = C2 - P1 = 00 - P2 = 00 - LC = {L} - Data = E4 01 01
ENVELOPE_LOCAL_ENABLE	- CLA = 80 - INS = C2 - P1 = 00 - P2 = 00 - LC = {L} - Data = E4 01 00
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

Name	Content in hexadecimal string
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
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

Name	Content in hexadecimal string
GET_EMERGENCY	- CLA = 80 - INS = F2 - P1 = 40 - P2 = 02 - LC = 09 - Data = 4F 00 #ISD_P_ATTRIBUTE 01 02 5C 02 4F #ISD_P_ATTRIBUTE - 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

Name	Content in hexadecimal string
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
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

Name	Content in hexadecimal string
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_SSD_AID_ECASD	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 44 - Data = 07 A0 00 00 01 51 53 50 08 A0 00 00 01 51 53 50 41 10 #ECASD_AID 03 80 80 00 1C EA 11 80 0F 01 00 00 01 00 00 00 02 01 12 03 11 22 33 00 C9 07 81 02 80 00 82 01 F0 00 -LE = 00

Name	Content in hexadecimal string
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
INSTALL_SSD_TAR_ISDR	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 44 - Data = 07 A0 00 00 01 51 53 50 08 A0 00 00 01 51 53 50 41 10 A0 00 00 05 59 10 10 01 73 73 64 31 11 22 33 01 03 80 80 00 1C EA 11 80 0F 01 00 00 01 00 00 00 02 01 12 03 #ISD_R_TAR 00 C9 07 81 02 80 00 82 01 F0 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

Name	Content in hexadecimal string
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
INSTALL_SSD2	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 44 - Data = 07 A0 00 00 01 51 53 50 08 A0 00 00 01 51 53 50 41 10 A0 00 00 05 59 10 10 01 73 73 64 32 11 22 33 01 03 80 80 00 1C EA 11 80 0F 01 00 00 01 00 00 00 02 01 12 03 11 22 33 00 C9 07 81 02 80 00 82 01 F0 00 -LE = 00
INSTALL_SSD3	- CLA = 80 - INS = E6 - P1 = 0C - P2 = 00 - LC = 44 - Data = 07 A0 00 00 01 51 53 50 08 A0 00 00 01 51 53 50 41 10 A0 00 00 05 59 10 10 01 73 73 64 33 44 55 66 01 03 80 80 00 1C EA 11 80 0F 01 00 00 01 00 00 00 02 01 12 03 44 55 66 00 C9 07 81 02 80 00 82 01 F0 00 -LE = 00

Name	Content in hexadecimal string
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
MASTER_DEL_ISDP1_INV_SDIN	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = {L} - Data = 4F 10 #ISD_P_AID1 B6 {L} 42 {L} #ISD_P_SIN 45 {L} #ISD_P_RID 5F 20 {L} #ISD_P_PROV_ID 93 01 #TOKEN_ID 9E 10 {TOKEN_VALUE} - LE = 00

Name	Content in hexadecimal string
MASTER_DEL_ISDP1_INV_SIN	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = {L} - Data = 4F 10 #ISD_P_AID1 B6 {L} 42 {L} #ISD_P_RID 45 {L} #ISD_P_SDIN 5F 20 {L} #ISD_P_PROV_ID 93 01 #TOKEN_ID 9E 10 {TOKEN_VALUE} - LE = 00
MASTER_DEL_ISDP1_RID	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = {L} - Data = 4F 10 #ISD_P_AID1 B6 {L} 42 04 #ISD_P_SIN 45 08 #ISD_P_SDIN 5F 20 05 #ISD_P_RID 93 01 #TOKEN_ID 9E 10 {TOKEN_VALUE} - LE = 00
MASTER_DEL_ISDP1_NO_PROV_ID	- CLA = 80 - INS = E4 - P1 = 00 - P2 = 40 - LC = {L} - Data = 4F 10 #ISD_P_AID1 B6 {L} 42 04 #ISD_P_SIN 45 08 #ISD_P_SDIN 93 01 #TOKEN_ID 9E 10 {TOKEN_VALUE} - LE = 00

Name	Content in hexadecimal string
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_CHANGE_DEFAULT	E1 {L} 4C 10 #EID 4D 01 02 4E 02 {NOTIF_NUMBER} 2F 10 #DEFAULT_ISD_P_AID see Note 1
NOTIF_PROFILE_EMERGENCY	E1 {L} 4C 10 #EID 4D 01 06 4E 02 {NOTIF_NUMBER} 2F 10 #DEFAULT_ISD_P_AID see Note 1
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

Name	Content in hexadecimal string
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
OPEN_SCP81_SESSION_WITH_NO_IP_ADDRESS	81 {L} 83 {L} 84 {L} 35 07 #BEARER_DESCRIPTION 39 02 #BUFFER_SIZE 47 0A #NAN_VALUE 3C 03 02 #TCP_PORT 89 {L} 8A 09 #ADMIN_HOST 8B {L} #AGENT_ID 8C 10 #ADMIN_URI

Name	Content in hexadecimal string
OPEN_SCP81_WITH_RETRY	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 86 {L} 00 02 A5 03 00 00 10 89 {L} 8A 09 #ADMIN_HOST 8B {L} #AGENT_ID 8C 10 #ADMIN_URI
PUTKEY_SCP81	- CLA = 80 - INS = D8 - P1 = 00 - P2 = 81 - LC = {L} #SCP81_KVN 85 11 10 #NEW_SCP81_PSK (see Note 2) 03 {NEW_SCP81_PSK KCV} (see Note 3) 88 11 10 #PSK_DEK (see Note 4) 03 {PSK_DEK KCV} (see Note 3) - LE = 00
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

Name	Content in hexadecimal string
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
SELECT_SSD3	- CLA = 00 - INS = A4 - P1 = 04 - P2 = 00 - LC = 10 - Data = A0 00 00 05 59 10 10 01 73 73 64 33 44 55 66 01 - LE = 00
SET_EMERGENCY	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 15 - Data = 3A 09 12 4F 10 #ISD_P_AID1

Name	Content in hexadecimal string
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
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

Name	Content in hexadecimal string
STORE_DP_CERTIF	- CLA = 80 - INS = E2 - P1 = 09 - P2 = 00 - LC = {LC} - Data = 3A 01 {L} #VALID_SM_DP_CERTIFICATE - LE = 00
STORE_DNS_PARAM	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A5 {L} 81 {L} #SM-SR_FQDN A2 {L} 3E {L} #DNS_IP 82 02 #DNS_PORT
STORE_DNS_PARAM_ERASE	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 02 A5 00

Name	Content in hexadecimal string
STORE_HTTPS_PARAM	- CLA = 80 - INS = E2 - P1 = 90 - P2 = 00 - LC = {L} - Data = A5 {L} 84 {L} 3C 03 02 #TCP_PORT 3E 05 21 #IP_VALUE 39 02 #BUFFER_SIZE 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_HTTPS_PARAM_NO_IP_ADDRESS	- CLA = 80 - INS = E2 - P1 = 90 - P2 = 00 - LC = {L} - Data = A5 {L} 84 {L} 3C 03 02 #TCP_PORT 39 02 #BUFFER_SIZE 89 {L} 8A 09 #ADMIN_HOST 8B {L} #AGENT_ID 8C 10 #ADMIN_URI
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

Name	Content in hexadecimal string
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
STORE_SIN	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 09 - Data = 00 70 06 42 04 #ISD_P_SIN

Name	Content in hexadecimal string
STORE_SMS_PARAM	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 0C - Data = 3A 07 09 A3 07 81 05 #DEST_ADDR
STORE_SMS_PARAM_ISDPS	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A3 {L} 81 05 #DEST_ADDR A2 {L} 81 03 #DEFAULT_ISD_P_ID 82 {L} #DEST_ADDR2 A2 {L} 81 03 #ISD_P_ID1 82 {L} #DEST_ADDR3
STORE_SMS_PARAM_ISDP	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A3 {L} 81 05 #DEST_ADDR A2 {L} 81 03 #DEFAULT_ISD_P_ID 82 {L} #DEST_ADDR2

Name	Content in hexadecimal string
STORE_SMSCATTP_PARAM	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 38 - Data = 3A 07 35 A2 28 35 07 #BEARER_DESCRIPTION 47 0A #NAN_VALUE 0D 06 #LOGIN 0D 09 #PWD A0 09 06 07 #TON_NPI #DIALING_NUMBER
STORE_HTTPSSMS_PARAM	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = 38 - Data = 3A 07 35 A1 28 35 07 #BEARER_DESCRIPTION 47 0A #NAN_VALUE 0D 06 #LOGIN 0D 09 #PWD A0 09 06 07 #TON_NPI #DIALING_NUMBER
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

Name	Content in hexadecimal string
STORE_SMS_PARAM_MNO1	- CLA = 80 - INS = E2 - P1 = 88 - P2 = 00 - LC = {L} - Data = 3A 07 {L} A0 {L} 06 07 #TON_NPI #DIALING_NUMBER 81 01 #PID 82 01 #DCS
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
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

Name	Content in hexadecimal string
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. *Note 2: #NEW_SCP81_PSK SHALL be encrypted as defined in GlobalPlatform Amendment B [18]* *Note 3: Key check value (KCV) of #NEW_SCP81_PSK and #PSK_DEK SHALL be computed as defined in [2]* *Note 4: #PSK_DEK SHALL be encrypted with the session KEK key of the key set used to open the SCP session as defined in [3]*	

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_PUTKEY	AB {L} 80 02 00 01 23 {L} ... 90 00 -- *any response data MAY be returned* 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

Name	Content in hexadecimal string
R_AB_02RC	AB {L} 80 02 00 02 23 {L} 85 {L} {RC} 90 00 see Note 2
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_6A88	AB 08 80 02 00 01 23 02 6A 88 see Note 2

Name	Content in hexadecimal string
R_AB_9000	AB 08 80 02 00 01 23 02 90 00 see Note 2
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

Name	Content in hexadecimal string
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
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

Name	Content in hexadecimal string
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_EM	AB 1F 80 02 00 01 23 19 E3 15 4F 10 #ISD_P_AID1 #ISD_P_ATTRIBUTE 01 02 90 00 see Note 2

Name	Content in hexadecimal string
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
R_AB_NOTIF	AB 0A 80 02 00 01 23 04 80 00 90 00 see Note 2

Name	Content in hexadecimal string
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_RC	AB {L} 80 02 00 01 23 {L} 85 {L} {RC} 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

Name	Content in hexadecimal string
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
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

Name	Content in hexadecimal string
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

Name	Content in hexadecimal string
R_AF_E3_ISDP1_EM	AF 80 23 19 E3 15 4F 10 #ISD_P_AID1 #ISD_P_ATTRIBUTE 01 02 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
R_AF_RECEIPT	AF 80 23 {L} 86 {L} {RECEIPT} 90 00 00 00
R_AF_SCP03T_PP_01	AF 80 9F 47 01 01 00 00
R_AF_SCP03T_PP_02	AF 80 9F 47 01 02 ..00 00
R_CASD_SC2B	7F 21 {L} #CASD_CERTIFICATE_SC2B 90 00

Name	Content in hexadecimal string
R_CASD_SC3	7F 21 {L} #CASD_CERTIFICATE_SC3 90 00
R_E4E_SUCCESS	80 01 00
R_E4E PROFILE_REF	80 01 08
R_E4E ALREADY_ENABLED	80 01 09
R_E4E ALREADY_DISABLED	80 01 0A
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}
R_SCP03T_PROF_PROT_OK	87 00

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

Direction	Sequence / Description
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 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 • 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 GSMA CI Certificate used to verify certificates issued and signed by the CI. • A public key (PK.CI.ECDSA), part of that GSMA CI 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 GSMA CI 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
3.1.5	The notification of “First network attachment” has been generated by the eUICC and confirmed by the SM-SR.
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 A / 2.1	This specification mandates usage of pre-defined namespace prefixes: “ds” and “rps3” for XML elements which are used as signature materials.

