RSP Technical Specification
Version 1.1
09 June 2016

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

Security Classification: Non-confidential

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

Copyright Notice (Test)

Disclaimer

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

Antitrust Notice

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

Table of Contents

1 Introduction

1.1 Overview

This document provides a technical description of the GSMA's 'Remote Sim Provisioning (RSP) Architecture for consumer Devices'.

1.2 Scope

This specification provides a technical description of:

- The eUICC Architecture;
- The interfaces used within the Remote SIM Provisioning Architecture; and
- The security functions used within the Remote SIM Provisioning Architecture.

1.3 Document Purpose

This document defines a technical solution for the remote provisioning and management of the Embedded UICC (eUICC) in consumer Devices as defined in RSP Architecture SGP.21 [4]. The adoption of this technical solution will provide the basis for global interoperability between different Operator deployment scenarios, for example network equipment (e.g. Subscription Manager Data Preparation (SM-DP+)) and various eUICC platforms.

1.4 Intended Audience

Technical experts working for Operators, SIM solution providers, consumer Device vendors, standards organisations, network infrastructure vendors, Service Providers and other industry bodies, etc.

1.5 Definition of Terms

Term	Description
Certificate Authority	A Certificate Authority is an entity that issues digital certificates.
Certificate Issuer	An Entity that is Authorised to Issue digital certificates.
Companion Device	A Device that relies on the capabilities of a Primary Device for the purpose of Remote SIM Provisioning.
Device	User equipment used in conjunction with an eUICC to connect to a mobile network. E.g. a tablet, wearable, smartphone or handset.
Disabled (Profile)	The state of a Profile where all files and applications (e.g. NAA) present in the Profile are not selectable over the eUICC-Terminal interface.
Embedded UICC	A UICC which is not easily accessible or replaceable, is not intended to be removed or replaced in the Device, and enables the secure changing of Profiles.
Enabled (Profile)	The state of a Profile when its files and/or applications (e.g., NAA) are selectable over the UICC-Terminal interface.
eUICC Certificate	A certificate issued by the EUM for a specific eUICC. This Certificate can be verified using the EUM Certificate.

eUICC Manufacturer	Supplier of the eUICCs and resident software (e.g. firmware and operating system).
EUM Certificate	A certificate issued to a GSMA accredited EUM which can be used to verify eUICC Certificates. This Certificate can be verified using the Root Certificate.
Integrated Circuit Card ID	Unique number to identify a Profile in an eUICC. NOTE: the ICCID throughout this specification is used to identify the Profile.
International Mobile Subscriber Identity	Unique identifier owned and issued by Mobile operators to (U)SIM applications to enable Devices to attach to a network and use services as defined in 3GPP TS 23.003 section 2.2.
Issuer Security Domain	A security domain on the UICC as defined by GlobalPlatform Card Specification [8].
Local Profile Management	Local Profile Management are operations that are locally initiated on the End User (ESeu) interface.
Local Profile Management Operation	Local Profile Management Operations include enable Profile, disable Profile, delete Profile, query Profile Metadata, eUICC Memory Reset and Add/Update Profile Nickname.
MatchingID	Equivalent to “Activation Code Token” as defined is SGP.21 [4]
Mobile Network Operator	An entity providing access capability and communication services to its End User through a mobile network infrastructure.
Mobile Network Operator Security Domain (MNO-SD)	Part of the Profile, owned by the Operator, providing the Secured Channel to the Operator’s Over The Air (OTA) Platform. It is used to manage the content of a Profile once the Profile is enabled.
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 Application Identifier (AID).
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.
Primary Device	A Device that can be used to provide some capabilities to a Companion Device for the purpose of Remote SIM Provisioning.
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 MNO-SD.
Profile	Combination of a file structure, data and applications to be provisioned onto, or present on, an eUICC and which allows, when enabled, the access to a specific mobile network infrastructure.

Profile Management	A set of functions related to the downloading, installation and content update of a Profile in a dedicated ISD-P on the eUICC. Download and installation are protected by Profile Management Credentials shared between the SM-DP+ and the ISD-P.
Profile Management Credentials	Data required within an eUICC so that a Profile downloaded from an external entity can be decrypted and installed on the eUICC.
Profile Nickname	Alternative name of the Profile set by the End User.
Profile Type	Operator specific defined type of Profile. This is equivalent to the “Profile Description ID” as described in Annex B of SGP.21 [4]
Roles	Roles are representing a logical grouping of functions.
Service Provider	The organization through which the End User obtains PLMN telecommunication services. This is usually the network operator or possibly a separate body.
Subscription	Describes the commercial relationship between the End User and the Service Provider.
Subscription Manager Data Preparation+ (SM-DP+)	This role prepares Profile Packages, secures them with a Profile protection key, stores Profile protection keys in a secure manner and the Protected Profile Packages in a Profile Package repository, and allocates the Protected Profile Packages to specified EIDs. The SM-DP+ binds Protected Profile Packages to the respective EID and securely downloads these Bound Profile Packages to the LPA of the respective eUICC.
User Intent	Describes the direct, real time acquisition and validation of the manual End User instruction by the LUI to trigger locally a Profile download or Profile Management operation. As defined in SGP.21 [4].

1.6 Abbreviations

Abbreviation	Description
AID	Application Identifier
BPP	Bound Profile Package
CA	Certificate Authority
CASD	Controlling Authority Security Domain
CERT.CA.SIGN	Certificate of the CA for verifying CERT.DP.TLS
CERT.CI.ECDSA	Certificate of the CI for its Public ECDSA Key
CERT.DP.ECDSA	Certificate of the SM-DP+ for its Public ECDSA key
CERT.EUICC.ECDSA	Certificate of the eUICC for its Public ECDSA key
CERT.EUM.ECDSA	Certificate of the EUM for its Public ECDSA key
CERT.DP.TLS	Certificate of the SM-DP+ for securing TLS
CI	Certificate Issuer
CMAC	Cipher-based MAC

CRT	Control Reference Template
ECASD	eUICC Controlling Authority Security Domain
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve cryptography Digital Signature Algorithm
ECKA	Elliptic Curve cryptography Key Agreement algorithm
EID	eUICC-ID
ETSI	European Telecommunications Standards Institute
eUICC	Embedded Universal Integrated Circuit Card
EUM	eUICC Manufacturer
FFS	For Further Study
FQDN	Fully Qualified Domain Name
GP	GlobalPlatform
GSMA	GSM Association
HLR	Home Location Register
ICCID	Integrated Circuit Card ID
IMSI	International Mobile Subscriber Identity
ISD	Issuer Security Domain
ISD-P	Issuer Security Domain Profile
ISD-R	Issuer Security Domain Root
ISO	International Standards Organisation
ITU	International Telecommunications Union
LPA	Local Profile Assistant
LPD	Local Profile Download
LTE	Long Term Evolution
LUI	Local User Interface
MAC	Message Authentication Code
MEP	Message Exchange Pattern
MNO	Mobile Network Operator
MOC	Mandatory, Optional or Conditional
NAA	Network Access Application
OTA	Over The Air
otPK.DP.ECKA	One-time Public Key of the SM-DP+ for ECKA
otPK.EUICC.ECKA	One-time Public Key of the eUICC for ECKA
otSK.DP.ECKA	One-time Private Key of the SM-DP+ for ECKA
otSK.EUICC.ECKA	One-time Private Key of the eUICC for ECKA
PE	Profile Element
PIX	Proprietary application Identifier eXtension
PK.CI.ECDSA	Public Key of the CI, part of the CERT.CI.ECDSA
PK.DP.ECDSA	Public Key of the SM-DP+ part of the CERT.DP.ECDSA.

PK.EUICC.ECDSA	Public Key of the eUICC, part of the CERT.EUICC.ECDSA
PK.EUM.ECDSA	Public Key of the EUM, part of the CERT.EUM.ECDSA
POS	Point Of Sale
PPK-ENC	Profile Protection Key for message encryption/decryption
PPK-MAC	Profile Protection Key for message MAC generation/verification
PPK-RMAC	Profile Protection Key for response MAC generation/verification
PPP	Protected Profile Package
RFU	Reserved for Future Use
SBPP	Segmented Bound Profile Package
SCP	Secure Channel Protocol
SD	Security Domain
S-ENC	Session key for message encryption/decryption
S-MAC	Session Key for message MAC generation/verification
S-RMAC	Session Key for response MAC generation/verification
ShS	Shared Secret
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.EUICC.ECDSA	Private key of the eUICC for creating signatures
SK.EUM.ECDSA	Private key of the EUM for creating signatures
SK.CI.ECASD	Private key of the CI for signing certificates
SK.DP.TLS	Private key of the SM-DP+ for securing TLS connection
SM-DP+	Subscription Manager Data Preparation (Enhanced compared to the SM-DP in SGP.02 [2])
SVN	SGP.22 Specification Version Number (referred to as ‘eSVN’ in SGP.21 [4].
TAC	Type Allocation Code
TAR	Toolkit Application Reference
TLS	Transport Layer Security
UPP	Unprotected Profile Package
URI	Uniform Resource Identifier
URL	Uniform Resource locator
USIM	Universal Subscriber Identity Module
W3C	World Wide Web Consortium

1.7 References

Ref	Document Number	Title
[1]	TF_Req	GSMA 'Embedded SIM Task Force Requirements and Use Cases' Version 1.0
[2]	SGP.02	GSMA Remote Provisioning of Embedded UICC Technical specification V3.0
[3]	Void	Void
[4]	SGP.21	RSP Architecture V1.0
[5]	SIMalliance	SIMalliance eUICC Profile Package: Interoperable Format Technical Specification V2.0
[6]	ETSI TS 102 221	Smart Cards; UICC-Terminal interface
[7]	Void	Void
[8]	GPC_SPE_034	GlobalPlatform Card Specification v.2.2.1
[9]	GPC_SPE_025	GlobalPlatform Card Specification v.2.2 Amendment A:
[10]	GPC_SPE_025	GlobalPlatform Card Specification v.2.2 Amendment C: Contactless Services v1.1.1
[11]	GPC_SPE_014	GlobalPlatform Card Specification v.2.2 Amendment D: Secure Channel Protocol '03' v1.1.1.
[12]	GPC_SPE_042	GlobalPlatform Card Specification v.2.2 Amendment E: Security Upgrade for Card Content Management v1.0
[13]	GPC_SPE_093	GlobalPlatform Card Specification v.2.2 Amendment F: Secure Channel Protocol '11'
[14]	ISO/IEC 7816-4:2013	Identification cards – Integrated circuit cards - Part 4: Organization, security and commands for interchange
[15]	ISO/IEC 18004:2015	Information technology -- Automatic identification and data capture techniques -- QR Code bar code symbology specification
[16]	RFC 5246	The TLS Protocol – Version 1.2
[17]	RFC 5280	Internet X.509 PKI Certificate and CRL Profile
[18]	RFC 5639	Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation
[19]	RFC 793	Transmission Control Protocol, DARPA Internet Program, Protocol specification, Sept 1981.
[20]	ANSSI ECC FRP256V1	Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français. JORF n°0241 du 16 octobre 2011 page 17533. texte n° 30. 2011
[21]	ITU E.118	The International Telecommunication Union charge card
[22]	GSMA Security Principles Related to Handset Theft	GSMA Doc Reference: Security Principles Related to Handset Theft 3.0.0 EICTA CCIG Doc. Reference: EICTA Doc: 04cc100
[23]	GSMA SAS-SM	GSMA SAS Standard for Subscription Manager Roles Version 2.0 - 13 May 2015

[24]	ITU-T X.520	IT-T X.520 Information technology – Open Systems Interconnection – The Directory: Selected attribute types
[25]	RFC 5758	Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA
[26]	RFC 5759	Suite B Certificate and Certificate Revocation List (CRL) Profile
[27]	RFC 5480	Elliptic Curve Cryptography Subject Public Key Information
[28]	RFC 4519	Lightweight Directory Access Protocol (LDAP):
[29]	NIST SP 800-56A	NIST Special Publication SP 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revision 2), May 2013
[30]	ITU E.212	The international identification plan for public networks and Subscriptions
[31]	ETSI TS 102 223	Smart Cards; Card Application Toolkit (CAT)
[32]	ETSI TS 124 008	Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3
[33]	ETSI TS 101 220	Smart Cards; ETSI numbering system for telecommunication application providers
[34]	RFC 768	User Datagram Protocol, Aug 1980.
[35]	ETSI TS 123 003	Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification
[36]	3GPP2 S.R0048-A	3GPP2 - 3G Mobile Equipment Identifier (MEID)
[37]	ISO/IEC 7812-1:2015	Identification cards -- Identification of issuers -- Part 1: Numbering system

2 General Architecture

This section contains a technical description and architecture of the Remote SIM Provisioning System for consumer Devices. The statements in this section define the basic characteristics that need to be taken into account when reviewing this specification.

2.1 General Architecture

This section further specifies the Roles and interfaces associated with the Remote SIM Provisioning and Management of the eUICC for consumer Devices.

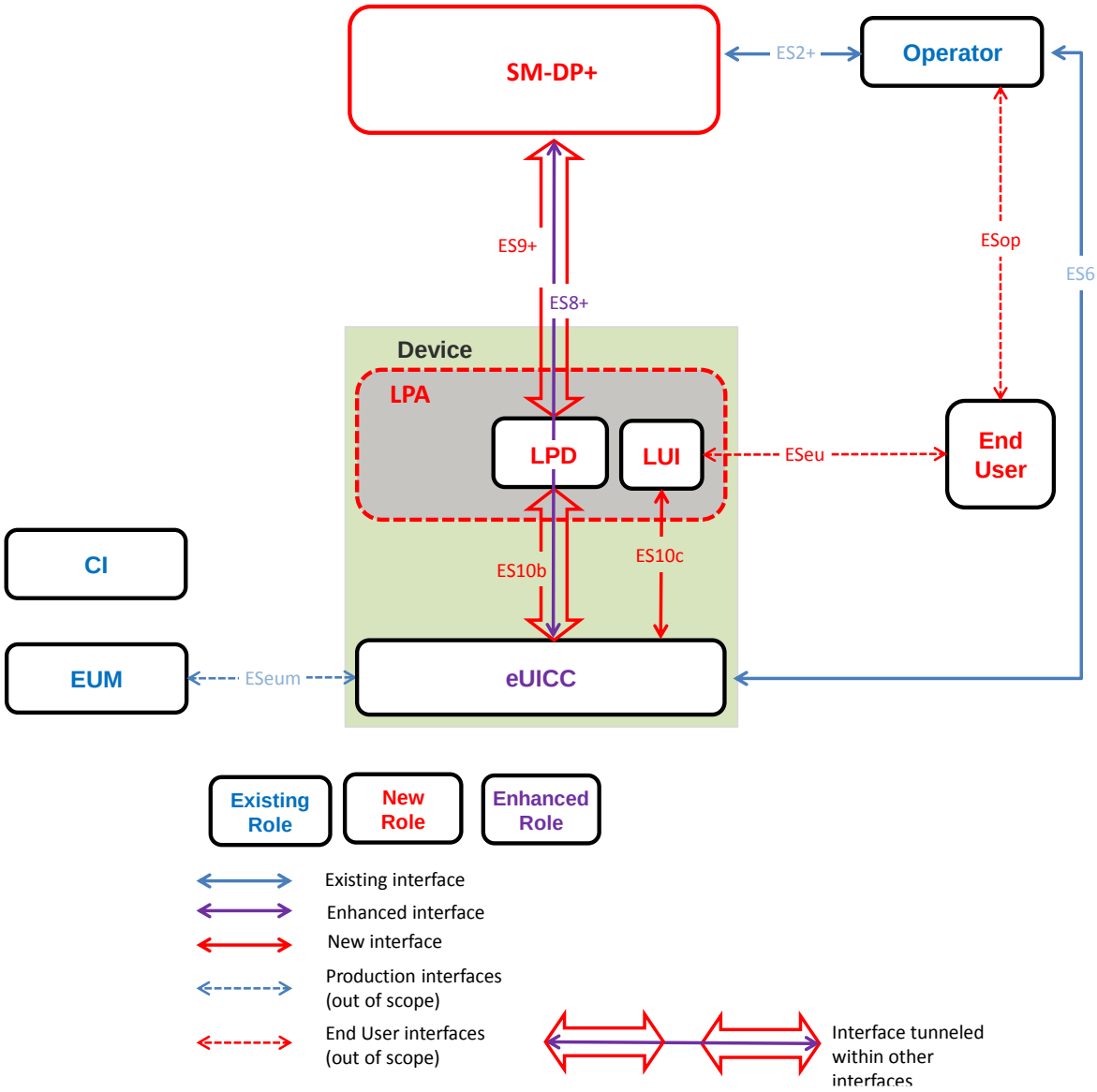

Figure 1: Remote SIM Provisioning System

The above figure provides the complete description of the consumer Remote SIM Provisioning and Management system.

2.2 Roles

Roles are defined within SGP.21 [4] Architecture Specification section 3.

2.3 Interfaces

The following table provides information about the interfaces within the architecture

Interface	Between		Description
ES2+	Operator	SM-DP+	Used by the Operator to order Profile Package preparation for specific eUICCs and delivery of the Profile Package.

Interface	Between		Description
ES6	Operator	eUICC	Used by the Operator to update an Enabled Profile in the eUICC, and Operator services management.
ES8+	SM-DP+	eUICC	Provides a secure end-to-end channel between the SM-DP+ and the eUICC for the administration of the ISD-P and the associated Profile(s) during download and installation. It provides Perfect Forward Secrecy
ES9+	SM-DP+	LPA (LPD)	Used to provide a secure transport between the SM-DP+ and the LPA (LPD) for the delivery of the Profile Package.
ES10b	LPD	eUICC	This interface is used by the LPA to transfer a Profile Package to the eUICC, this interface plays no role in the decryption of Profile Packages.
ES10c	LUI	eUICC	This interface is used for local End User management of Profiles installed on the eUICC.
ESop	Operator	End User	These interfaces will not be specified within the technical specifications, They are specific to business relationships between entities involved.
ESeu	End User	LUI	
ESeum	eUICC	EUM	

Table 1:Interfaces

2.4 eUICC Architecture

2.4.1 eUICC Overview

This section describes the internal high-level architecture of the eUICC. It SHOULD be noted that the eUICC architecture is very similar to that used in the GSMA Remote SIM Provisioning of Embedded UICC Technical specification SGP.02 [2]. Operator Profiles are stored inside security domains within the eUICC and are implemented using Global Platform standards. These ensure that it is impossible for any Profile to access the applications or data of any other Profile stored on the eUICC. The same mechanism is currently in use within SIM cards to ensure payment applications are kept secure.