Chapter	Initial state
Annex B / 2.1.3	XML elements which are used as signature materials SHALL be trimmed.
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).
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 25 and 26 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 ProfileComponent 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

ID	Source	Chapter	Support	Description	Functional group
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
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_REQ58_1	[2]	4.1.3.3	M	The command and response TLVs are protected in the same way as SCP03 APDUs, using either: • the SCP03t sessions keys resulting from the secure channel initiation or • random keys which replaces session keys.	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

ID	Source	Chapter	Support	Description	Functional group
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 H. 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 Operator 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
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

ID	Source	Chapter	Support	Description	Functional group
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 [4] and 3GPP TS 31.115[13] 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 and optionally for the SMS for HTTPS session triggering described in section 2.4.3.1), • 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. The SM-SR SHALL verify that the counter value of the PoR is the same as the counter value sent in the command packet.	eUICC Management
EUICC_REQ21_1	[2]	2.4.3	M	The SM-SR MAY choose to request a PoR or not for this special SMS, and set the SPI2 byte of the SMS accordingly.	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
PROC_REQ23	[2]	2.4.4.5	C	If supported and if correctly configured by SM-SR and eUICC, the ISD-R MAY request a DNS resolution to retrieve the IP Address of the SM-SR.	Procedure Flow
PROC_REQ24	[2]	2.4.5	C	DNS resolution is an optional feature that is triggered only when: •The eUICC includes a DNS resolver Client configured to initiate the DNS queries to server •The SM-SR relies upon a DNS Resolver Server able to provide the IP address associated to the domain name sent by the client query •The eUICC determines that it has to resolve the IP address of the SM-SR server	Procedure Flow
PROC_REQ25	[2]	2.4.5.2	C	The DNS resolver of SM-SR and eUICC shall: •Be compliant to RFC 1035 and RFC 3596 defining the Domain Name System and protocol •Support Query type A (IPv4) and AAAA (IPv6) •Use UDP protocol •Support only Recursive mode: the DNS resolver Server SHALL recursively resolve the given FQDN query, meaning that the answer SHALL contain all the available IP addresses •Send short responses: any response returned by DNS Server must fit in one UDP packet	Procedure Flow
PROC_REQ26	[2]	2.4.5.3	C	The DNS resolution process must be compliant with the Figure 10 and with the procedure described in this section.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ62	[2]	2.4.5.1	C	If: • the eUICC is requested to open an HTTPS session and • the eUICC supports DNS resolution and • the ISD-R has no IP address configured in the Connection Parameters of its Administration Session Triggering Parameters (as defined by Global Platform Amendment B [8]) and • the ISD-R has a FQDN, and IP addresses of DNS servers, configured in DNS parameters.the ISD-R has not already resolved the FQDN to an IP address, or has resolved it but has reasons to consider the resolved value is stale then the eUICC shall perform a DNS resolution as described in the procedure 2.4.5.3 to retrieve the IP address(es) of the SM-SR server.	eUICC Management
EUICC_REQ63	[2]	2.4.5.1	C	The eUICC MAY also support other heuristics to determine that DNS resolution is needed and to which DNS servers to send the DNS queries.	eUICC Management
EUICC_REQ64	[2]	4.1.1.10	C	Each of the Tag 'A3', 'A4' and 'A5', shall be used to create or update the complete set of addressing parameters for corresponding protocol as defined in table bellow.	eUICC Management
EUICC_REQ65	[2]	4.1.1.10	C	The values of the profile-specif connectivity parameter, used by the eUICC to open the BIP channel to communicate with the DNS Resolver Server, are those defined in the HTTPS Connectivity Parameters of the currently Enabled ISD-P defined in Table 95.	eUICC Management
EUICC_REQ66	[2]	4.1.1.10	C	If the SM-SR does not support a DNS Resolver Server, then it shall set the IP address in the HTTPS Connectivity Parameters of the ISD-R as defined in GlobalPlatform Card Specification Amendment B [8].	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
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 4.1.3.3), 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
EUICC_REQ28	[2]	4.1.1.11	M	ES5: HandleDefaultNotification A protocol priority order for default notification MAY be defined for every Profile during profile installation or download, and updated using the functions defined in 4.1.2.2 and 4.1.3.4. This protocol priority order specifies which protocols to use, and in which order, among SMS, HTTPS and CAT_TP. If not defined for a Profile, the default priority order is set as follow: SMS, HTTPS, CAT_TP.	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_REQ1_1	[2]	3.1.1	M	ISD-P creation procedure: The SM-SR shall create a new Profile entry for the EIS having a state "In-Creation". The Profile with this state SHALL NOT appear in the EIS returned on ES3.GetEIS and ES4.GetEIS.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
PROC_REQ1_2	[2]	3.1.1	M	ISD-P creation procedure: In case the SM-SR does not receive a function execution response from the eUICC (e.g. due to a disrupted connection), the SM-SR SHALL trigger ES5.DeleteISDP function on the targeted ISD-P and update the EIS by removing the new Profile entry with status "In Creation" from the EIS accordingly.	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 14 must be called when an error occurs during the key-establishment procedure or during the steps 1 to 11 of the Profile Download and Installation procedure (before the optional enabling of the Profile). 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_REQ5_1	[2]	3.2.1		Profile Enabling process: If the previously Enabled Profile (now the Disabled) has the Fall-Back Attribute, and its POL1 contains the rule “Profile deletion is mandatory when its state is changed to disabled”, this rule SHALL be ignored according to Sections 2.4 and 3.6.3.2 in GSMA Remote Provisioning Architecture for the Embedded UICC [1], and the procedure SHALL continue at step 10.	Procedure Flow
PROC_REQ6	[2]	3.2.2	M	The Connectivity failure case described in Figure 16 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
PROC_REQ7	[2]	3.3.1	M	The Profile Enabling via SM-DP must be compliant with the Figure 17 and with the procedure described in this section.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
PROC_REQ8	[2]	3.3.2	M	The connectivity failure case described in Figure 18 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 19 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 20 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 21 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 22 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 24 and with the procedure described in this section.	Procedure Flow
PROC_REQ13_1	[2]	3.8	M	The length of the Random Challenge SHALL be 16 or 32.	Procedure Flow
PROC_REQ13_2	[2]	3.8	M	If for any reason the procedure fails or expires on SM-SR2 before starting step 24, SM-SR2 SHALL delete the EIS from its database and send an error to SM-SR1, and SM-SR1 SHALL forward the error to the Initiator Operator.	Procedure Flow
PROC_REQ13_3	[2]	3.8	M	In case the procedure expires on SM-SR1 side after step 22, even before the procedure completes or expires on SM-SR2, SM-SR1 SHALL inform the Initiator Operator. The Operator MAY then retry the procedure from step 1 or from step 4.	Procedure Flow
PROC_REQ14	[2]	3.9	M	The eUICC registration process must be compliant with the Figure 24 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 26 and with the procedure described in this section.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
PROC_REQ17	[2]	3.12	M	The POL1Update by Operator process must be compliant with the Figure 27 and with the procedure described in this section.	Procedure Flow
PROC_REQ18	[2]	3.13	M	The Connectivity Parameters Update by Operator must be compliant with the Figure 28 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 29 and with the procedure described in this section.	Procedure Flow
PROC_REQ20	[2]	3.15.1	M	The Default Notification Procedure using SMS must be compliant with the Figure 30 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 31 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.17.1	[2]	3.17.1	M	The Profile Enabling via M2M SP process must be compliant with the Figure 33 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.20.1	[2]	3.20.1	M	The Set Profile Lifecycle Management Authorisation (PLMA) process must be compliant with the Figure 37 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.20.2	[2]	3.20.2	M	The Set PLMA via SM-DP process must be compliant with the Figure 38 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.20.5	[2]	3.20.5	M	The Retrieve PLMA by M2M SP process must be compliant with the Figure 41 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.25_1	[2]	3.25	O	The Emergency Profile Attribute management process must be compliant with the Figure 325-A and with the procedure for “case 1” (first Emergency Profile) described in this section.	Procedure Flow
PROC_REQ_3.26_1	[2]	3.26	O	The Emergency Profile Attribute management via the M2M SP process must be compliant with the Figure 326-A and with the procedure for “case 1” (first Emergency Profile) described in this section.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
PROC_REQ_3.27_1	[2]	3.27	M	The Fall-Back Attribute process must be compliant with the Figure 327 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.27_2	[2]	3.27	M	The Operator1 needs to grant PLMA for Operator2 in order to authorise to set the Fall-Back Attribute in Operator1 owned Profile, as a consequence, Operator2 owned Profile has the Fall-Back Attribute un-set (see GSMA Remote Provisioning Architecture for Embedded UICC [1] section 3.5.19). NOTE There is no operation that explicitly un-sets the Fall-Back Attribute on a Profile. The Fall-Back Attribute is only un-set as the consequence of setting the Fall-Back Attribute on another Profile.	Procedure Flow
PROC_REQ_3.29_1	[2]	3.29	M	The Fall-Back Attribute via M2M SP process must be compliant with the Figure 329 and with the procedure described in this section.	Procedure Flow

ID	Source	Chapter	Support	Description	Functional group
PF_REQ3	[2]	4.1.1.1	M	ES5: CreateISDP Description: This function creates an ISD-P on the eUICC. Parameters: • ISD-P-AID • Cumulative Granted Non Volatile Memory for the ISD-P (optional) Prerequisite: • The SM-SR has assigned an ISD-P-AID. 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

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 Prerequisites: • SM-SR has checked that POL2 of both the currently Enabled Profile and the target Profile allow this action. • The target Profile SHALL NOT be the Test Profile 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 • Verify that the target Profile is not the Test Profile • If any of these verifications fail, terminate the command with an error status word • If the current profile has been enabled by the activation of the Fall-Back Mechanism then • If the target Profile is not the previously Enabled profile and the POL1 of the previously enabled profile does not allow its own disabling, or contains the rule “Profile deletion is mandatory when its state is changed to disabled”, 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.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
				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 or Profile is the Test Profile. '69 E1': POL1 of the currently Enabled Profile prevents this action or of the previously enabled profile prevents this action.	

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 Prerequisites: • SM-SR has checked that POL2 allows this action • The target Profile SHALL NOT be the Test 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 Test Profile • 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 or Profile is the Test Profile. ’69 E1’: POL1 of the Profile prevents disabling.	Platform Management

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 Prerequisites: • SM-SR SHALL check that POL2 allows this action • The target Profile SHALL NOT be the Profile with the Fall-Back Attribute set • The target Profile SHALL NOT be the Test Profile Function flow Upon reception of the DELETE command, the eUICC SHALL: • Verify that POL1 of the target Profile allows its deletion. This includes, if the target Profile has been Disabled by the activation of the Fall-Back Mechanism described in section **Error! Reference source not found.**, verify that POL1 of the target Profile allows D isabling. • Verify that the target Profile is not the Profile with Fall-Back Attribute set • Verify that the target Profile is not the Test Profile • 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 or Profile is the Test Profile. ’69 E1’: POL1 of the Profile prevents deletion (including the case where the Profile has	Platform Management

ID	Source	Chapter	Support	Description	Functional group
				been Disabled by the activation of the Fall-Back Mechanism, and its POL1 prevents disabling).	
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

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 Prerequisites: • The target Profile shall notSHALL NOT be the Profile which has the Fall-BackFall-Back Attribute set. • The target Profile shallSHALL be in the Disabled state. 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). This includes verifying the signature of the Token, and verifying that the values of tags 42, 45, and 5F20 in the Token match the corresponding values in 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_REQ8_1	[2]	4.1.1.6	M	The eUICC SHALL support setting the value of tags 42, 45, and 5F20 by a STORE DATA command defined in GlobalPlatofrm Card Specification [6].	Platform Management
PF_REQ8_2	[2]	4.1.1.6	M	If the value of tag 5F20 is not set by the SM-DP, the default value SHALL be the value of the RID of ISD-P defined in section 2.2.3.	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 • If the currently Enabled profile is the Profile with the Fall-Back Attribute set, and has been Enabled by the activation of the Fall-Back Mechanism, and the previously Enabled Profile has either of the POL1 rules “Disable not allowed” or “Profile deletion is mandatory when its state is changed to Disabled” set, then the eUICC SHALL prevent the execution of the function “Set Fall-Back Attribute” 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. Processing State Returned in the Response Message: As defined in GlobalPlatform Card Specification [6] section 11.11.3.2. , with the following addition: ’69 E1’: POL1 of the Profile Disabled by the activation of the Fall-Back Mechanism prevents this action.	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 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
EUICC_REQ26_1	[2]	4.1.1.10	M	Each of the Tag 'A3', 'A4' and 'A5', SHALL be used to create or update the complete set of addressing parameters for corresponding protocol. This structure can contain as many TLVs with tag ‘A2’ as there are ISD-Ps. The SM-SR is responsible to update this list as it sees fit when a new ISD-P is created or after an SM-SR change. The SM-SR MAY use Tag ‘A5’ with a length of zero to erase the DNS parameters.	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
EUICC_REQ29_1	[2]	4.1.1.13	M	ES5: SetEmergencyProfileAttribute Description: This function sets the Emergency Profile Attribute for one Profile on the target eUICC. Parameters: • ISD-P AID Prerequisites: • The target profile SHALL NOT be enabled. • The target Profile SHALL NOT have the Fall-Back Attribute set. Function flow Upon reception of the STORE DATA command, the eUICC SHALL: • Verify that the target Profile has not the Fall-Back Attribute set • Set the Emergency Profile Attribute for the target Profile • Remove the Emergency Profile Attribute from the Profile that has the attribute currently set 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 4.1.1.13. 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
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