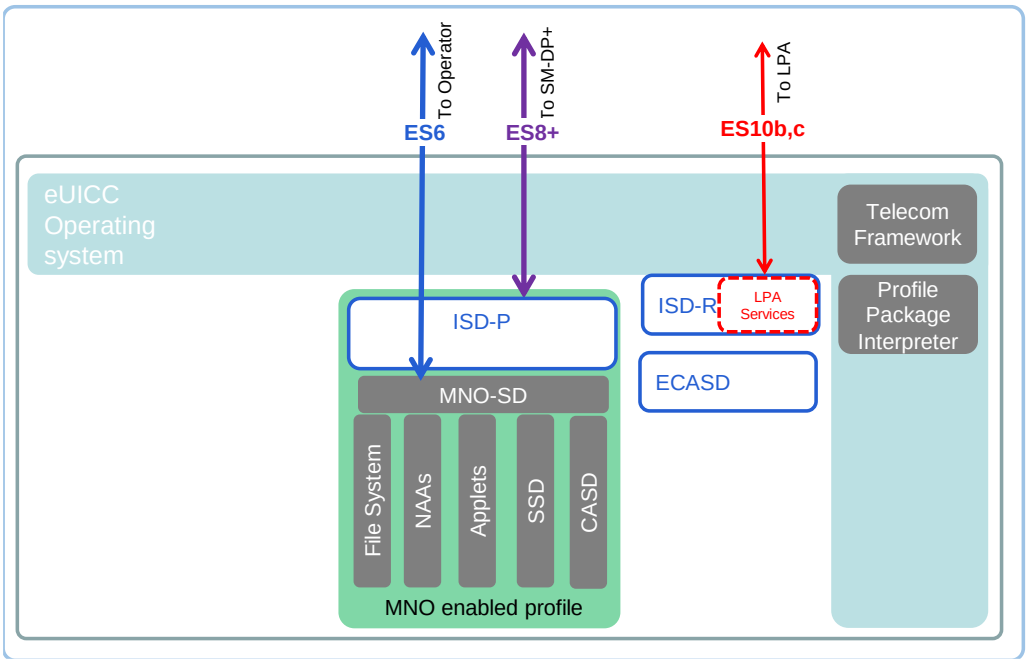

Figure 2: Schematic Representation of the eUICC

2.4.2 ECASD

The Embedded UICC Controlling Authority Security Domain (ECASD) is responsible for secure storage of credentials required to support the required security domains on the eUICC.

There SHALL be only one ECASD on an eUICC. The ECASD SHALL be installed and personalized by the EUM (eUICC Manufacturer) during the eUICC manufacturing. After eUICC manufacturing, the ECASD SHALL be in life-cycle state PERSONALIZED as defined in GlobalPlatform Card Specification [8] section 5.3.

The AID of the ECASD SHALL follow SGP.02 [2].

The ECASD contains:

- The eUICC's Private Key (SK.EUICC.ECDSA) for creating ECDSA signatures
- The eUICC's Certificate for eUICC authentication (CERT.EUICC.ECDSA) containing the eUICC's public key (PK.EUICC.ECDSA)
- The Certificate Issuer's (CI) Public Key (PK.CI.ECDSA) for verifying SM-DP+ certificates
- The certificate of the EUM (CERT.EUM.ECDSA)
- eUICC Manufacturer's (EUMs) keyset for key/certificate renewal

NOTE: FFS for Phase 2 the ECASD SHALL provide the following services to the ISD-R and to ISD-Ps:

- eUICC signature creation on material provided by an ISD-R or an ISD-P
- Verification of the Certificates of SM-DP+ provided by an ISD-R or an ISD-P with the CI public key (PK.CI.ECDSA)

NOTE: In Phase 1 there is no usage from ISD-P. This is FFS for Phase 2.

Personalisation of the ECASD SHALL be done in a certified 'GSMA SAS environment.'

NOTE: Per NIST publication SP800-57 part.1, ECC256 (128-bit security strength) is sufficient for current implementation beyond year 2031. It is FFS for Phase 2 whether higher security is needed to be future proof.

2.4.3 ISD-R

The ISD-R is responsible for the creation of new ISD-Ps and lifecycle management of all ISD-Ps.

2.4.4 ISD-P

The ISD-P is the on-card representative of the SM-DP+ and is a secure container (Security Domain) for the hosting of an Operator Profile. The ISD-P is used for the Profile download and installation in collaboration with the Profile Package Interpreter for the decoding/interpretation of the received Profile Package.

2.4.5 MNO–SD

The MNO-SD is the on-card representative of the Operator. It contains the Operator's Over-The-Air (OTA) Keys and provides a secure OTA channel.

2.4.6 Telecom Framework

The Telecom Framework is an Operating System service that provides standardised network authentication algorithms to the NAAs hosted in the ISD-Ps. Furthermore, it provides the capabilities to configure the algorithm with the needed parameters.

2.4.7 Profile Package Interpreter

An eUICC Operating System is a service that translates the Profile Package data as defined in SIMalliance eUICC Profile Package Specification [5] into an installed Profile using the specific internal format of the target eUICC.

2.4.8 LPA Services

This role provides the necessary access to the services and data required by LPA functions. These services are:

- Profile Package transfer from the LPA to the ISD-P
- List of installed Profiles
- Retrieve EID
- Local Profile Management Operations

2.5 Profile Protection and Delivery

This section describes how an Operator's Profile is protected within a Profile Package prior to download to the eUICC. This also applies when the Profile Package is protected during transmission between Roles within the system.

2.5.1 Profile Package Types Overview

From generation to download, the Profile Package will take different formats. This specification uses the following identification:

- Unprotected Profile Package (UPP): Raw SIMalliance TLV sequence
- Protected Profile Package (PPP): Segmented and protected in SCP03t TLVs
- Bound Profile Package (BPP): Prepended with session key agreement info, key replacement package, ISD-P creation and configuration info
- Segmented Bound Profile Package (SBPP): BPP segmented into STORE DATA APDU script for loading into eUICC. This step in performed by the LPD when LPD is in the Device.

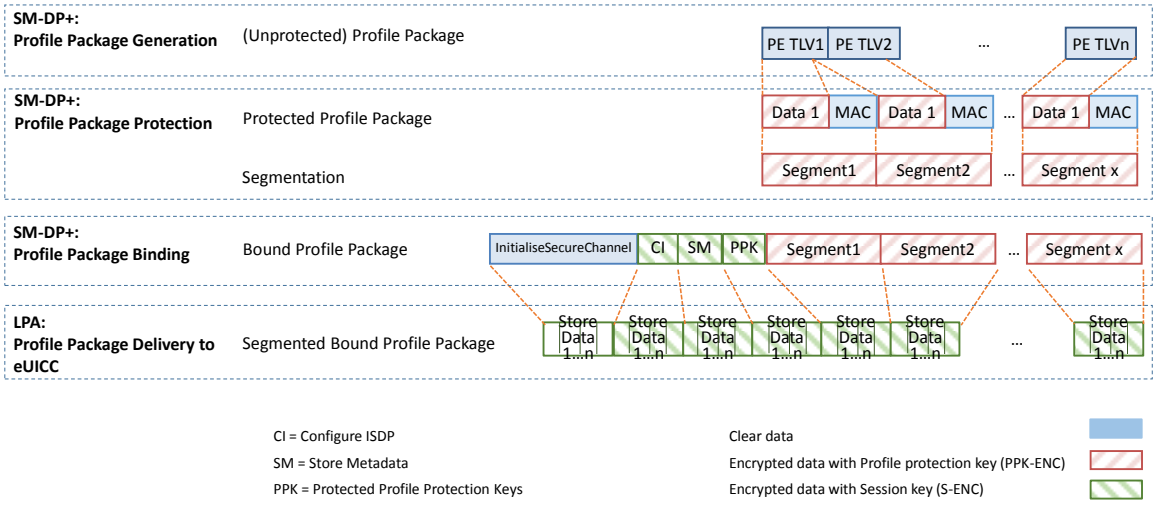

Figure 3: Profile Package stage Description

The above diagram **Error! Reference source not found.** describes the case where Profile Package is protected with keys different from the session keys established during the key agreement with the eUICC (S-ENC, S-MAC). It MAY also be possible to have a Profile Package protected with the session keys; in that case the 'Profile Protection keys' block SHALL not be present.

2.5.2 Unprotected Profile Package

The Unprotected Profile Package (UPP) is generated by the SM-DP+, within the Profile Package Generation function. The Profile Package Generation takes as input the profile specifications established with the Operator and input data provided by the Operator. The processes of profile specification and input data acquisition are out of scope of this specification.

The Unprotected Profile Package consists of a sequence of Profile Element (PE) TLVs according to the SIMalliance specification [5].

2.5.3 Protected Profile Package

The Protected Profile Package (PPP) is generated by the SM-DP+, within the Profile Package Protection function.

The PPP is protected with SCP03t. Command TLV encryption and MACing follows SGP.02 [2] section 5.1.1.3.5. During this step the internal UPP structure is not considered, and rather seen as a unique block of data. That block of data is split into segments of a maximum size

of 1020 bytes (including the tag, length field and MAC). The eUICC SHALL support receiving data segments of at least up to this size.

NOTE: From the 1020 bytes of each data segment, only 1008 bytes are usable for payload (deduced the 1 byte tag, 3 bytes length field and 8 bytes MAC) . Considering the necessary padding during encryption (16 bytes length block encryption and necessary '80' byte padding), then each data segment can only contain 1007 bytes of the PPP data block.

Profile protection can optionally be performed using either:

- Session keys (S-ENC, S-MAC) resulting from the key agreement with eUICC (see Matching ID section 4.1.1**Error! Reference source not found.**).
- Or random keys (denoted PPK-ENC and PPK-MAC in this document and referred to as S-ENC and S-MAC respectively in SGP.02 [2], generated by the SM-DP+.