ID	Source	Chapter	Support	Description	Functional group
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
EUICC_REQ4_1_3_3_1	[2]	4.1.3.3	M	Profile protection: Profile protection SHALL performed using either: • Session keys (S-ENC, S-MAC, S-RMAC) resulting from the key agreement with eUICC (INITIALIZE UPDATE and EXTERNAL AUTHENTICATE). Or • random keys per Profile (denoted PPK-ENC, PPK-MAC, PPK-RMAC in this document), generated by the SM-DP. The eUICC SHALL be able to support both modes	eUICC Management
EUICC_REQ4_1_3_3_2	[2]	4.1.3.3	M	PPK-ENC, PPK-MAC and PPK-RMAC SHALL have the same length as S-ENC, S-MAC, S-RMAC.	eUICC Management
EUICC_REQ4_1_3_3_3	[2]	4.1.3.3	M	Session keys and, if used, the random keys SHALL only be used in the Profile download process. They SHALL be deleted on the eUICC at the latest at the end of the process.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ4_1_3_3_4	[2]	4.1.3.3	M	Replace session key command TLV This command is used, during the download of a Protected Profile, to replace the SCP03t session keys (S-ENC, S-MAC and S-RMAC) by a new set of session keys (typically the PPK-ENC, PPK-MAC and PPK-RMAC) protecting the command and response TLVs. Note that all keys (S-ENC, S-MAC and S-RMAC) have to be replaced. This command doesn't allow to replace only a part of the session keys. The response SHALL be encrypted by PPK-ENC and MAC-ed by PPK-RMAC, where PPK-RMAC SHALL be different for each download attempt of the same Profile. Command Message The Replace session key command TLV SHALL be coded according to Tables defined in section 4.1.3.3.	eUICC Management
PF_REQ4_1_3_3_1	[2]	4.1.3.3	M	When using random keys for profile protection, the Replace session key command SHALL be sent directly before the SCP03t command TLVs containing the protected profile package (tag 86).	Platform Management
PF_REQ4_1_3_3_2	[2]	4.1.3.3	M	When using session keys for profile protection, the Replace session key command SHALL NOT be present.	Platform Management
EUICC_REQ4_1_3_3_5	[2]	4.1.3.3	M	On reception of the replace session key command the eUICC SHALL: • Verify that the new keys are of same length as the old keys. If not the eUICC SHALL return an error ('01'), and the loading of the Profile SHALL be aborted. • Replace the current session keys with the new set of keys	eUICC Management
EUICC_REQ4_1_3_3_6	[2]	4.1.3.3	M	Once the command is successfully executed, the eUICC SHALL use this new set of keys for decryption and MAC verification of subsequent SCP03t blocks of data, and encryption and MACing of responses. The key type of the new set of keys is the same as the session keys they replace.	eUICC Management
EUICC_REQ4_1_3_3_7	[2]	4.1.3.3	M	If the command message is accepted, a Response TLV with tag '87' and length zero SHALL be returned. This TLV does not return an R-MAC.	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ4_1_3_3_8	[2]	4.1.3.3	M	Replace session key command TLV In case of an error, tag '9F47' is used (see NOTE 1 above). The following values are defined: • '01': error in length or structure of command data • '02': security error	eUICC Management
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_REQ31_1	[2]	4.1.4.1	O	ESX.LocalEnableEmergencyProfile Description: This function is used by the Device to locally enable the Emergency Profile. The eUICC SHALL NOT send notifications to the SM-SR. The Emergency Profile SHOULD remain enabled even after a restart of the Device. It is up to the Device to disable the Emergency Profile. Prerequisites: • A Profile with the Emergency Profile Attribute set exists on the eUICC. • The Profile with the Emergency Profile Attribute set is not already enabled. Command Description: The Local Enable and Local Disable of a Profile with the Emergency Profile Attribute set is realised by using the ENVELOPE command with a dedicated Tag (as defined in the Table defined in section 4.1.4.1).	Profile Management
EUICC_REQ31_2	[2]	4.1.4.2	O	ESX.LocalDisableEmergencyProfile Description: This function is used by the Device to locally disable the Emergency Profile and enable the previously enabled Profile. In case the Local Disable fails, the Fall-Back Mechanism SHALL be activated. The Fall-Back Mechanism SHALL consider that its previously enabled Profile is the Profile that was enabled before the Local Enable of the Emergency Profile. After disabling the Emergency Profile the eUICC MAY send a notification to the SM-SR (see 4.1.1.11). Prerequisites: • A Profile with the Emergency Profile Attribute set exists on the eUICC • The Profile with the Emergency Profile Attribute set is enabled	Profile 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 Operator 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
PF_REQ_5.3.12	[2]	5.3.12	M	ES2: AuditEIS Description: This function allows the Operator to retrieve the up to date information for the Operator's Profiles. The SM-DP SHALL forward the request to the SM-SR. Input data described in Tables present in section 5.3.12.	Platform Management

PM_REQ11	[2]	5.3.2	M	ES2: DownloadProfile Description: This function allows the Operator to request that the SM-DP downloads a Profile, identified by its ICCID, via the SM-SR identified by the Operator 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 **Error! Reference source not found.** indicating that the Profile has not been downloaded before the expiration of the specified Validity Period. • A 'Function execution status' indicating 'Failed' with a status code as defined in section **Error! Reference source not found.** or a specific status code as defined in the table below, indicating that the Profile has not been downloaded. • A 'Function execution status' indicating 'Executed_WithWarning' indicating that the Profile has been downloaded successfully, but the optional Enable has failed or expired. In this case, the Status Code and the eUICCResponseData if available, are the ones reporting the failure or expiration of the Enable Input/Output data described in Tables present in section 5.3.2.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ12	[2]	5.3.3	M	ES2: UpdatePolicyRules Description: This function allows the Operator 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
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

ID	Source	Chapter	Support	Description	Functional group
PF_REQ12	[2]	5.3.5	M	ES2: EnableProfile Description: This function allows the Operator 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 Operator 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 Operator. 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 'Executed-WithWarning', with a status code as defined in section 5.4.9, indicating that the Profile has been disabled on the eUICC, and deleted after application of a POL1 or POL2 rule • 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 5.4.9 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 Operator 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

ID	Source	Chapter	Support	Description	Functional group
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 Operator 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
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 Operator 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 Operator. 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

ID	Source	Chapter	Support	Description	Functional group
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
PF_REQ_5.4.23	[2]	5.4.23	O	Description: This function allows the SM-DP authorised by the Operator to request the setting of the Emergency Profile Attribute on the targeted Profile to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. The SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. The SM-SR SHALL verify that the request is • Either sent on behalf of an Operator owning the targeted Profile or • Sent on behalf of an Operator that is not the owner of the targeted Profile, but the Operator owning the targeted Profile has granted a PLMA allowing the operation "SetEmergencyProfileAttribute" to the Operator requesting the operation. If one Profile currently has the Emergency Profile Attribute set, the SM-SR SHALL verify that the Operator owning the Profile with the Emergency Profile Attribute set has granted a PLMA authorising the operation "UnsetEmergencyProfileAttribute" to the Operator requesting the operation. The SM-SR MAY provide additional verifications. The SM-SR receiving this request SHALL process it according to "Emergency Profile Attribute Management" procedure described in the section 3.25 of this specification. After setting the Emergency Profile Attribute, the SM-SR SHALL add or update the AdditionalProperty 'gsma.ESIM.EmergencyProfile.AID' of the EIS. This function may return: • A 'Function execution status' with 'Executed- Success' indicating that the Emergency Profile Attribute has been set on the targeted Profile. • 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 of a specific status code as defined in the table here after. Input data described in Tables present in section 5.4.23.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.4.24	[2]	5.4.24	O	ES3: HandleEmergencyProfileAttributeSetNotification Description: This function SHALL be called to notify that the Emergency Profile Attribute has been set on the Profile identified by its ICCID on the eUICC identified by its EID. The SM-SR SHALL send this notification to all SM-DP servers that match one or the other of the following conditions: • The SM-DP can relay the notification to the Operator that owns the Profile, and the Operator has opted to receive such notifications (see section 3.21) • The SM-DP can relay the notification to another Operator, and the Operator owner of the Profile has granted the other Operator with a PLMA authorising this Operation “HandleEmergencyProfileAttributeSetNotification”. • The SM-DP can relay the notification to any Operator having a Profile on this eUICC. In this case Identification of the Profile that has the Emergency Profile Attribute set and Identification of the Operator owner of the Profile that has the Emergency Profile Attribute set are optional. ICCID may be not enough to identify right address of recipient, SM-SR should map it internally to Operator notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case multiple handlers are served, the SM-SR SHOULD ensure completionTimestamp to be equal for every message. What is performed by the Operator receiving this notification is out of scope of this specification. Input data described in Tables present in section 5.4.24.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.4.28	[2]	5.4.28	M	ES3: HandleProfileFallBackAttributeUnsetNotification Description: This function SHALL be called to notify that the Fall-Back Attribute has been unset on the Profile identified by its ICCID on the eUICC identified by its EID. The SM-SR SHALL send this notification to all SM-DP servers that match one or the other of the following conditions: • The SM-DP can relay the notification to the Operator that owns the Profile, and the Operator has opted to receive such notifications (see section 3.21) • The SM-DP can relay the notification to another Operator, and the Operator owner of the Profile has granted the other Operator with a PLMA authorising this Operation “HandleProfileFallBackAttributeUnsetNotification”. ICCID may be not enough to identify right address of recipient, SM-SR should map it internally to Operator notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case multiple handlers are served, the SM-SR SHOULD ensure completionTimestamp to be equal for every message. What is performed by the Operator receiving this notification is out of scope of this specification. Input data described in Tables present in section 5.4.28.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.5.18	[2]	5.5.18	O	ES4: SetEmergencyProfileAttribute Description: This function allows an Operator or an M2M SP authorised by the Operator via PLMA to request the setting of the Emergency Profile Attribute on the targeted Profile to the SM-SR in charge of the management of the targeted eUICC, eUICC being identified by its EID. The SM-SR SHALL verify that the request is: • Either sent by an Operator owning the targeted Profile or • Sent by an M2M SP, but the Operator owning the targeted Profile has granted a PLMA allowing the operation “SetEmergencyProfileAttribute” to the M2M SP requesting the operation. If one Profile currently has the Emergency Profile Attribute set, the SM-SR SHALL verify that the Operator owning the Profile with the Emergency Profile Attribute set has granted a PLMA authorising the operation “UnsetEmergencyProfileAttribute” to the Operator requesting the operation. The SM-SR MAY provide additional verifications. The SM-SR receiving this request SHALL process it according to “Emergency Profile Attribute Management” procedure described in sections 3.25 and 3.26 of this specification. After setting the Emergency Profile Attribute, the SM-SR SHALL add or update the AdditionalProperty ‘gsma.ESIM.EmergencyProfile.AID’ of the EIS. This function may return: • ‘Function execution status’ with ‘Executed- Success’ indicating that the Emergency Profile Attribute has been set on the targeted Profile • 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 of a specific status code as defined in the table here after Input data described in Tables present in section 5.5.18.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.5.19	[2]	5.5.19	M	ES4: HandleProfileFallBackAttributeSetNotification Description: This function SHALL be called to notify that the Emergency Profile Attribute has been set on the Profile identified by its ICCID on the eUICC identified by its EID. The SM-SR SHALL send this notification to all Operator and M2M SP servers that match one or the other of the following conditions: • The Operator that owns the Profile, and the Operator has not set an ONC to discard such notifications (see section 3.21). • The M2M SP, where the Operator owner of the Profile has granted the M2M SP with a PLMA authorising this Operation "HandleEmergencyProfileAttributeSetNotification". • Any Operator having a Profile on this eUICC. In this case identification of the Profile that has the Emergency Profile Attribute set and Identification of the Operator owner of the Profile that has the Emergency Profile Attribute set are optional. ICCID may be not enough to identify right address of recipient, the SM-SR should map it internally to Operator or M2M SP notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case multiple handlers are served the SM-SR SHOULD ensure completionTimestamp to be equal for every message. What is performed by the Operator or M2M SP receiving this notification is out of scope of this specification. Input data described in Tables present in section 5.5.19.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.5.21	[2]	5.5.21	M	ES4: SetFallBackAttribute Description: This function allows the Operator owner of the Profile or an M2M SP authorised by the Operator owner of the Profile, to request the SM-SR to set the Fall-Back Attribute on a Profile in a specified eUICC, eUICC being identified by its EID. On reception of this 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 Operator, or by an Operator that had granted a PLMA that authorises the requesting M2M SP to perform the operation “setFallbackAttribute” on a Profile that matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria) • The Operator owning the Profile which currently has the Fall-Back Attribute set has granted a PLMA that authorises the requesting Operator or M2M SP to perform the operation “UnsetFallBackAttribute”, and that the Profile that currently has the Fall-Back Attribute set matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria). The SM-SR MAY provide additional verifications. The SM-SR receiving this request SHALL process it according to “Fall-Back Attribute Management” procedures described in section 3.27 and 3.29 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the Fall-Back Attribute has been set on the targeted Profile. • 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 of a specific status code as defined in the table here after. Input data described in Tables present in section 5.5.21.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.5.22	[2]	5.5.22	M	ES4: HandleProfileFallBackAttributeSetNotification Description: This function SHALL be called to notify the Operator and the M2M SP that the Fall-Back Attribute has been set on the Profile identified by its ICCID on the eUICC identified by its EID. The SM-SR SHALL send this notification to: • the Operator owning the Profile, if it has not set an ONC to not receive those notifications • the M2M SP SP (including, another Operator that is not the owner of the Profile), if the Operator owner of the Profile has granted the M2M SP with a PLMA authorising the operation “HandleProfileFallBackAttributeSetNotification” ICCID may be not enough to identify right address of recipient, the SM-SR should map it internally to Operator or M2M SP notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case multiple handlers are served the SM-SR SHOULD ensure completionTimestamp to be equal for every message. What is performed by the Operator or M2M SP receiving this notification is out of scope of this specification. Input data described in Tables present in section 5.5.22.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.5.23	[2]	5.5.23	M	ES4: HandleProfileFallBackAttributeUnsetNotification Description: This function SHALL be called to notify that the Fall-Back Attribute has been unset on the Profile identified by its ICCID on the eUICC identified by its EID. The SM-SR SHALL send this notification to: • the Operator owning the Profile, if it has not set an ONC to not receive those notifications • the M2M SP SP (including, another Operator that is not the owner of the Profile), if the Operator owner of the Profile has granted the M2M SP with a PLMA authorising the operation “HandleProfileFallBackAttributeUnsetNotification” ICCID may be not enough to identify right address of recipient, the SM-SR should map it internally to Operator or M2M SP notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case multiple handlers are served the SM-SR SHOULD ensure completionTimestamp to be equal for every message. What is performed by the Operator or M2M SP receiving this notification is out of scope of this specification. Input data described in Tables present in section 5.5.23.	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 SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. 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' 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. The SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. If the SM-DP provides a list of ICCID of Profiles to audit, the SM-SR SHALL verify for each profile that the Operator, on behalf of which the SM-DP requests this operation, • is either the owner of the targeted Profile or • is authorised by the Operator owning the targeted Profile(s) to perform the operation “AuditEIS” on a Profile that matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria). This SHALL also be applied if the list of ICCIDs identifies • Profiles that are owned by this Operator and / or • Profiles that are owned by other Operators. The SM-SR MAY provide additional verifications. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the 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 **Error! Reference source not found.** • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section **Error! Reference source not found.** of a specific status code as defined i n the table below.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
				Input/Output data described in Tables present in section 5.4.2.	
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
PM_REQ16_1	[2]	5.4.3	M	If the "RequiredMemory" parameter of this ES3.CreateISDP function call is equal to ‘0’, the "Cumulative Granted Non Volatile Memory" parameter SHALL NOT be used in the INSTALL command of the ES5.CreateISDP function.	Profile Management

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 (designated in the sd-aid or in the commands) is created on the eUICC and is managed by the calling SM-DP • If the SM-DP requests to send the commands to the ISD-R: the commands are allowed to be executed by ISD-R, including ISD-P key establishment as described in section **Error! Reference source not found.** NOTE1: this verification implies the parsing and analysing of the commands. NOTE2: this verification allows to prevent the SM-DP to perform arbitrary operations in the ISD-R. 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 or TLV commands as defined in this document, section **Error! R eference source not found.**. 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	Profile Management

ID	Source	Chapter	Support	Description	Functional group
				Input/Output data described in Tables present in section 5.4.4.	
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

ID	Source	Chapter	Support	Description	Functional group
PM_REQ19	[2]	5.4.6	M	ES3: UpdatePolicyRules Description: This function allows the SM-DP authorized by the Operator 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 SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. The SM-SR SHALL verify that the request is: • Either sent on behalf of an Operator owning the targeted Profile or • Sent on behalf of an Operator that is not the owner of the targeted Profile, but the Operator owning the targeted Profile has granted a PLMA allowing the operation “UpdateSubscriptionAddress” to the Operator requesting the operation The SM-SR MAY provide additional verifications. 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

ID	Source	Chapter	Support	Description	Functional group
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 SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. The SM-SR SHALL verify that the request is • Either sent on behalf of an Operator owning the targeted Profile or • Sent on behalf of an Operator that is not the owner of the targeted Profile, but the Operator owning the targeted Profile has granted a PLMA allowing the operation “EnableProfile” to the Operator requesting the operation The SM-SR MAY provide additional verifications. 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 Operator 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 SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. The SM-SR receiving this request SHALL process it according to Profile Disabling procedure described in section 3.5 of this specification. The SM-SR SHALL verify that the request is: • Either sent on behalf of an Operator owning the targeted Profile or • Sent on behalf of an Operator that is not the owner of the targeted Profile, but the Operator owning the targeted Profile has granted a PLMA allowing the operation “DisableProfile” to the Operator requesting the operation The SM-SR MAY provide additional verifications. 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 ‘Executed-WithWarning’, with a status code as defined below, indicating that the Profile has been disabled on the eUICC, and deleted after application of a POL1 or POL2 rule • 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

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 Operator. The SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. 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 Operator 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 set The SM-SR SHALL verify that the request is: • Either sent on behalf of an Operator owning the targeted Profile or • Sent on behalf of an Operator that is not the owner of the targeted Profile, but the Operator owning the targeted Profile has granted a PLMA allowing the operation “DeleteProfile” to the Operator requesting the operation The SM-SR MAY provide additional verifications. In case one of these conditions is not satisfied, the SM-SR SHALL refuse the function request and return a ‘Function execution status’ indicating ‘Failed’ with the relevant status code (see table below). The SM-SR receiving this request SHALL process it according to “Profile and ISD-P	Platform Management

ID	Source	Chapter	Support	Description	Functional group
				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 • A 'Function execution status' with 'Executed-WithWarning' indicating that the Profile has been deleted on the eUICC with a status code as defined in section **Error! R eference source not found.** (The ISD-P identified by its AID does not exist on the targeted eUICC) Input/Output data described in Tables present in section 5.4.10.	

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 Operator 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

ID	Source	Chapter	Support	Description	Functional group
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. The SM-SR SHALL send this notification to all SM-DP servers that match one or the other of the following conditions: • The SM-DP can relay the notification to the Operator that owns the Profile, and the Operator has not opted to not receive such notifications (see section **Error! Reference source not found.**). The SM-DP can relay the notification to another Operator, and the Operator owner of the Profile has granted the other Operator with a PLMA authorising this Operation “HandleProfileDisabledNotification”. ICCID MAY be not enough to identify right address of recipient; SM-SR SHOULD map it internally to Operator notification endpoint. This notification also conveys the date and time specifying when the operation has done. In case multiple handlers are served, the 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. The SM-SR SHALL send this notification to all SM-DP servers that match one or the other of the following conditions: • The SM-DP can relay the notification to the Operator that owns the Profile, and the Operator has not opted to not receive such notifications (see section **Error! Reference source not found.**) • The SM-DP can relay the notification to another Operator, and the Operator owner of the Profile has granted the other Operator with a PLMA authorising this Operation “HandleProfileEnabledNotification” ICCID MAY be not enough to identify right address of recipient; SM-SR SHOULD map it internally to Operator notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case multiple handlers are served, the SM-SR SHOULD ensure completionTimestamp to be equal for every message. Input data described in Tables present in section 5.4.13.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
EUICC_REQ34	[2]	5.4.14	M	ES3: HandleSMSRChangeNotification Description: This function SHALL be called for notifying each SM-DP authorized by the Operator 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 Operator. 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

ID	Source	Chapter	Support	Description	Functional group
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. The SM-SR SHALL send this notification to all SM-DP servers that match one or the other of the following conditions: • The SM-DP can relay the notification to the Operator that owns the Profile, and the Operator has not opted to not receive such notifications (see section **Error! Reference source not found.**) • The SM-DP can relay the notification to another Operator, and the Operator owner of the Profile has granted the other Operator with a PLMA authorising this Operation “HandleProfileDeletedNotification” 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
PF_REQ_5.4.16	[2]	5.4.16	M	ES3: SetPLMA Description: This function allows the Operator owning Profiles to grant a PLMA to an M2M SP to perform certain operations, or receive certain notifications, related to a set of Profiles, identified by a Profile Type. The SM-SR receiving this request SHALL verify that the mno-id in the PLMA matches the mno-id of the Operator on behalf of which the SM-DP declares to send this request. If the request is acceptable, the SM-SR SHALL record the PLMA. The new PLMA overwrites the previous PLMA that might have been granted with the same identifiers. From this point on, any request from the M2M SP on a Profile matching these identifiers, or any notification to the M2M SP related to a Profile matching these identifiers, SHALL be allowed or not based on the new PLMA, as described in sections 5.7.1.1, 5.7.1.2, and 5.7.1.3. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the authorisations have been configured in the SM-SR. • A ‘Function execution status’ with ‘Executed-WithWarning’ with a specific status code as defined in the table here after, indicating that the authorisations have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator. • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 or a specific status code as defined in the table here after. Input data described in Tables present in section 5.4.16.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.4.20	[2]	5.4.20	M	ES3: HandleSetPLMANotification Description: This function SHALL be called to notify an Operator (acting as an M2M SP) that a PLMA, granted by another Operator to it, has been set or updated. This notification also conveys the date and time specifying when the operation has been done. In case of multiply handlers are served the SM-SR SHOULD ensure 'completionTimestamp' to be equal for every message. Input data described in Tables present in section 5.4.20.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.4.21	[2]	5.4.21	M	ES3: SetONC Description: This function allows the Operator to configure for which of its own Profiles, associated with a Profile Type, it wants to receive which kind of status change notifications; whatever the origin of the status change is. The SM-SR receiving this request SHALL verify that the mno-id in the ONC matches the mno-id of the Operator on behalf of which the SM-DP declares to send this request. If the request is acceptable, the SM-SR SHALL record the ONC. The new ONC overwrites the previous ONC that might have been granted with the same identifiers. From this point on, any status change notification, irrespective of the cause and related to a Profile matching these identifiers, SHALL be sent or not based on the new ONC. This function may return: • A 'Function execution status' with 'Executed-Success' indicating that the notifications have been configured in the SM-SR. • A 'Function execution status' with 'Executed-WithWarning' with a specific status code as defined in the table here after, indicating that the notifications have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator. • A 'Function execution status' indicating 'Failed' with a status code as defined in section 5.1.6.4 or a specific status code as defined in the table here after. NOTE: If no Operator Notification Configuration has yet been set in the SM-SR for a given Profile Type, then the Operator will receive all notifications for status changes for its own Profiles, associated with this Profile Type, see also section 3.21 for details. Input data described in Tables present in section 5.4.21.	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' indicating 'Failed' with a status code as defined in section **Error! Reference source not found.** of a specific status code as defined in the table 176 i n section 5.5.1 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 Operator 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

ID	Source	Chapter	Support	Description	Functional group
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. For consistency within the system, it is the responsibility of the caller to ensure that all data is provided. 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 Operator, or an Operator that had granted a PLMA that authorises the requesting M2M SP to perform the operation “UpdateSubscriptionAddress” on a Profile that matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria). 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