If random key mode is selected by the SM-DP+, an additional random key of 128 bits length for R-MAC computation/verification SHALL be generated (denoted PPK-RMAC and referred to as S-RMAC in SGP.02 [2]). This key SHALL be used by the eUICC to protect the response and used by the SM-DP+ to verify it.

If random key mode is selected by the SM-DP+, the initial MAC chaining value to be used for the first segment of the PPP is provided together with the random key (see section 5.2.2.4) and the ICV for encryption is reset to its initial state (i.e. the value on 16 bytes is '00…01'). Else the MAC chaining method defined in SGP.02 [2] SHALL be applied (i.e. the MAC chaining value of the previous SHALL be used.

S-ENC, S-MAC, S-RMAC, PPK-ENC, PPK-MAC and PPK-RMAC SHALL be 128 bits length.

Each data segment of the PPP is identified by the tag '86' as defined in SGP.02 [2].

It is the SM-DP+ choice to use this random keyset (PPK-ENC, PPK-MAC and PPK-RMAC). This mode allows to perform Profile Package Protection in advance without having any eUICC knowledge. It MAY help to provide a better SM-DP+ scalability. The eUICC SHALL be able to support both modes.

In case the random key mode is used, the PPP is not bound to any particular eUICC or ISD-P AID value at this stage.

2.5.4 Bound Profile Package

The Bound Profile Package (BPP) is generated by the SM-DP+, within the Profile Package Binding function. The purpose of this operation is to link a Protected Profile Package to a particular eUICC. This is done within a key agreement between the eUICC and the SM-DP+. See Download and install procedure (see section 3.1.2).

The BPP comprises a sequence of TLV commands (in this order):

- TLV command for Key agreement in clear.
- Set of SCP03t payload TLVs (tag '87') containing:
    - TLV commands for ConfigureISDP and StoreMetadata

- (Optional) TLV command for 'Profile Protection keys'

- Followed by the SCP03t payload TLVs (tag '86') of the PPP

Error! Reference source not found.Error! Reference source not found. gives an overview of the full data structure of the Bound Profile Package.

Tag	Length	Value Description	MOC
'BF23'	Var	Tag for 'ES8+.InitialiseSecureChannel' function Content: TLV for 'ES8+.InitialiseSecureChannel' function (See section 5.2.2.1)	M
'87'	Var	SCP03t segment containing ConfigureISDP, protected with session keys resulting of the key agreement (S-ENC, S-CMAC) (See section 2.6)**Error! Reference source not found.** Content: TLV for 'ES8+. ConfigureISDP' function(s. See section 5.2.2.2) The receipt calculated during the generation of the session keys is used as first MAC chaining value.	M
'87'	Var	SCP03t segment containing StoreMetadata, protected with session keys resulting of the key agreement (S-ENC, S-CMAC) (See section 2.6) Content: TLV for 'ES8+. StoreMetadata' function ; (See section 5.2.2.3)	M
'87'	Var	SCP03t segment containing the remainder of StoreMetadata if one '87' TLV is not able to contain the whole data structure.	C
'87'	Var	SCP03t segment containing the Profile Protection Keys, protected with session keys resulting of the key agreement (S-ENC, S-CMAC). (See section 2.6). Content: TLV for 'ES8+.ReplaceSessionKeys' function (See section 5.2.2.4)	O
'86'	Var	SCP03t payload, segment b1 protected with Profile Protection keys (PPK-ENC, PPK-MAC) or with session keys resulting of the key agreement (S-ENC, S-CMAC). (See section 2.6).	M

Table 2: Bound Package Profile Data Structure

2.5.4.1 Description of 'InitialiseSecureChannel' block

This block comprises the TLVs for opening a remote personalization session with eUICC, including the key agreement.

These TLVs are part of the function ES8+.InitialiseSecureChannel function are listed and described in section 5.2.2.1. These TLVs SHALL not be encrypted. Integrity and authenticity are ensured by the signatures.

The execution of this function by the eUICC will result in the generation of the SCP03t session keys, denoted S-ENC, S-MAC, S-RMAC and initial MAC chaining value, that will be used by the SM-DP+ to protect subsequent TLVs.

2.5.4.2 Description of 'ConfigureISDP' block

This block comprises one TLV for ISD-P creation and configuration.

The TLV is part of the function ES8+.ConfigureISDP and is listed and described in section 5.2.2.2. This TLV SHALL be encrypted and MACed with the SCP03t session keys.

2.5.4.3 Description of 'StoreMetadata' block

This block comprises one or two TLV(s) containing metadata of the Profile.

The TLV(s) are part of the function ES8+.StoreMetadata and are listed and described in section 5.2.2.3. These TLV(s) SHALL be encrypted and MACed with the SCP03t session keys.

2.5.4.4 Description of 'Profile Protection keys' block

The ‘Profile Protection keys’ block contains the function ES8+.ReplaceSessionKeys to replace the session S-ENC, S-MAC and S-RMAC keys resulting of the key agreement, by the keys used for protecting the Protected Profile Package, PPK-ENC, PPK-MAC and PPK_RMAC.

This function is protected by SCP03t with the S-ENC, S-MAC keys resulting of the key agreement.

This block is optional depending on the mode selected by the SM-DP+ to protect the Profile package (see section 2.5.3).

NOTE: The response of this function shall be protected with the initial S-ENC and S-RMAC as described in section 2.5.6.1.

2.5.5 Segmented Bound Profile Package

The Segmented Bound Profile Package (SBPP) is generated by the LPA, when the LPA is present, to transfer the Bound Profile Package to the eUICC using the local interface ES10b.

The segmentation SHALL be done in blocks of 255 bytes or less according to the structure of the Bound Profile Package, i.e. each TLV of the BPP that is up to 255 bytes is transported in one APDU. Larger TLVs are sent in blocks of 255 bytes for the first blocks and a last block that MAY be shorter.

2.5.6 Profile Installation Result

The Profile Installation Result is composed of a Profile Installation Report and a Profile Installation Receipt. The Profile Installation Result data structure is detailed in section 5.2.3.5 (function ES10b.GetProfileInstallationResult).

The Profile Installation Report SHALL be created along the process of BPP loading as the concatenation of the individual protected Response messages of each TLV of the BPP.

The Profile Installation Receipt SHALL be created by the eUICC after the execution of the last TLVs of the BPP, or right after the first BPP’s TLV executed with error.

Profile Installation Result is returned at the end of processing the BPP.

Until the Profile Download and Installation process is completed, no Profile Installation Result is available for the LPA.

Both Profile Installation Report and Receipt SHALL be kept by the eUICC until explicitly deleted by the LPA, after successfully delivered to the SM-DP+. Before being deleted Profile Installation Result MAY be retrieved at any time by the LPA.

The eUICC SHALL only store the Profile Installation Result of the last Profile installation. When a new Profile Download and Installation starts, an old Profile Installation Result still stored SHALL be deleted.

2.5.6.1 Profile Installation Report

The Profile Installation Report SHALL be the concatenation of the individual protected Response messages of each TLV of the BPP.

The first individual protected Response message is the one of the ES8+.InitialiseSecureChannel. It is composed of a not encrypted TLV and a signature (with SK.EUICC.ECDSA, as no R-MAC can be provided at that point of time) over this TLV to ensure integrity and authenticity (see section 5.2.2.1).

The subsequent protected Response messages are built according to SGP.02 [2] and extended to tag ‘87’. The Initial MAC Chaining calculated during the generation of the session keys is used as first MAC chaining value. The MAC chaining method defined in SGP.02 [2] SHALL be applied for subsequent response messages (i.e. the MAC chaining value used for the computation of the R-MAC SHALL be the MAC Chaining value of the corresponding command TLV).

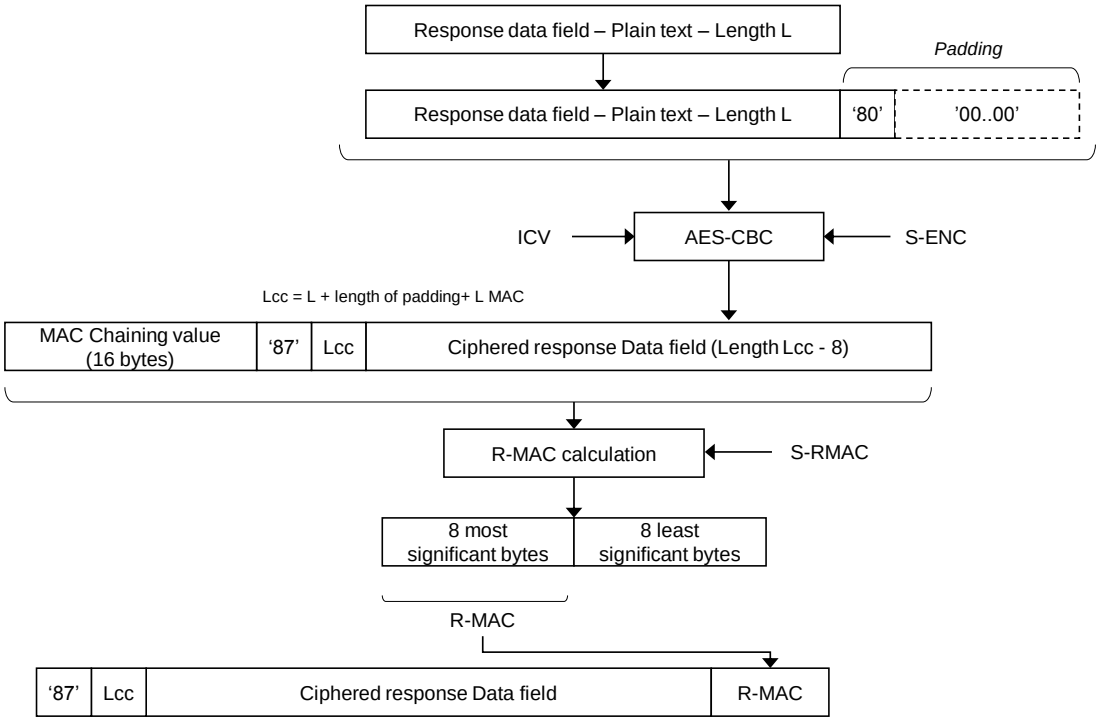

Figure 4: Building of Protected Response messages for BPP’s TLV '87'

The Response messages identified by tag '87' SHALL be protected using S-ENC and S-RMAC resulting from the key agreement (performed during execution of the ES8+.InitialiseSecureChannel).