PM_REQ25	[2]	5.5.4	M	ES4: AuditEIS Description: This function allows the Operator to retrieve the up to date EIS information. 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. If the function caller provides a list of ICCID of Profiles to audit, the SM-SR SHALL verify for each Profile that the function caller • is either the owner of the targeted Profile or • is authorised by the Operator owning the targeted Profile(s) to perform the operation “AuditEIS” on a Profile that matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria). This SHALL also be applied if the list of ICCIDs identifies • Profiles that are owned by this Operator and / or • Profiles that are owned by other Operators The SM-SR MAY provide additional verifications. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the 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 **Error! Reference source not found.** • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section **Error! Reference source not found.** of a specific status code as defined i n the table 181 in section 5.5.4 Input/Output data described in Tables present in section 5.5.4.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PM_REQ26	[2]	5.5.4	M	ES4: AuditEIS If no list of ICCIDs is provided, it is implied that all authorised Profiles in are required. The SM-SR SHALL filter the list of Profiles returned in the EIS, considering the authorisation granted by the Profile owners; for each Profile, this includes: • If the function caller is the owner of the Profile, the SM-SR SHALL include this Profile in the returned EIS. • If the function caller is not the owner of the targeted Profile, the SM-SR SHALL include the Profile in the returned EIS only if the Operator owning the Profile has granted a PLMA allowing the operation “AuditEIS” to the function caller.	Profile Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ24	[2]	5.5.5	M	ES4: EnableProfile Description: This function allows the Operator to request a Profile Enabling to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. 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 Operator or an Operator that had granted a PLMA that authorises the requesting M2M SP to perform the operation “EnableProfile” on a Profile that matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria) • 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 Operator to request a Profile Disabling to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. 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 Operator, or an Operator that had granted a PLMA that authorises the requesting M2M SP to perform the operation “DisableProfile” on a Profile that matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria) • The target Profile is in Enabled state • The POL2 of the target Profile allows the disabling 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 ‘Executed-WithWarning’, with a status code as defined below, indicating that the Profile has been disabled on the eUICC, and deleted after application of a POL1 or POL2 rule • 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

PF_REQ26	[2]	5.5.7	M	ES4: DeleteProfile Description: This function allows the Operator 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. 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 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 Operator , or an Operator that had granted a PLMA that authorises the requesting M2M SP to perform the operation "DeleteProfile" on a Profile that matches the criteria of the PLMA (see section 5.7.1.1 for the detail of the matching of the criteria) 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 • A 'Function execution status' with 'Executed-WithWarning' indicating that the Profile has been deleted on the eUICC with a status code as defined in section **Error! Reference source not found.** (The ISD-P identified by its AID does not exist on the targeted eUICC) Input/Output data described in Tables present in section 5.5.7.	Platform Management
EUICC_REQ35	[2]	5.5.8	M	ES4: PrepareSMSRChange	eUICC Management

ID	Source	Chapter	Support	Description	Functional group
				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_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. In this case, the eUICC is unambiguously managed by the new SM-SR (SM-SR2).
				• A ‘Function execution status’ with ‘Executed-WithWarning’ indicating either:
				  • that the eUICC has been successfully transferred to the new SM-SR, but additional configuration has not completed and may need to be done again. In this case, the eUICC is unambiguously managed by the new SM-SR (SM-SR2), but the new SM-SR SHALL perform such configuration operations automatically at a later point in time
				  • or that the eUICC was already managed by the new SM-SR (SM-SR2). This happens when this is the second attempt to perform the SM-SR Change, after the first attempt expired whereas it was already successful from the point of view of the new SM-SR
				• A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section **Error! Reference source not found.** of a specific status code as defined i n the Specific status code table below, to indicate that the procedure has failed or expired before the effective transfer of OTA management to the new SM-SR. In this case, the eUICC is still managed unambiguously by the current SM-SR (SM-SR1).
				• A ‘Function execution status’ indicating ‘Expired’ with the status code as defined in section **Error! Reference source not found.**, indicating that the procedure h as expired before confirming the proper transfer.
					eUICC Management

ID	Source	Chapter	Support	Description	Functional group
				Input/Output data described in Tables present in section 5.5.9.	
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, if and only if: • The recipient of the notification is the Operator owning the Profile and has not set an ONC to discard those notifications, or • The recipient of the notification is an M2M SP (including, another Operator that is not the owner of the Profile), and the Operator owner of the Profile has granted the M2M SP with a PLMA authorising the Operation "HandleProfileDisabledNotification". 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 multiple handlers are served, the SM-SR SHOULD ensure completionTimestamp to be equal for every message. Input data described in Tables present in section 5.5.10.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
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, if and only if: • The recipient of the notification is the Operator owning the Profile and has not set an ONC to discard those notifications Or • The recipient of the notification is an M2M SP (including, another Operator that is not the owner of the Profile), and the Operator owner of the Profile has granted the M2M SP with a PLMA authorising the Operation "HandleProfileEnabledNotification". 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 multiple handlers are served, the 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., if and only if: • The recipient of the notification is the Operator owning the Profile and has not set an ONC to discard those notifications or • The recipient of the notification is an M2M SP (including, another Operator that is not the owner of the Profile), and the Operator owner of the Profile has granted the M2M SP with a PLMA authorizing the Operation “HandleProfileDeletedNotification”. ICCID MAY be not enough to identify right address of recipient; SM-SR SHOULD map it internally to Operator 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
PF_REQ_5.5.16	[2]	5.5.16	M	ES4: HandleSetPLMANotification Description: This function SHALL be called to notify an M2M SP that a PLMA concerning this M2M SP has been set or updated. This notification also conveys the date and time specifying when the operation has been done. In case of multiple handlers are served the SM-SR SHOULD ensure ‘completionTimestamp’ to be equal for every message. Input data described in Tables present in section 5.5.16.	Platform Management

PF_REQ_5.5.17	[2]	5.5.17	M	ES4: GetPLMA Description: This function allows the Operator owner of Profiles to retrieve the list of PLMAs applicable to a certain Profile, or a certain Profile Type, or for a certain M2M SP. The same function can also be used by the M2M SP to retrieve the list of PLMAs granted to this M2M SP, and applicable to a certain Profile, or a certain Profile Type. The SM-SR receiving this request SHALL verify the requester is allowed to retrieve such information, and return the list of all PLMAs applicable to the specified search criterion: • If the requester is the owner of the targeted Profiles, the authorisation is implied • If the requester is an M2M SP (including, another Operator that is not the owner of the targeted Profiles), the list of PLMAs is only returned if at least a PLMA exist for this M2M SP and for the targeted Profile or Profile Type If this verification fails, the SM-SR SHALL terminate the request and return a response with the 'Function execution status' indicating 'Failed', and no PLMA. Otherwise, the SM-SR SHALL return the complete list of all PLMAs applicable to the specified search criterion; if the search criterion is on a specific Profile or Profile Type, this includes even PLMAs that are granted to an M2M SP that is not the function requester. In case the list of PLMAs is very long, the SM-SR MAY truncate the result. The caller can then issue another call to getPLMA with more restrictive criteria. NOTE The order of the PLMAs returned in the truncated list is implementation-dependant. This function may return: • A 'Function execution status' with 'Executed- Success', and additional output data providing the PLMAs. • A 'Function execution status' with 'Executed-WIthWarning', to indicate that the result was truncated, plus additional output data providing part of the list of applicable PLMAs. A 'Function execution status' indicating 'Failed' if the requester was not allowed to request this information. Input data described in Tables present in section 5.5.17.	Platform Management
EUICC_REQ38	[2]	5.6.1	M	ES7: CreateAdditionalKeySet	eUICC Management

				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, using the following rules: • The order of TLVs SHALL follow the order denoted in table 45 • The following parameters of 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. o Scenario identifier SHALL be set to '03' o Key Usage Qualifier SHALL be set to '10' (3 secure channel keys) o Key Access SHALL NOT be present, meaning a default value of '00' (The key MAY be used by the Security Domain and any associated Application) o Key Type SHALL be set to '88' (AES) o Key Length SHALL be set to '10' (16 bytes) o Key Identifier SHALL be set to '01' • The length of Initial value of sequence counter SHALL be 0, meaning the sequence counter SHALL have its default value • The SDIN (tag 45 in Table 44) SHALL be included if and only if the bit b3 of the byte of Parameter for Scenario #3 is set to 1. In this case, the value of this field SHALL be the value of the SDIN of the ISD-RThe value of other parameters are provided by the new SM-SR 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.	

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. In particular, if one of the operations fails or expires before having verified the receipt, the function provider shallSHALL return an error (Function execution status indicating 'Failed'). 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 'Executed-WithWarning' with a status code defined in the table below, indicating that theeUICC has been successfully transferred to the new SM-SR, but additional configuration has not completed and may need to be done again. The new SM-SR shallSHALL perform such operations automatically at a later point in time • 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

ID	Source	Chapter	Support	Description	Functional group
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
PF_REQ_5.7.1	[2]	5.7.1	M	ES4A: SetPLMA Description: This function allows the Operator owner of Profiles to grant a PLMA to an M2M SP to perform certain operations, or receive certain notifications, related to a cerain subset of the Profiles owned by the Operator. The SM-SR receiving this request SHALL verify that the mno-id in the PLMA matches the mno-id of the Operator who sends this request. If the request is acceptable, the SM-SR SHALL record the PLMA. The new PLMA overwrites the previous PLMA that might have been granted with the same identifiers. From this point on, any request from the M2M SP on such a Profile, or any notification to the M2M SP related to such a Profile, SHALL be allowed or not based on the new PLMA, as described in sections 5.7.1.1 to 5.7.1.3. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the authorisations have been configured in the SM-SR. • A ‘Function execution status’ with ‘Executed-WithWarning’ with a specific status code as defined in the table here after, indicating that the authorisations have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator. • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section **Error! Reference source not found.** or a specific status code as defined i n the table here after. Input data described in Tables present in section 5.7.1.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
PF_REQ_5.7.2	[2]	5.7.2	M	ES4A: GetPLMA Description: This function allows the Operator owner of Profiles to retrieve the list of PLMA applicable to a certain Profile, or a certain Profile type, or for a certain M2M SP. The SM-SR receiving this request SHALL verify that the requester is the owner of the targeted Profile(s), and return the list of all PLMAs applicable to the specified search criteria. In case the list of PLMAs is very long, the SM-SR MAY truncate the result. The caller can then issue another call to getPLMA with more restrictive criteria. *NOTE The order of the PLMAs returned in the list is implementation-dependant.* This function may return: • A 'Function execution status' with 'Executed- Success', and additional output data providing the PLMAs. • A 'Function execution status' with 'Executed-WIthWarning', to indicate that the result was truncated, plus additional output data providing part of the list of applicable PLMAs. Input data described in Tables present in section 5.7.2.	Platform Management
EUICC_REQ41	VOID				
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

ID	Source	Chapter	Support	Description	Functional group
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
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 verification of the eUICC certificate during 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

ID	Source	Chapter	Support	Description	Functional group
SEC_REQ20	[1]	4.4.4	M	The Operator 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
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	In addition to restrictions to the TLS protocol specified in GP Amendment B [8], the ISD-R and SM-SR SHALL NOT support TLS Session resumption (RFC 4507 or RFC 5077) nor several parallel TLS sessions.	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

ID	Source	Chapter	Support	Description	Functional group
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
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

ID	Source	Chapter	Support	Description	Functional group
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_REQ51_1	[2]	2.4.4.3	M	Intermediate POST response sent by the SM-SR containing no command to execute but instructing to not close the HTTP session: the eUICC SHALL accordingly send a POST on the next URI provided, with no response body: HTTP/1.1 204 CRLF X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF X-Admin-Next-URI: <uri of the next POST> CRLF CRLF	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

ID	Source	Chapter	Support	Description	Functional group
SOAP_REQ_B211_1	[2]	B.2.1	O	• /wsa:From This element is defined in WS-Addressing core specifications [41] as: *This OPTIONAL element (of type wsa:EndpointReferenceType) provides the value for the [source endpoint] property.* In the context of this specification this element is MANDATORY except in the synchronous response and defines the function requester. It SHALL be filled with: • The sender URI. This value is not mapped from any value of the RPS Header, but it should be representative of the sender entity. • A mandatory query parameter “EntityId” containing the <rps3:SenderEntity>/<rps3:EntityId> value. Identifies the direct function caller. • An optional query parameter “EntityName” containing the <rps3:SenderEntity>/<rps3:EntityName> value. Names the direct function caller. • An optional query parameter “UserName” containing the <rps3:SenderName> A mandatory query parameter "MnoId" only for ES3 request messages containing the <rps3:MnoId></rps3:MnoId> value, to identify the Operator which sent the request to the SM-DP via ES2.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
SOAP_REQ_B211_2	[2]	B.2.1	O	• /wsa:To This element is defined in WS-Addressing core specifications [41] as: *This REQUIRED element (of type xs:anyURI) provides the value for the [destination] property.* In the context of this specification this element is MANDATORY and defines the function provider. It SHALL be filled with: • The URL of the web service endpoint to which the message is sent. This value is not mapped from any value of the RPS Header, but it should be representative of the receiving entity. • An optional query parameter “EntityId” containing the <rps3:ReceiverEntity>/<rps3:EntityId> value • A mandatory query parameter “MnoId” only for ES3 response and notification messages containing the <rps3:MnoId></rps3:MnoId> value, to identify the Operator to which the SM-DP SHALL send the response or notification via ES2. The parameter “MnoId” represents: • Either the Operator which is owner of the Profile Or the Operator which is an M2M SP and has a PLMA set to receive this notification.	Platform Management

ID	Source	Chapter	Support	Description	Functional group
SOAP_REQ_B211_4	[2]	B.2.1	O	• /wsa:MessageID This element is defined in WS-Addressing core specifications [41] as: *This OPTIONAL element (whose content is of type xs:anyURI) conveys the [message id] property.* In the context of this specification this element is MANDATORY whatever the MEP. This element SHALL be filled with: • The value set in <rps3:MessageId>. • An optional query parameter “TransactionID” containing the <rps3:TransactionId> value. This query parameter SHALL be present only if <rps3:TransactionId> is present. • An optional query parameter “ContextID” containing the <rps3:ContextId> value. If this optional query parameter is present, it SHALL be included in any new request generated by the function provider entity for another functional provider entity. This identifier MAY be used to provide end-to-end logging management between the different web services. • A mandatory query parameter “MessageDate” containing the <rps3:MessageDate> value A mandatory query parameter "ProfileType" only for notifications messages containing the <rps3:ProfileType></rps3:ProfileType> value.	Platform 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_REQ5_2	[2]	3.2.1		Profile Enabling process Unless Operator2 has set an ONC (Operator Notifications Configuration) to not receive those notifications,the SM-SR SHALL send the “ES4.HandleProfileDisabledNotification” or “ES4.HandleProfileDeletedNotification” (if deletion was triggered by the evaluation of POL1 and POL2) to Operator2, the owner of the Profile that was enabled at the beginning of the procedure. In case Operator2 has no direct connection with the SM-SR (SM-SR SHALL be able to detect such a situation based on its own database), the SM-SR SHALL send this notification to the SM-DP authorised by Operator2 by calling the “ES3.HandleProfileDisabledNotification” or the “ES3.HandleProfileDeletedNotification”. The SM-SR can retrieve the SM-DP identity based on the EIS content. Then the SM-DP, on reception of this notification, SHALL forward it to Operator2 by calling the “ES2.HandleProfileDisabledNotification” or the “ES2.HandleProfileDeletedNotification”.	Procedure Flow
PROC_REQ5_3	[2]	3.2.1		Profile Enabling process The SM-SR SHALL send the "ES4.HandleProfileEnabledNotification" to a M2M SP, if authorised by Operator1 the owner of the Profile that was Disabled at the beginning of the procedure. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator1, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileEnabledNotification”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator1, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileEnabledNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileEnabledNotification”.	Procedure Flow

PROC_REQ5_4	[2]	3.2.1		Profile Enabling process The SM-SR SHALL send the "ES4.HandleProfileDisabledNotification" or "ES4.HandleProfileDeletedNotification" (if deletion was triggered by the evaluation of POL1 and POL2) to a M2M SP, if authorised by Operator2 the owner of the Profile that was Enabled at the beginning of the procedure. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator2, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileDisabledNotification” or “ES4.HandleProfileDeletedNotification”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator2, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileDisabledNotification” or “ES3.HandleProfileDeletedNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileDisabledNotification” or “ES2.HandleProfileDeletedNotification”. NOTE: This M2M SP might be the same M2M SP as for Operator1 or any other M2M SP.	Procedure Flow
PROC_REQ7_1	[2]	3.3.1	M	Profile Enabling via SM-DP process Unless Operator2 has set an ONC to not receive those notifications, The the SM-SR shallSHALL send the “ES4.HandleProfileDisabledNotification” or “ES4.HandleProfileDeletedNotification” (if deletion was triggered by the evaluation of POL1 and POL2) to MNO2Operator2, the owner of the Profile that was enabled at the beginning of the procedure. In case MNO2Operator2 has no direct connection with the SM-SR, the SM-SR shallSHALL apply the same process as described in point (14) of section 3.2.1.	Procedure Flow

PROC_REQ7_2	[2]	3.3.1	M	Profile Enabling via SM-DP process The SM-SR SHALL send the “ES4.HandleProfileEnabledNotification” to a M2M-SP, if authorised by Operator1 the owner of the Profile that was disabled at the beginning of the procedure. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator1, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileEnabledNotification”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator1, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileEnabledNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileEnabledNotification”.	Procedure Flow
PROC_REQ7_3	[2]	3.3.1	M	Profile Enabling via SM-DP process The SM-SR SHALL send the “ES4.HandleProfileDisabledNotification” or “ES4.HandleProfileDeletedNotification” (if deletion was triggered by the evaluation of POL1 and POL2) to a M2M-SP, if authorised by Operator2 the owner of the Profile that was enabled at the beginning of the procedure. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator2, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileDisabledNotification” or “ES4.HandleProfileDeletedNotification” If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator2, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileDisabledNotification” or “ES3.HandleProfileDeletedNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileDisabledNotification” or “ES3.HandleProfileDeletedNotification”. NOTE: This M2M-SP might be the same M2M-SP as for Operaor1 or any other M2M-SP.	Procedure Flow

PROC_REQ9_1	[2]	3.4	M	Profile Disabling process Unless Operator2 has set an ONC to not receive those notifications, the he SM-SR shallSHALL send the “ES4.HandleProfileEnabledNotification” to MNO2Operator2, the owner of Profile with Fall-backFall-Back Attribute set that is now enabled. In case MNO2Operator2 has no direct connection with the SM-SR (SM-SR shallSHALL be able to detect such situation based on its own database), the SM-SR shallSHALL send this notification to the SM-DP authorizedauthorised by MNO2Operator2 by calling the “ES3.HandleProfileEnabledNotification”. The SM-SR can retrieve the SM-DP identity based on the EIS content. Then the SM-DP, on reception of this notification, shallSHALL forward it to MNO2Operator2 by calling the “ES2.HandleProfileEnabledNotification”.	Procedure Flow
PROC_REQ9_2	[2]	3.4	M	Profile Disabling process The SM-SR SHALL send the "ES4.HandleProfileDisabledNotification" to a M2M SP, if authorised by Operator1 the owner of the Profile that was enabled at the beginning of the procedure. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator1, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileDisabledNotification”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator1, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileDisabledNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileDisabledNotification”.	Procedure Flow

PROC_REQ9_3	[2]	3.4	M	Profile Disabling process The SM-SR SHALL send the "ES4.HandleProfileEnabledNotification" to a M2M SP, if authorised by Operator2 the owner of Profile with Fall-Back Attribute set that is now enabled. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator2, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileEnabledNotification”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator2, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileEnabledNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileEnabledNotification”.	Procedure Flow
PROC_REQ10_1	[2]	3.5	M	Profile Disabling via SM-DP process Unless Operator2 has set an ONC to not receive those notifications, theThe SM-SR shallSHALL send the “ES4.HandleProfileEnabledNotification” to MNO2Operator2, the owner of the Profile with Fall-backFall-Back Attribute set that is now enabled.	Procedure Flow
PROC_REQ10_2	[2]	3.5	M	Profile Disabling via SM-DP process The SM-SR SHALL send the "ES4.HandleProfileDisabledNotification" to a M2M SP, if authorised by Operator1 the owner of the Profile that was enabled at the beginning of the procedure. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator1, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileDisabledNotification”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator1, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileDisabledNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileDisabledNotification”.	Procedure Flow

PROC_REQ10_3	[2]	3.5	M	Profile Disabling via SM-DP process The SM-SR SHALL send the "ES4.HandleProfileEnabledNotification" to a M2M SP, if authorised by Operator2 the owner of Profile with Fall-Back Attribute set that is now enabled. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by Operator2, the SM-SR SHALL send this notification to this other Operator by calling the “ES4.HandleProfileEnabledNotification”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by Operator2, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “ES3.HandleProfileEnabledNotification”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “ES2.HandleProfileEnabledNotification”. NOTE: This M2M SP might be the same M2M SP as for Operator1 or any other M2M SP.	Procedure Flow
PROC_REQ11_1	[2]	3.6	M	Profile and ISD-P deletion process The SM-SR SHALL send the "**ES4.HandleProfileDeletedNotification**" to a M2M SP, if authorised by the Operator who owns the Profile, indicating that the profile has been deleted. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by the Operator owning the Profile, the SM-SR SHALL send this notification to this other Operator by calling the “**ES4.HandleProfileDeletedNotification**”. If the M2M SP is another Operator connected through its SM-DP and it is authorised by the Operator owning the Profile, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the “**ES3.HandleProfileDeletedNotification**”. Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the “**ES2.HandleProfileDeletedNotification”.**	Procedure Flow

PROC_REQ12_1	[2]	3.7	M	Profile and ISD-P deletion process via SM-DP The SM-SR SHALL send the "ES4.HandleProfileDeletedNotification" to a M2M SP, if authorised by the Operator owning the Profile, indicating that the profile has been deleted. If the M2M SP is another Operator directly connected to the SM-SR and it is authorised by the Operator owning the Profile, the SM-SR SHALL send this notification to this other Operator by calling the "ES4.HandleProfileDeletedNotification". If the M2M SP is another Operator connected through its SM-DP and it is authorised by the Operator owning the Profile, the SM-SR SHALL send this notification to the SM-DP associated to this other Operator by calling the "ES3.HandleProfileDeletedNotification". Then the SM-DP, on reception of this notification, SHALL forward it to the Operator, acting as the M2M SP, by calling the "ES2.HandleProfileDeletedNotification".	Procedure Flow
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
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
PROC_REQ_3.18	[2]	3.18	M	The Profile Disabling via M2M SP process must be compliant with the Figure 35 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.19	[2]	3.19	M	The Profile and ISD-P Deletion via M2M SP process must be compliant with the Figure 36 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.20.3	[2]	3.20.3	M	The Retrieve PLMA by Operator process must be compliant with the Figure 39 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.20.4	[2]	3.20.4	M	The Retrieve PLMA by Operator via SM-DP process must be compliant with the Figure 40 and with the procedure described in this section.	Procedure Flow
PROC_REQ_3.25_2	[2]	3.25	O	The Emergency Profile Attribute management process must be compliant with the Figure 325-B and with the procedure for "case 2" (change of Emergency Profile) described in this section.	Procedure Flow

PROC_REQ_3.26_2	[2]	3.26	O	The Emergency Profile Attribute management via M2M SP process must be compliant with the Figure 326-B and with the procedure for “case 2” (change of Emergency Profile) described in this section.	Procedure Flow
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 Operator and the M2M SP 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
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	Security

				such as MPLS or GRE are not considered as dedicated links; a solution such as an authenticated and secured VPN is considered as dedicated.	
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 M2M Device SHALL NOT be able to access nor modify sensitive Profile data, i.e. credentials, management commands, Policy Rules, authentication algorithm parameters.	Security
PROC_REQ_3.28_1	[2]	3.28	M	The Fall-Back Attribute via SM-DP process must be compliant with the Figure 328 and with the procedure described in this section.	Procedure Flow
PF_REQ9_1	[2]	4.1.1.7	M	If the currently Enabled profile is the Profile with the Fall-Back Attribute set, and has been Enabled by the activation of the Fall-Back Mechanism, and the previously Enabled Profile has either of the POL1 rules “Disable not allowed” or “Profile deletion is mandatory when its state is changed to Disabled” set, then the eUICC SHALL prevent the execution of the function “Set Fall-Back Attribute”. Processing State Returned in the Response Message: As defined in GlobalPlatform Card Specification **Error! Reference source not found.** section 11.11.3.2, with the following addition: ’69 E1’: POL1 of the Profile Disabled by the activation of the Fall-Back Mechanism prevents this action.	Platform Management
PF_REQ_5.3.13	[2]	5.3.13	M	ES2: SetPLMA Description:	Platform Management

				This function allows the Operator owning Profiles to grant PLMAs to an M2M SP to perform certain operations, or receive certain notifications, related to Profiles, identified by a Profile Type. The SM-DP receiving this request SHALL forward it to the SM-SR indicated by the Operator, according to procedure “Set Profile Lifecycle Management Authorisations via SM-DP” described in section 3.20.2 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the authorisations have been configured in the SM-SR. • A ‘Function execution status’ with ‘Executed-WithWarning’ with a specific status code as defined in the table below, indicating that the authorisations have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator. • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 or a specific status code as defined in the table below. Input data described in Tables present in section 5.3.13.	
PF_REQ_5.3.14	[2]	5.3.14	M	ES2: GetPLMA Description: This function allows the Operator owner of Profiles to retrieve a list of PLMAs applicable to a certain Profile, or a certain Profile Type, or for a certain M2M SP. The same function can also be used by the Operator playing the role of an M2M SP, to retrieve the list of PLMAs granted to this Operator, and applicable to a certain Profile, or a certain Profile Type, owned by another Operator. The SM-DP receiving this request SHALL forward it to the SM-SR indicated by the Operator, according to procedure “Retrieve Profile Lifecycle Management Authorisations via SM-DP” described in section 3.20.4 of this specification. This function may return:	Platform Management