No R-MAC SHALL be generated and no protection SHALL be applied to a Response message when a system error has occurred; in this case only tag '9F47', followed by one of the possible hereunder defined error code byte, SHALL be included in the Profile Installation Report:

- '01': error in length or structure of command data
- '02': security error

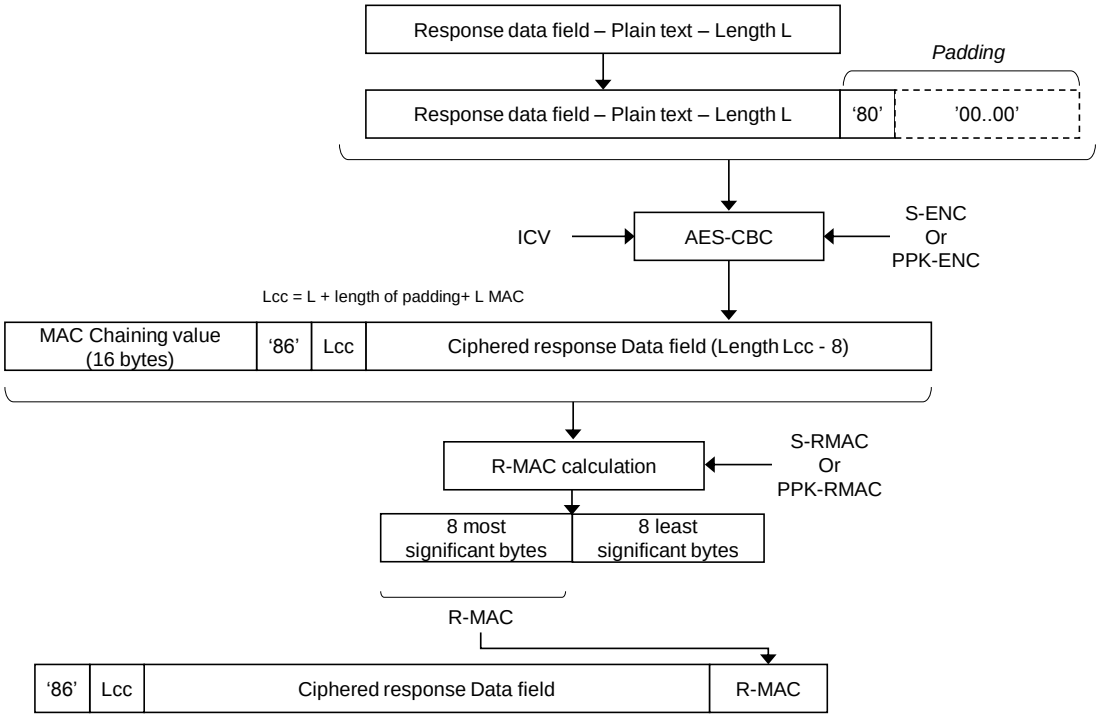

Figure 5: Building of Protected Response messages for BPP’s TLV '86'

The response messages from processing of the Profile Elements contained in the Profile Segments with tag '86' shall be collected by the eUICC until the end of the BPP is reached or an error which terminates processing is encountered. This data is included in one Response message identified by tag '86' which SHALL be protected using S-ENC and S-RMAC resulting from the key agreement (performed during execution of the ES8+.InitialiseSecureChannel), or PPK-ENC and PPK-RMAC if BPP was containing an ES8+.ReplaceSessionKeys command. ICV and MAC chaining value shall be set to the values of the first Profile segment with tag '86'.

No R-MAC SHALL be generated and no protection SHALL be applied to a Response message when a system error has occurred; in this case only tag '9F46' followed by an error code byte as defined in SGP.02 [2] SHALL be included in the Profile Installation Report.

R-MAC calculation is a Cipher-based MAC (CMAC) using AES-128 as defined in GlobalPlatform Amendment D [11].

: Profile Installation Report provides an overview of the Profile Installation Report structure for a successful installation:

Tag	Length	Value Description	MOC
‘BF23’	Var	Tag for ‘ES8+.InitialiseSecureChannel’ function response	M
‘5F37’	Var	eUICC’s signature for ‘ES8+.InitialiseSecureChannel’ function response	M
‘87’	Var	Protected Response message (encrypted and with R-MAC) for ES8+.ConfigureISDP function	M
‘87’	Var	Protected Response message (encrypted and with R-MAC) for ES8+.StoreMetadata function	M
'87'	Var	Protected Response message (encrypted and with R-MAC) for second segment of ES8+.StoreMetadata function – present if a second segment is present in the BPP	C
‘87’	Var	Protected Response message (encrypted and with R-MAC) for ES8+.ReplaceSessionKeys – present if the function is present in the BPP.	C
‘86’	Var	Protected Response message (encrypted and with R-MAC)	M

Table 3: Profile Installation Report

: Profile Installation Report Failed Device provides an example of Profile Installation Report structure for a failed installation resulting of an error during ES8+.StoreMetadata function execution:

Tag	Length	Value Description	MOC
‘BF23’	Var	Tag for ‘ES8+.InitialiseSecureChannel’ function response	M
‘5F37’	Var	eUICC’s signature for ‘ES8+.InitialiseSecureChannel’ function response	M
‘87’	Var	Protected Response message (encrypted and with R-MAC) for ES8+.ConfigureISDP function	M
‘87’	Var	Protected Response message (encrypted and with R-MAC) for ES8+.StoreMetadata function	M

Table 4: Profile Installation Report Failed Device

Error! Reference source not found. Provides an example of a Profile Installation Report structure for a failed installation resulting of system error at ES8+.StoreMetadata function execution:

Tag	Length	Value Description	MOC
‘BF23’	Var	Tag for ‘ES8+.InitialiseSecureChannel’ function response	M
‘5F37’	Var	eUICC’s signature for ‘ES8+.InitialiseSecureChannel’ function response	M
‘87’	Var	Protected Response message (encrypted and with R-MAC) for ES8+.ConfigureISDP function	M
‘9F47’	Var	Unprotected error code as defined above.	M

Table 5: Profile Installation Report Failed System

2.5.6.2 Profile Installation Receipt

The Profile Installation Receipt contains the following data:

- Result Code: these two bytes provide the final Profile installation status:
  - '00 00': Installation successful
  - '00 01': Installation successful with Warning" as specified in SIMalliance specification [5]
  - '01 01': ICCID already exists on the eUICC
  - '01 02': Insufficient memory for the Profile
  - '01 03': Profile Installation failed due to interruption
  - 'XX XX': Any other value means "Installation failed". The value is the error code that provides further detail on the error.
- Transaction ID: The Transaction Identifier given to the eUICC during the Profile "Download and Installation" procedure (see section 3.1.2).
- SM-DP+ Address: The SM-DP+ address as an FQDN given to the eUICC during the Profile "Download and Installation" procedure (see section 3.1.2).
- SM-DP+ ID (optional): The SM-DP+ identifier given to the eUICC during the Profile Download and Installation" procedure (see section 3.1.2).
- Signature: A signature created by the eUICC ensuring the authenticity and the integrity of the Profile Installation Receipt.

Table 6: Profile Installation Receipt described the structure of the Profile Installation Receipt

Tag	Length	Value Description			MOC
'BF27'	Var	Profile Installation Receipt			M
		Tag	**Length**	**Value Description**	
		'80'	1-16	Transaction ID	M
		'81'	2	Result code	M
		'5F50'	Var	SM-DP+ address	M
		'06'	1-16	SM-DP+ OID	O
'5F37'	Var	eUICC's signature. Computed as described in GlobalPlatform Card Specification Amendment E [12], using the eUICC private key SK.EUICC.ECDSA across the TLV 'BF27' (Profile Installation Receipt)			M

Table 6: Profile Installation Receipt

2.6 Protocol for Profile Protection and eUICC binding

The Profile is protected by security mechanisms which are based on SCP11a as specified by GlobalPlatform Card Specification Amendment F [13].

This section describes the differences between SCP11a and the Protocol for Profile Protection. The SM-DP+ plays the role of the OCE (Off Card Entity) specified in GlobalPlatform Card Specification Amendment F [13].

- The mutual authentication defined in SCP11a is modified: Whereas in SCP11a, authentication is achieved by a shared secret calculated from static key pairs being fed into the generation of the session keys, in the Protocol for Profile Protection signatures of each side is used to authenticate to the other side. ECKA certificates are not used for mutual authentication.

NOTE: Using ECDSA signature keys for the authentication during the key establishment allows the same keys also to be used to sign other content, e.g. notifications. Only one certificate per entity is required in this case.