				• A ‘Function execution status’ with ‘Executed- Success’, and additional output data providing the PLMAs. • A ‘Function execution status’ with ‘Executed-WIthWarning’, to indicate that the result was truncated, plus additional output data providing part of the list of applicable PLMAs. • A 'Function execution status' indicating 'Failed' with a status code as defined in section 5.1.6.4 or a specific status code as defined in the table below. Input data described in Tables present in section 5.3.14.	
PF_REQ_5.3.17	[2]	5.3.17	M	ES2: HandleSetPLMANotification Description: This function SHALL be called to notify an Operator (acting as an M2M SP from the point of view of another Operator) that a PLMA concerning this M2M SP has been set or updated. This notification also conveys the date and time specifying when the operation has been done. In case of multiply handlers are served the SM-SR SHOULD ensure ‘completionTimestamp’ to be equal for every message. Input data described in Tables present in section 5.3.17.	Platform Management
PF_REQ_5.3.18	[2]	5.3.18	M	ES2: SetONC Description: This function allows the Operator to configure for which of its own Profiles, associated with a Profile Type, it wants to receive which kind of status change notifications; whatever the origin of the status change is. The SM-DP receiving this request SHALL forward it to the SM-SR indicated by the Operator, according to procedure “Set Operator Notifications Configuration via SM-DP” described in section 3.21.2 of this specification.	Platform Management

				This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the notifications configuration has been configured in the SM-SR. • A ‘Function execution status’ with ‘Executed-WithWarning’ with a specific status code as defined in table 5318-C, indicating that the authorisations have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator. • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 or a specific status code as defined in the table here after Input data described in Tables present in section 5.3.18.	
PF_REQ_5.3.19	[2]	5.3.19	M	ES2: GetONC Description: This function allows the Operator to retrieve a list of status change notifications it does not want to receive for its own Profiles, associated with a Profile Type. The SM-DP receiving this request SHALL forward it to the SM-SR indicated by the Operator, according to procedure “Retrieve Operator Notifications Configuration via SM-DP” described in section 3.21.4 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed- Success’, and additional output data providing the configured ONC. • A ‘Function execution status’ with ‘Executed-WithWarning’ with a specific status code as defined in table 5319-C, indicating that the authorisations have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator, and additional output data providing the configured ONC.	Platform Management

				• A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 or a specific status code as defined in the table here after. Input data described in Tables present in section 5.3.18.	
PF_REQ_5.3.20	[2]	5.3.20	O	ES2: SetEmergencyProfileAttribute Description: This function allows the Operator owner of the Profile to request an SM-DP to set the Emergency Profile Attribute on a Profile in a specified eUICC, eUICC being identified by its EID. The SM-DP receiving this request SHALL process it according to the “Emergency Profile Attribute Management” procedure described in the section 3.25 of this specification (option b: via SM-DP). This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the Emergency Profile Attribute has been set on the targeted Profile. • 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 or a specific status code as defined in the table here after. Input data described in Tables present in section 5.3.20.	Platform Management
PF_REQ_5.3.23	[2]	5.3.23	M	ES2: SetFallBackAttribute Description: This function allows the Operator owner of the Profile to request an SM-DP to set the Fall-Back Attribute on a Profile in a specified eUICC, eUICC being identified by its EID.	Platform Management

				The SM-DP receiving this request SHALL process it according to the “Fall-Back Attribute Management” procedure described in sections **Error! Reference source not found.** and **Error! Reference source not found.**. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the Fall-Back Attribute has been set on the targeted Profile. • A ‘Function execution status’ with ‘Expired’ with a status code as defined in section **Error! Reference source not found.**. • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section **Error! Reference source not found.** or a s pecific status code as defined in the table here after. Input data described in Tables present in section 5.3.23.	
PF_REQ_5.3.24	[2]	5.3.24	M	ES2: HandleProfileFallBackAttributeSetNotification Description: This function SHALL be called to notify that the Fall-Back Attribute has been set on the Profile identified by its ICCID, on the eUICC identified by its EID. This notification also conveys the date and time specifying when the operation has been done. What is performed by the Operator receiving this notification is out of scope of this specification. Input data described in Tables present in section 5.3.24.	Platform Management
PF_REQ_5.3.25	[2]	5.3.25	M	ES2: HandleProfileFallBackAttributeUnsetNotification Description: This function SHALL be called to notify that the Fall-Back Attribute has been unset on the Profile identified by its ICCID, on the eUICC identified by its EID. This notification also conveys the date and time specifying when the operation has been done. What is performed by the Operator receiving this notification is out of scope of this specification.	Platform Management

				Input data described in Tables present in section 5.3.25.	
PF_REQ_5.4.17	[2]	5.4.17	M	ES3: GetPLMA Description: This function allows the SM-DP to retrieve, on behalf of an Operator owning Profiles, a list of PLMAs applicable to a certain Profile, or a certain Profile Type, or for a certain M2M SP. The same function can also be used on behalf of an Operator playing the role of an M2M SP, to retrieve the list of PLMAs granted to this Operator, and applicable to a certain Profile, or a certain Profile Type, owned by another Operator. The SM-SR SHALL verify that the request is • Either sent on behalf of an Operator owning the targeted Profile or • Sent on behalf of an Operator that is not the owner of the targeted Profile, but the Operator owning the targeted Profile has granted a PLMA allowing at least one operation for the target Profile or Profile Type to the Operator requesting the operation. If this verification fails, the SM-SR SHALL terminate the request and return a response with the 'Function execution status' indicating 'Failed', and no PLMA. Otherwise, the SM-SR SHALL return the complete list of all PLMAs applicable to the specified search criterion; if the search criterion is on a specific Profile or Profile Type, this includes even PLMAs that are granted to an M2M SP that is not the Operator on behalf of which the SM-DP sent this request. In case the list of PLMAs is very long, the SM-SR MAY truncate the result. The caller can then issue another call to getPLMA with more restrictive criteria. NOTE The order of the PLMAs returned in the truncated list is implementation-dependant. This function may return:	Platform Management

				• A ‘Function execution status’ with ‘Executed- Success’, and additional output data providing the PLMAs. • A ‘Function execution status’ with ‘Executed-WIthWarning’, to indicate that the result was truncated, plus additional output data providing part of the list of applicable PLMAs. • A ‘Function execution status’ indicating ‘Failed’ if the requester was not allowed to request this information. Input data described in Tables present in section 5.4.17.	
PF_REQ_5.4.22	[2]	5.4.22	M	ES3: GetONC Description: This function allows the Operator to retrieve a list of status change notifications it wants not to receive for its own Profiles, associated with a Profile Type. The SM-SR receiving this request SHALL verify that the mno-id in the ONC matches the mno-id of the Operator on behalf of which the SM-DP declares to send this request. If the request is acceptable, the SM-SR SHALL return the ONC including the list of notifications the Operator does not want to receive, applicable to the specified search criterion. This function may return: • A ‘Function execution status’ with ‘Executed-Success’, and additional output data providing the configured ONC. • A ‘Function execution status’ with ‘Executed-WithWarning’ with a specific status code as defined in the table below, indicating that the notifications have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator, and additional output data providing the configured ONC. • A ‘Function execution status’ indicating ‘Failed’ with a status code as defined in section 5.1.6.4 or a specific status code as defined in the table here after.	Platform Management

				Input data described in Tables present in section 5.4.22.	
PF_REQ_5.4.26	[2]	5.4.26	M	ES3: SetFallBackAttribute Description: This function allows the SM-DP authorised by the Operator to request the setting of the Fall-Back Attribute on the targeted Profile to the SM-SR in charge of the management of the targeted eUICC; eUICC being identified by its EID. The SM-DP SHALL indicate on behalf of which Operator it is requesting this operation. The SM-SR SHALL verify that the request is • Either sent on behalf of an Operator owning the targeted Profile or • Sent on behalf of an Operator that is not the owner of the targeted Profile, but the Operator owning the targeted Profile has granted a PLMA allowing the operation “SetFallBackAttribute” to the Operator requesting the operation. In both cases, the SM-SR SHALL verify that the Operator owning the Profile which currently has the Fall-Back Attribute set has granted, to the Operator requesting the operation, a PLMA authorising the operation “UnsetFallBackAttribute”, applicable for the Profile that currently has the Fall-Back Attribute set. The SM-SR MAY provide additional verifications. The SM-SR receiving this request SHALL process it according to “Fall-Back Attribute Management via SM-DP” procedure described in the section 3.28 of this specification. This function may return: • A ‘Function execution status’ with ‘Executed- Success’ indicating that the Fall-Back Attribute has been set on the targeted Profile. • 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 of a specific status code as defined in the table	Platform Management

				here after. Input data described in Tables present in section 5.4.26.	
PF_REQ_5.4.27	[2]	5.4.27	M	ES3: HandleProfileFallBackAttributeSetNotification Description: This function SHALL be called to notify that the Fall-Back Attribute has been set on the Profile identified by its ICCID on the eUICC identified by its EID. The SM-SR SHALL send this notification to all SM-DP servers that match one or the other of the following conditions: • The SM-DP can relay the notification to the Operator that owns the Profile, and the Operator has opted to receive such notifications (see section 3.21). • The SM-DP can relay the notification to another Operator, and the Operator owner of the Profile has granted the other Operator with a PLMA authorising this Operation "HandleProfileFallBackAttributeSetNotification". ICCID may be not enough to identify right address of recipient, SM-SR should map it internally to Operator notification endpoint. This notification also conveys the date and time specifying when the operation has been done. In case multiple handlers are served, the SM-SR SHOULD ensure completionTimestamp to be equal for every message. What is performed by the Operator receiving this notification is out of scope of this specification. Input data described in Tables present in section 5.4.27.	Platform Management
PF_REQ_5.7.3	[2]	5.7.3	M	ES4A: SetONC Description:	Platform Management

				This function allows the Operator to configure for which of its own Profiles, associated with a Profile Type, it wants to receive which kind of status change notifications; whatever the origin of the status change is. The SM-SR receiving this request SHALL verify that the mno-id of the function caller matches with the one in the ONC. If the request is acceptable, the SM-SR SHALL record the ONC. The new ONC overwrites the previous ONC that might have been granted with the same identifiers. From this point on, any status change notification, irrespective of the cause and related to a Profile matching these identifiers, SHALL be sent or not based on the new ONC. This function may return: • A 'Function execution status' with 'Executed-Success' indicating that the notifications have been configured in the SM-SR. • A 'Function execution status' with 'Executed-WithWarning' with a specific status code as defined in the table here after, indicating that the notifications have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator. • A 'Function execution status' indicating 'Failed' with a status code as defined in section **Error! Reference source not found.** or a s pecific status code as defined in the table here after. NOTE: If no Operator Notification Configuration has yet been set in the SM-SR for a given Profile Type, then the Operator will receive all notifications for status changes for its own Profiles, associated with this Profile Type, see also section 3.21 for details. Input data described in Tables present in section 5.7.3.	
PF_REQ_5.7.4	[2]	5.7.4	M	ES4A: GetONC Description: This function allows the Operator to retrieve a list of status change notifications it does not want to receive for its own Profiles, associated with a Profile Type.	Platform Management

				The SM-SR receiving this request SHALL verify that the mno-id of the function caller matches with the one in the ONC. If the request is acceptable, the SM-SR SHALLreturn the ONC including the list of requested notifications applicable to the specified search criterion. This function may return: • A 'Function execution status' with 'Executed-Success', and additional output data providing the configured ONC. • A 'Function execution status' with 'Executed-WithWarning' with a specific status code as defined in the table below, indicating that the notifications have been configured in the SM-SR but that some side-effects of this configuration may require the attention of the Operator, and additional output data providing the configured ONC. • A 'Function execution status' indicating 'Failed' with a status code as defined in section **Error! Reference source not found.** or a s pecific status code as defined in the table here after. Input data described in Tables present in section 5.7.4.	
SOAP_REQ_B211_3	[2]	B.2.1	O	• /wsa:ReplyTo This element is defined in WS-Addressing core specifications [41] as: *This OPTIONAL element (of type wsa:EndpointReferenceType) provides the value for the [reply endpoint] property. If this element is NOT present, then the value of the [address] property of the [reply endpoint] EPR is "http://www.w3.org/2005/08/addressing/anonymous".* In the context of this specification this element is OPTIONAL. This element SHALL be present only when: • MEP follows Asynchronous Request-Response with callback and • When Message sender wants the response to be sent to a specific endpoint If missing, the response SHALL be sent to (in the preferred order):	Platform Management