- For the retrieval of the eUICC Certificate, the GET DATA command as specified in section 6.2 of GlobalPlatform Card Specification Amendment F [13] is used.
- Ephemeral keys are renamed to one-time keys in this specification, as they MAY live longer and are stored in non-volatile memory. With respect to Forward Secrecy, they serve the same purpose.
- The command data of the PERFORM SECURITY OPERATION and the MUTUAL AUTHENTICATE command is sent to the ISD-R encapsulated in TLVs. The ISD-R SHALL not persistently store any SM-DP+ public key (see GlobalPlatform Card Specification Amendment F [13] sections 4.1 and 4.2).
- Establishment of the session keys SHALL use only the shared secret generated from the one-time key pairs.
- The first TLV(s) following the data for key establishment are protected with the session keys generated in the key agreement. MACing and encryption is done as specified for SCP03t in SGP.02 [2] (see NOTE below).
- The data contains the ISD-P configuration data. When this is processed by the eUICC, the ISD-P is created.
- Optionally, the session keys can be replaced by the Profile protection keys. The Profile protection keys are themselves secured by the session, Subsequent data is exchanged as TLVs as specified for SCP03t in SGP.02 [2] (see NOTE below), protected by the Profile protection keys.

NOTE: This specification only reuses part of SCP03t as specified in SGP.02 [2]. Session key establishment is done via the modified PERFORM SECURITY OPERATION and the MUTUAL AUTHENTICATE commands as specified above. Thus no INITIALIZE UPDATE and EXTERNAL AUTHENTICATE command TLVs are required. Only the MACed and encrypted data TLVs (tag '86') are used in the context of this specification.

3 Procedures

This section specifies the Procedures associated with the Remote SIM Provisioning and Management of the eUICC for consumer Devices.

3.1 Remote Provisioning

3.1.1 Profile Download Initiation (Informative)

Normal Case:

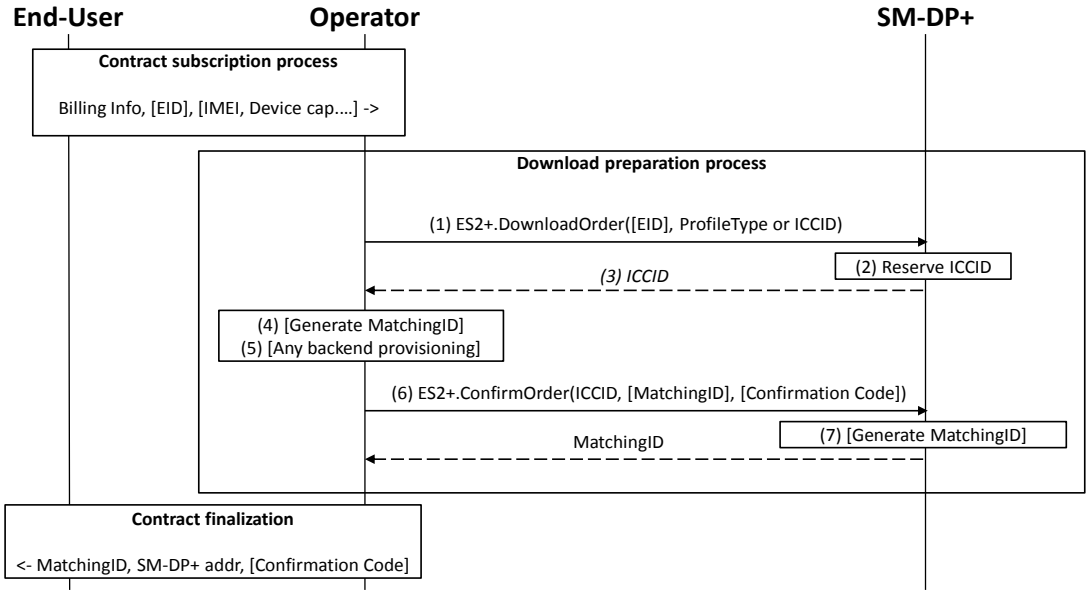

Figure 6: Profile Download Initiation

Start Conditions:

The End User has selected the Operator with which he wants to sign a contract with.

The End User MAY initiate the process:

- From any other Device (e.g. PC)
- Through a Customer Agent of the Operator
- Or any other convenient means provided by the Operator

Procedure:

The download initiation procedure consists of the following sub-processes:

- Contract Subscription process
- Download preparation process
- Contract Finalization

NOTE: This section describes the case where these sub-process are performed in the described order. In this case, it is most likely that the Download and installation procedure will happen right after. There also are cases where these sub-processes MAY be performed in different order like B -> A -> C or B -> C -> A (e.g. for prepaid Subscription). In these cases the download order requested to SM-DP+ MAY remain pending for a significant amount of time.

3.1.1.1 Contract Subscription process (Informative)

The contract selection process, while being out of scope of this specification, is given as it SHALL happen prior to the Profile download and installation procedure (see section 3.1.2). This process description describes the information exchanged and data that are used as input data for the Profile download and installation procedure.

This process can be performed at an Operator's Point of Sale (POS), using the Operator's web portal from a Device not being the one onto which the Profile will be downloaded (e.g. a PC) or from a web browser on the Primary Device, or even using a companion application on the Primary Device. Any other mean defined by the Operator can also be possible as far as it provides a convenient End-user experience and it provides the expected output data required for the execution of the Profile download and installation procedure.

During the execution of the process of contract Subscription, the Operator acquires the necessary information. As part of these data, the EID and IMEI of the target Device MAY be provided, and related Device capabilities MAY be acquired (e.g. based on the TAC information comprised in the IMEI). Acquisition and verification of these capabilities are out of scope of this specification. Additional information such as contract details, user details, payment details and similar are also out of scope of this specification.

If the EID and the IMEI are provided, the Operator can verify if the target Device (both eUICC and Device can be relevant for this verification) is supported, and determine the Profile Type for the target Device and the offer given to the End-user. If no information about the target Device is provided, this preliminary verification cannot be performed and it will be performed during the execution of the Profile download and installation procedure (see section 3.1.2). For Additional info please see Annex F on Profile eligibility check.

If EID and IMEI are provided and the Operator cannot provide an appropriate Profile, the process fails and stops at this point.

3.1.1.2 Download preparation process

1. The Operator calls the ES2+.DownloadOrder (see section 5.3.2.1) function of the SM-DP+ with its relevant input data.

'EID' is optional. One of the value 'ProfileType' or 'ICCID' shall be provided. If ICCID is given, the SM-DP+ SHALL verify that this ICCID is available. If 'ProfileType' is given, the SM-DP+ SHALL pick-up one of the related ICCID in its inventory.

The SM-DP+ MAY optionally verify additional compatibility between the eUICC (if EID is provided) and the requested Profile Type. This verification is out of scope of this specification.

NOTE: If no EID is given at this stage, the Operator MAY be involved later during the Download and installation procedure to determine the right ‘ProfileType’/’ICCID’ in case the provided 'ProfileType'/'CCID' is not compatible with the eUICC identified by the EID once it is acquired by SM-DP+ during Download and installation procedure. See Annex F on Profile eligibility check.

2. The SM-DP+ reserves the ICCID for this request. At this stage the SM-DP+ MAY simply pick the related Protected Profile Package in its inventory or generate and protect the Profile corresponding to this ICCID.
3. The SM-DP+ returns the acknowledged ICCID (SHALL be the same value as the received one, if any).
4. Optionally, the Operator MAY generate a MatchingID (see section 4.1.1).

5. At this stage the Operator knows the ICCID selected for this contract Subscription. It MAY perform any relevant operation on its back-end (e.g. provisioning of HLR). If an error occurs during this step, the process fails and stops at this point.
6. And 7 The Operator SHALL confirm the download order by calling the ES2+.ConfirmOrder (see section 5.3.2.2) function of the SM-DP+ with its relevant input data. If generated in Step 4, the MatchingID SHALL be included in the input data and then the SM-DP+ SHALL return the acknowledged value that is the same as the received one. Otherwise, the SM-DP+ SHALL generate a MatchingID and return the generated value to the Operator. The ICCID SHALL be associated to the generated MatchingID.
7. If it is required for the End User to enter the Confirmation Code to download the Profile, the Confirmation Code SHALL be included in the input data of the ES2+.ConfirmOrder (see section 5.3.2.2) function.

3.1.1.3 Contract Finalization (Informative)

The Operator SHALL provide the End User with the MatchingID, SM-DP+ address, and optional Confirmation Code.

In this version of this specification, the MatchingID and SM-DP+ address are provided via Activation Code as described in section (4.1). If the optional Confirmation Code is to be used, it SHOULD be provided to the End User separately from the Activation Code

3.1.2 Download and Installation

This section describes the Profile Download and Installation procedure.

```
@startuml
hide footbox
skinparam sequenceMessageAlign center
skinparam sequenceArrowFontSize 11
skinparam noteFontSize 11
skinparam monochrome true
skinparam lifelinestrategy solid
'skinparam shadowing false

participant "<b>SM-DP+" as DP
participant "<b> LPA " as LPA
participant "<b>eUICC" as E

rnote over LPA #FFFFFF : [1] Get SM-DP+ Address, Activation Code Token, [SM-DP+
OID] from AC

rnote over LPA #FFFFFF : [2] [Prompt the End User to input Confirmation Code]
LPA -> E : [3] ES10b.GetEUICCChallenge
rnote over E #FFFFFF : [4] ES10b.GetEUICCChallenge
E --> LPA : [5] euiccChallenge
LPA -> E : [5a] [ES10b.GetEUICCInfo]
E --> LPA : [5b] [euiccInfo1]
LPA -> DP : [6] ES9+.InitiateAuthentication \n (euiccChallenge, SVN, euiccInfo1,
SM-DP+ Address)
rnote over DP #FFFFFF
[7]
- Check SM-DP+ Address
- Check euiccInfo1
- Generate TransactionID
- Generate SM-DP+ Challenge
- Build dpSigned1 = {euiccChallenge,
 SM-DP+ Challenge, TransactionID, SM-DP+ Address}
- Compute smdpSignature1 over dpSigned1
```

```
endrnote
DP --> LPA : [8] TransactionID, dpSigned1, smdpSignature1,\n certFormatTobeUsed,
curveTobeUsed,CERT.DP.ECDSA
LPA -> E : [9] ES10b.PrepareDownload\n(dpSign1, smdpSignature1, Activation Code
Token, Device Info, [SM-DP+ OID],\n [Hashed Confirmation Code], certFormatTobeUsed
,\n curveTobeUsed, CERT.DP.ECDSA)


rnote over E #FFFFFF
[10]
- Verify CERT.DP.ECDSA
- Verify dpSigned1 (Authentication of the SM-DP+)
- [Check if SM-DP+ OID is the same as
   Subject Identifier in CERT.DP.ECDSA]
- Generate one time ECKA key pair
   (otPK.EUICC.ECKA, otSK.EUICC.ECKA)
- Generate euiccSigned1 = { SM-DP+ Challenge, TransactionID,
   SM-DP+ Address,
   Activation Code Token, Device Info, [SM-DP+ OID], [Hashed Confirmation Code],
   Device_Info, euicc_Info2}
- Compute euiccSignature1 over euiccSigned1
endrnote

E --> LPA : [11] euiccSigned1, euiccSIgnature1, \n CERT.EUICC.ECDSA, CERT.EUM.ECDSA


LPA -> DP : [12] ES9+.GetBoundProfilePackage \n (euicc_Signed1, euiccSignature1, \n
CERT.EUICC.ECDSA, CERT.EUM.ECDSA)

rnote over DP #FFFFFF
[13]
- Verify CERT.EUM.ECDSA
- Verify CERT.EUICC.ECDSA
- Verify euiccSignature1
- Verify Activation Code Token
-Conditional: Verify Hashed Confirmation Code
- Eligibility Check using Device Info, euiccInfo2
- Generate one time ECKA key pair
   (otPK.DP.ECKA, otSK.DP.ECKA)
- Generate Session Keys
- Generate smdpSign2
   {CRT, otPK.DP.ECKA, otPK.EUICC.ECKA}
- Generate Bound Profile Package
   (including CRT, otPK.DP.ECKA, smdpSign2)
endrnote
DP --> LPA : [14] TranasctionID, Profile Metadata, Bound Profile Package
LPA -> E : [15] ES10b.LoadBoundProfilePackage x N\n(ES8+.InitialiseSecureChannel)

rnote over E #FFFFFF
[16]
- Verify smdpSign2
- Generate Session Keys
endnote
E --> LPA : Response APDU x N
LPA -> E : [17] ES10b.LoadBoundProfilePackage x N\n(ES8+.ConfigureISDP)
E --> LPA : Response APDU x N
LPA -> E : [18] ES10b.LoadBoundProfilePackage x N\n(ES8+.StoreMetadata)
E --> LPA : Response APDU x N
LPA -> E : [19] [ES10b.LoadBoundProfilePackage x N]\n(ES8+.ReplaceSessionKeys)
E --> LPA : [Response APDU x N]
LPA -> E : [20] ES10b.LoadBoundProfilePackage x N\n(ES8+.LoadProfileElements)
E --> LPA : Response APDU x N \n(Profile Installation Result)
LPA -> DP : [21] ES9+.HandleProfileInstallationResult(Profile Installation Result)
DP --> LPA : OK
rnote over DP #FFFFFF
[21.a] SM_DP+ notify the Operator using
ES2+.HandleProfileInstallationResult
(eid, iccid, profileType, completionTimestamp,
```

```
resultCode, operationStatus)
endrnote

LPA -> E : [22] ES10b.DeleteProfileInstallationResult

rnote over E #FFFFFF
[23] Delete ProfileInstallationResult
endrnote

E --> LPA : OK

@enduml
```

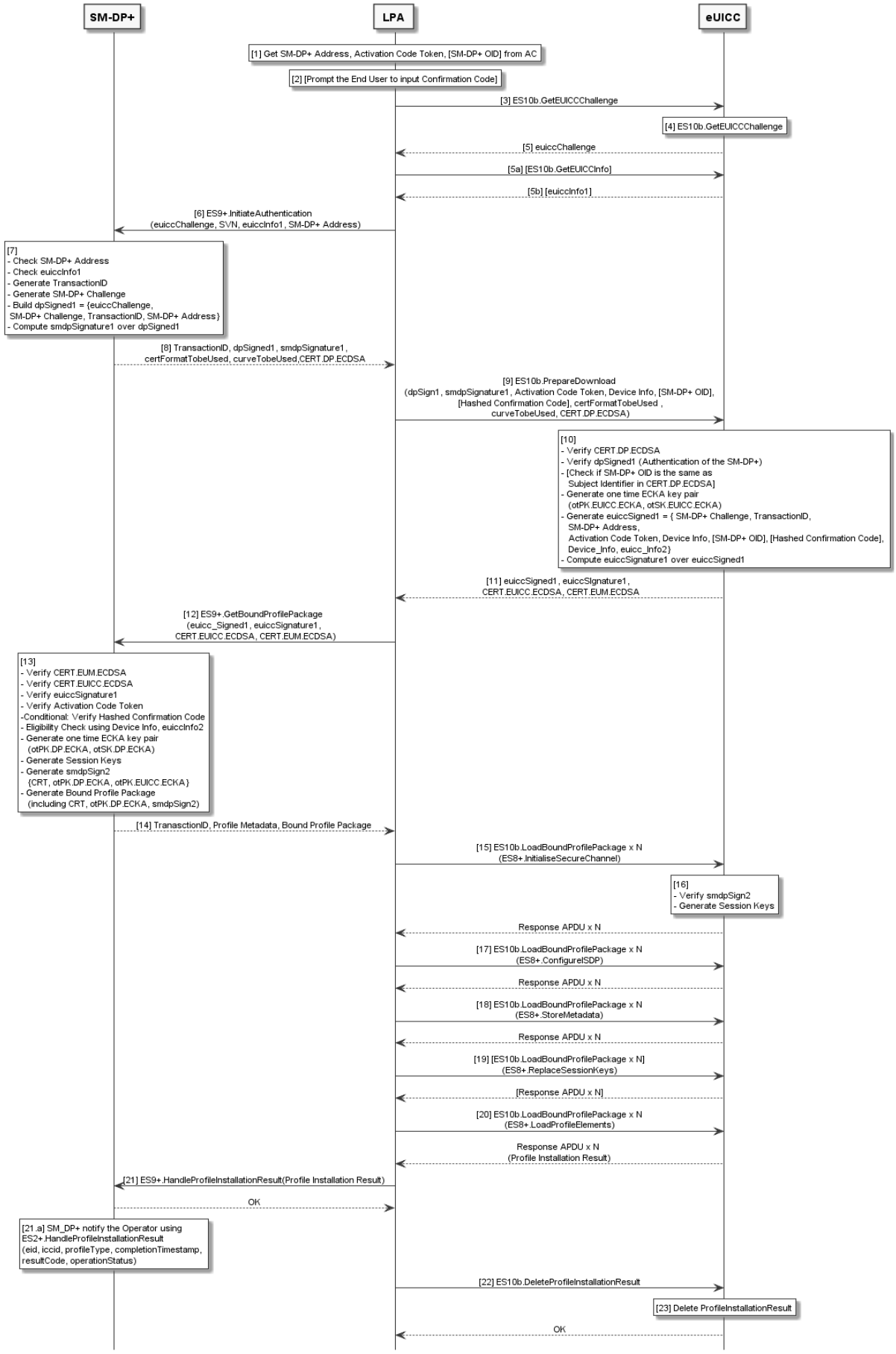

Figure 7: Download and Installation Procedure

Start Conditions:

The SM-DP+ is provisioned with its certificate (CERT.DP.ECDSA), its private key (SK.DP.ECDSA), the CI Certificate (CERT.CI.ECDSA), its TLS Certificate (CERT.DP.TLS) and its TLS Private Key (SK.DP.TLS).

The eUICC is provisioned with its certificate (CERT.EUICC.ECDSA), its private key (SK.EUICC.ECDSA), the EUM Certificate (CERT.EUM.ECDSA) and the CI Public Key (PK.CI.ECDSA).

NOTE: The certificate format MAY be GP or X.509.

The End User has an Activation Code that is coded as described in the section 4.1.

The End User explicitly selects the menu in the LPA for entering the Activation Code.

The End User enters the Activation Code to the LPA by QR code scanning or manual typing.

Procedure:

1. The LPA parses the Activation Code and finds the SM-DP+ address, Activation Code Token, and optional SM-DP+ OID. If the format of the Activation Code is invalid, the procedure SHALL be stopped with an error message to the End User.
2. If the Confirmation Flag in the Activation Code Token indicates “Confirmation Code required”, the LPA SHALL prompt the End User to enter the Confirmation Code which was provided by the Operator.
3. The LPA requests euiccChallenge from the eUICC by calling the "ES10b.GetEUICCChallenge" function (see section 5.2.3.3).
4. The eUICC SHALL generate eUICC Challenge which SHALL be signed later by the SM-DP+ for SM-DP+ authentication by the eUICC.
5. The eUICC returns to the LPA the eUICC Challenge.

    a) Optionally, the LPA MAY request eUICC Information from eUICC by calling the "ES10b.GetEUICCInfo" function. This is required if the LPA hasn’t already retrieved this information.
    b) The eUICC return to the LPA the euiccInfo1.