				• a well-known endpoint mutually agreed between message sender and message receiver • or to the message originating endpoint. If present, the /wsa:ReplyTo SHALL be filled with: • The value set in <rps3:ResponseEndpoint> An optional query parameter “EntityId” containing the <rps3:ReceiverEntity>/<rps3:EntityId> value	

Table 26: Out of Scope Requirements

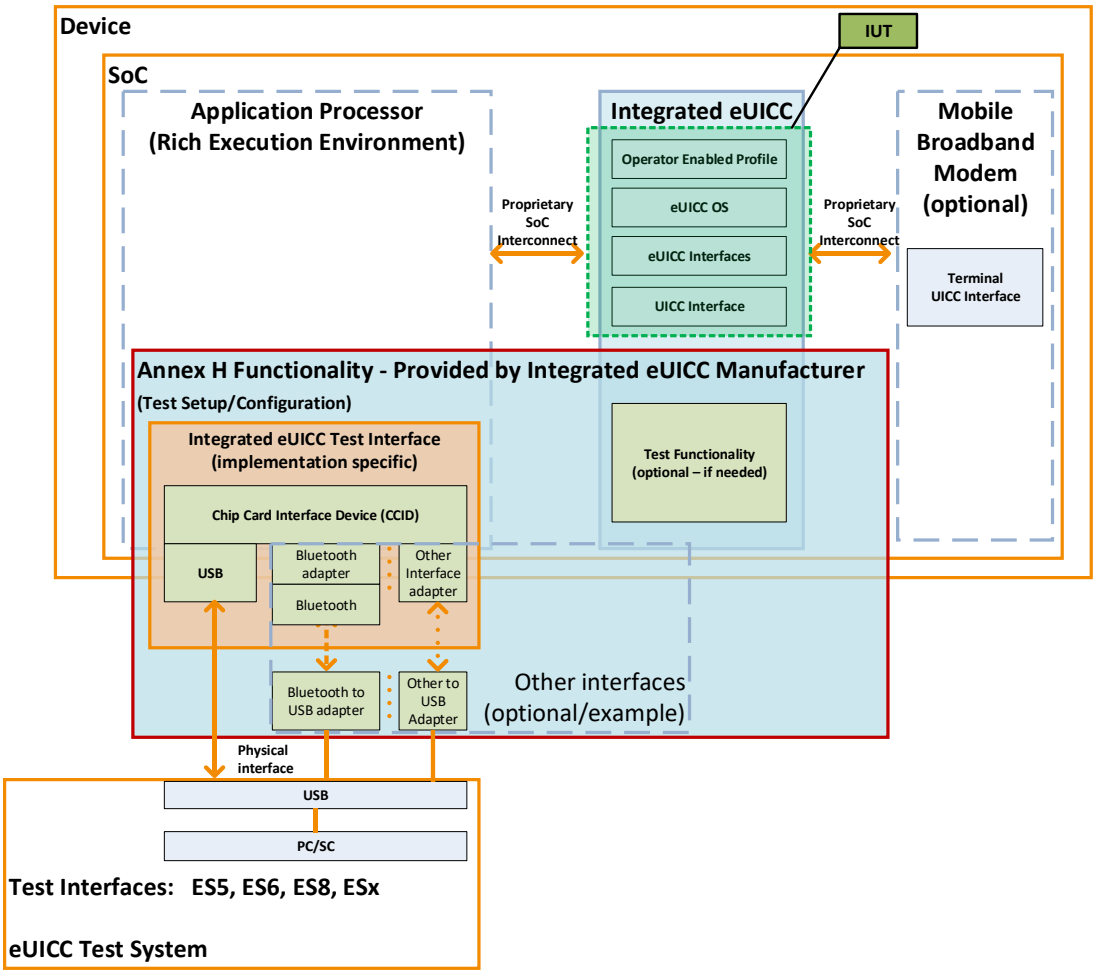
Annex K Integrated eUICC Testing (Normative)

K.1 Overview (Informative)

An Integrated eUICC hardware resides in an SoC along with other subsystems such as general processing and mobile broadband modem, all connected through a proprietary SoC interconnect channel. Alternatively, an Integrated eUICC may communicate with a mobile broadband modem external to the SoC via an external interface, which may be proprietary or based on a standard not associated with UICC. As such, Integrated eUICC may not include a physical UICC-Terminal interface.

In order to test the functionality and compliancy of an Integrated eUICC, hardware and OS, Integrated eUICC manufacturers need to provide and support a test interface to which testing equipment can be connected to. Having a standardized testing interface, will increase interoperable and reusability between different manufacturer of Integrated eUICC and test equipment.

For Integrated eUICC with USB CCID test interface[24], the properties are described in this annex. In cases where a USB interface is not available in a device containing an Integrated eUICC, an adapter to USB CCID needs to be provided, e.g. Bluetooth to USB CCID. The functionality needed to provide and support the test interface, shall be considered part of the test environment and not the IUT.



Integrated eUICC Test Interface

Note: The mechanism providing USB CCID to the eUICC Test System, and described in this Annex, is implementation specific. As such, it may be implemented in the SoC, on-Device, off-Device or any combination thereof.

K.2 Integrated eUICC test requirements

An Integrated eUICC manufacturer shall provide a USB CCID test interface implementing the functionality specified in H.3.

The test interface shall maintain the integrity and order of the data between the Integrated eUICC and the test system.

The Integrated eUICC manufacturer shall ensure that during testing no other clients or SoC subsystems interfere with the testing.

The Integrated eUICC may use any physical or logical interface between the Integrated eUICC and the test system, as long as a USB CCID is provided to the test system and the channel is reliable (i.e. maintain integrity and order).

K.3 USB CCID test interface

The Integrated eUICC USB CCID test interface shall operate in a card reader mode.

The Integrated eUICC USB CCID test interface shall support the following [24] section 6 messages:

- [24] section 6.1 Messages:
  - PC_to_RDR_IccPowerOn
  - PC_to_RDR_IccPowerOff
  - PC_to_RDR_GetSlotStatus
  - PC_to_RDR_Escape
  - PC_to_RDR_XfrBlock
  - PC_to_RDR_T0APDU
  - PC_to_RDR_Secure
  - PC_to_RDR_Abort
- [24] section 6.2 Messages:
  - RDR_to_PC_SlotStatus
  - RDR_to_PC_Escape
  - RDR_to_PC_DataBlock

Note: For test systems using wincard.h/PCSC lite APIs to connect to the Integrated eUICC USB CCID test interface, the following APIs are expected to be used:

- SCardEstablishContext
- SCardListReaders[A|W]
- SCardConnect[A|W]
- SCardControl
- SCardTransmit
- SCardDisconnect
- SCardStatus[A|W]
- SCardReleaseContext
- SCardReconnect
- SCardBeginTransaction
- SCardEndTransaction
- SCardGetStatusChange
- SCardFreeMemory
- SCardGetAttrib

7 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, 15ESIMWI314_04, 15ESIMWI314_05,	Sébastien Kuras, FIME

Version	Date	Brief description of change	Editor / Company
		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
3.2	June 2017	16ESIMWI323_Doc001, 16ESIMWI324_Doc002, 16ESIMWI324_Doc003, 16ESIMWI325_Doc004_r02,	Thomas Rhodes, Simulity

Version	Date	Brief description of change	Editor / Company
		16ESIMWI325_Doc003, 17ESIMWI325_Doc006_r03, 17ESIMWI327_Doc004r01, 17ESIMWI327_Doc005, 17ESIMWI327_Doc006r1, 17ESIMWI3281_Doc_002, 17ESIMWI3281_Doc_003r01, 17ESIMWI3281_Doc_004R3, 17ESIMWI3281_Doc_005R1, 17ESIMWI3281_Doc_006R1, 17ESIMWI3281_Doc_007R2, 17ESIMWI3281_Doc_008R1, 17ESIMWI3281_Doc_009R1, 17ESIMWI3281_Doc_010R3, 17ESIMWI3282_Doc_002r1, 17ESIMWI3282_Doc_003r1, 17ESIMWI3283_Doc_002R1, 17ESIMWI3284_Doc_002r1, 17ESIMWI3284_Doc_003r1, 17ESIMWI3284_Doc_004r1, 17ESIMWI3284_Doc_005r2, 17ESIMWI3284_Doc_007r1, 17ESIMWI3284_Doc_008r1, 17ESIMWI3284_Doc_009r1, 17ESIMWI3284_Doc_010r1, 17ESIMWI329_Doc_009r1, 17ESIMWI329_Doc_011r1, 17ESIMWI330_Doc_004r1, 17ESIMWI330_Doc_006r3, 17ESIMWI330_Doc_005r1, 17ESIMWI330_Doc_012r1, 17ESIMWI330_Doc_007r1, 17ESIMWI330_Doc_009r1, 17ESIMWI330_Doc_003r4, 17ESIMWI330_Doc_008r3, 17ESIMWI330_Doc_010r1, 17ESIMWI330_Doc_013r3, 17ESIMWI331_Doc_004R2, 17ESIMWI331_Doc_005r1, 17ESIMWI331_Doc_006r2, 17ESIMWI331_Doc_007r1, 17ESIMWI332_Doc_007r1, 17ESIMWI332_Doc_008r1, 17ESIMWI333_Doc_004r3	

Version	Date	Brief description of change	Editor / Company
3.3	July 2018	18ESIMWI345_Doc_005r1, 18ESIMWI345_Doc_006r1, 18ESIMWI345_Doc_007r1, 18ESIMWI346_Doc_003, 18ESIMWI346_Doc_005, 18ESIMWI346_Doc_006, 18ESIMWI346_Doc_007, 18ESIMWI346_Doc_011, 18ESIMWI346_Doc_012, 18ESIMWI346_Doc_013, 18ESIMWI346_Doc_004 18ESIMWI3471_Doc_003r1 18ESIMWI3471_Doc_004r1, 18ESIMWI348_Doc_003r1	Sébastien Kuras, FIME
4.0	August 2018	17ESIMWI338_Doc_003, 18ESIMWI342_Doc_003, 18ESIMWI342_Doc_004, 18ESIMWI344_Doc_003r1, 18ESIMWI348_Doc_004r1, 18ESIMWI348_Doc_005r1, 18ESIMWI348_Doc_011r1 18ESIMWI348_Doc_006r3, 18ESIMWI352_Doc_003, 18ESIMWI353_Doc_003r2, 18ESIMWI354_Doc_003r1, 18ESIMWI355_Doc_003, 18ESIMWI355_Doc_005r1, 18ESIMWI358_Doc_003r1, 18ESIMIWI356_Doc_003R04, 18ESIMWI357_Doc_003r2, 18ESIMWI359_Doc_005r2, 19ESIMWI360_Doc_003r1, 19ESIMWI360_Doc_004r1, 19ESIMWI361_Doc_003R01, 19ESIMWI362_Doc_003R01, 19ESIMWI362_Doc_004R01, 19ESIMWI362_Doc_006R01, 18ESIMWI355_Doc_004R05, 19ESIMWI362_Doc_007R04, 19ESIMWI363_Doc_006R01, 19ESIMWI363_Doc_007R01,	María José Carreño, VALID

Version	Date	Brief description of change	Editor / Company
		19ESIMWI364_Doc_003R02, 19ESIMWI364_Doc_004R04, 19ESIMWI365_Doc_002R01, 19ESIMWI365_Doc_002R01, 19ESIMWI365_Doc_003R01, 19ESIMWI362_Doc_005R07, 19ESIMWI363_Doc_003R01, 19ESIMWI363_Doc_004R01, 19ESIMWI363_Doc_005R03, 19ESIMWI366_Doc_003R01, 19ESIMWI366_Doc_002R03, 19ESIMWI367_Doc_002R01, 19ESIMWI367_Doc_005R01, 19ESIMWI367Doc_004R01, 19ESIMWI368_Doc_002r2, 19ESIMWI368_Doc_003r02, 19ESIMWI368_Doc_004R01, 19ESIMWI368_Doc_005R01	
4.1	January 2020	CR4001R01, CR4102R06, CR4104R01, CR4105R01, CR4106R00, CR4107r00, CR4109R00, CR4110R00, CR4112r00, CR4113r01, CR4114r01	María José Carreño, VALID
4.2	May 2020	CR4103R00, CR4111R01, CR4201R04	María José Carreño, VALID
4.2.1	20 October 2020	CR4211R00 CR4212R00	Yolanda Sanz, GSMA

7.1 Document Owner

Type	Description
Document Owner	eSIM Group
Editor / Company	María José Carreño, VALID

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.