6. The LPA establishes an HTTPS connection then calls the “ES9+.InitiateAuthentication” function (see section 5.3.3.1) its input data including the euiccChallenge, SVN, eUICCInfo1, and SM-DP+ Address. euiccInfo1 contains SVN, certificateInfo, curveSigningSupport, and curveVerificationSupport.
7. The SM-DP+ checks if the SM-DP+ Address sent by the LPA is valid. If the SM-DP+ Address is not valid, the SM-DP+ SHALL return an error status and the procedure SHALL be stopped.
8. Then, the SM-DP+ checks the received certificateInfo and then determines its Certificate Format and Key Parameter Reference Value which SHOULD be used for this Profile download and installation transaction. If the SM-DP+ determines that it doesn’t support any of requested Certificate formats or curves delivered by the eUICC, it SHALL return an error status and the procedure SHALL be stopped.
   Then, the SM-DP+ SHALL perform the following:

- Generate a TransactionID which is used to uniquely identify the Profile download and installation transaction and to correlate the multiple ES9+ request messages that belong to the same transaction.
- Generate a SM-DP+ Challenge which SHALL be signed later by the eUICC for the eUICC authentication.
- Generate a dpSigned1 data structure containing the eUICC Challenge, smdpChallenge, TransactionID, and SM-DP+ Address.
- Compute the smdpSignature1 over dpSigned1 using the SK.DP.ECDSA.

9. The SM-DP+ sends the TransactionID, dpSigned1, smdpSignature1, certFormatToBeUsed, curveToBeUsed and CERT.DP.ECDSA to the LPA.
   The certFormatToBeUsed and curveToBeUsed indicates to the eUICC which Certificate Format and Key Parameter Reference Value of the eUICC SHALL be used.
10. The LPA calls “ES10b.PrepareDownload” function with input data including the dpSigned1, smdpSignature1, Activation Code Token, Device Info, SM-DP+ OID (if included in the Activation Code), Hashed Confirmation Code (if required by the "Confirmation Code Required Flag" in the Activation Code), Device_Info, certFormatToBeUsed, curveToBeUsed and CERT.DP.ECDSA.
    If the Confirmation Code parameter in the Activation Code Token indicated “Confirmation Code Required”, the LPA calculates the Hashed Confirmation Code as follows:
    Hashed Confirmation Code = SHA256 (SHA256 (Confirmation Code) | SM-DP+ Challenge), where '|' means concatenation of data and the Confirmation Code is the string entered by the End User in the step 2.
11. The eUICC SHALL verify the CERT.DP.ECDSA using PK.CI.ECDSA. When verifying the CERT.DP.ECDSA, the eUICC SHALL also check the following:

    - Whether the format of the CERT.DP.ECDSA is one of its Supported DP+ Certificate Format(s).
    - Whether the Key Parameter Reference Value contained in the CERT.DP.ECDSA is one of its Supported Key Parameter Reference Value for signature verification
    - Then the eUICC SHALL verify the dpSigned1.
    - If any of the verifications fail, the eUICC SHALL return a relevant error status and the procedure SHALL be stopped.
    - If all the verifications succeed, the SM-DP+ is authenticated by the eUICC.
    - If the SM-DP+ OID is provided by the LPA, the eUICC SHALL verify that the Subject Identifier in the CERT.DP.ECDSA is the same as the SM-DP+ OID sent by the LPA. If there is a mismatch, the eUICC SHALL return an error status and the procedure SHALL be stopped.

    Then, the eUICC SHALL perform the following:

    - Generate a one-time ECKA key pair (otPK.EUICC.ECKA, otSK.EUICC.ECKA) using the curve indicated by the Key Parameter Reference Value of CERT.DP.ECDSA.
    - Generate the eUICC Signed1 data structure containing the TransactionID, SM-DP+ Challenge, SM-DP+ Address, SM-DP+ OID (if included in the Activation

Code), Hashed Confirmation Code (if indicated in the Activation Code Token), otPK.EUICC.ECKA and eUICC Info2.
- Compute the euiccSignature1 over euiccSigned1 using SK.EUICC.ECDSA. When generating the euiccSignature1, the eUICC SHALL use the Key Parameter Reference Value indicated in the curveToBeUsed.

12. The eUICC SHALL return the euiccSigned1, uiccSIgnature, CERT.EUICC.ECDSA, and CERT.EUM.ECDSA to the LPA. The Format of the CERT.EUICC.ECDSA and CERT.EUM.ECDSA SHALL be the same as the Certificate Format indicated in the CertFormatToBeUsed.
13. The LPA calls the “ES9+.GetBoundProfilePackage” function with input data including euiccSigned1, euiccSignature1, CERT.EUICC.ECDSA, and the CERT.EUM.ECDSA.
14. When the SM-DP+ receives the “ES9+.GetBoundProfilePackage” function call, it correlates it with the “ES9+.InitiateAuthentication” function processed in the step 7, by verifying the TransactionID match.
    The SM-DP+ SHALL verify the CERT.EUM.ECDSA using PK.CI.ECDSA. Then the SM-DP+ SHALL verify the CERT.EUICC.ECDSA using the PK.EUM.ECDSA contained in the CERT.EUM.ECDSA. Then the SM-DP+ SHALL verify the eUICCSignature1 using the PK.EUICC.ECDSA contained in the CERT.EUICC.ECDSA.
    Then, the SM-DP+ SHALL verify that the SVN included in the eUICCInfo2 is the same as the SVN which was delivered in the step 6.

    Next, the SM-DP+ performs the following checks:
    - The SM-DP+ SHALL check if there is a related pending Profile download order for the MatchingID provided. If not, the request is rejected.
    - Next, if this order is already linked to an EID, the SM-DP+ SHALL also check that the EID matches. If the EID does not match, the request is rejected.
    - If this order is not linked to an EID, no further checks are required.
    - Then the SM-DP+ identifies the Profile to be downloaded, which is associated with the pending order.
    - Then, if the SM-DP+ determines that the Confirmation Code verification is required, it SHALL verify that the Hashed Confirmation Code was received as an input parameter in the “ES9+.GetBoundProfilePackage” function call and compare the received Hashed Confirmation Code and the Hashed Confirmation Code calculated by the SM-DP+.

    NOTE: The SM-DP+ SHALL protect against excessive incorrect entries of the Confirmation Code.

    If any verification fails, the SM-DP+ SHALL return an error status to the LPA and the procedure SHALL be stopped.
    If all the verifications succeed, the SM-DP+ performs appropriate eligibility checks, based on the Device Info and eUICCInfo2. See Annex F for more information on Eligibility checks.
    If the eligibility check fails, the SM-DP+ SHALL return an error status to the LPA and the procedure SHALL be stopped.
    If the eligibility check succeeds, the SM-DP+ SHALL perform the following:

- Generate a one-time ECKA key pair (otPK.DP.ECKA, otSK.DP.ECKA) using the curve indicated by the Key Parameter Reference Value of CERT.DP.ECDSA.
- Generate Session Keys using the CRT, otPK.eUCC.ECKA, and otSK.DP.ECKA.
- Generate smdpSign2 across the CRT, otPK.DP.ECKA and otPK.EUICC.ECKA using the SK.DP.ECDSA.
- Optionally, secures the Profile Protection Keys (PPK).
- Prepare the Bound Profile Package according to the description given in section 2.5.4.

15. The SM-DP+ responds back to the LPA with the TransactionID, the Profile Metadata and the Bound Profile Package.
16. The LPA SHALL transfer the “ES8+.InitialiseSecureChannel” function call included in the Bound Profile Package to the eUICC by repeatedly calling the “ES10b.LoadBoundProfilePackage” function. The input data of the “ES8+.InitialiseSecureChannel” function includes the CRT, otPK.DP.ECKA, and smdpSign2.
17. The eUICC SHALL verify the smdpSign2 with the PK.DP.ECDSA. If the verification succeeds, the eUICC SHALL generate Session Keys using the input data received in step 16.
18. The LPA SHALL transfer the “ES8+.ConfigureISDP” function call included in the Bound Profile Package to the eUICC by repeatedly calling the “ES10b.LoadBoundProfilePackage” function.
19. The LPA SHALL transfer the “ES8+.StoreMetadata” function call included in the Bound Profile Package to the eUICC by repeatedly calling the “ES10b.LoadBoundProfilePackage” function.
20. If the Profile Protection Keys (PPK) were included in the Bound Profile Package received in the step 15, the LPA SHALL transfer the “ES8+.ReplaceSessionKeys” function call to the eUICC by repeatedly calling the “ES10b.LoadBoundProfilePackage” function. The input data of the “ES8+.ReplaceSessionKeys” function includes the PPK. Once the eUICC receives “ES8+.ReplaceSessionKeys” function call, it SHALL decrypt the Profile Protection Keys and replace the current SCP03t Session Keys with the decrypted Profile Protection Keys.
21. The LPA SHALL transfer the Profile Elements included in the "ES8+.LoadProfileElements" functions by repeatedly calling the “ES10b.LoadBoundProfilePackage” function.
    If all the Profile Elements are successfully processed and installed, with or without any warning, the last response of the “ES10b.LoadBoundProfilePackage” function SHALL deliver the Profile Installation Result (that contains the Profile Installation Receipt with a resultCode indicating final success indication, the TransactionID, SM-DP+ Address, and the eUICC_Sign2 which is generated across the ResultCode, TransactionID, and SM-DP+ Address using SK.EUICC.ECDSA).
    Otherwise, during the transfer and processing of the Profile Elements, if a fatal error occurs, the eUICC SHALL stop the procedure and SHALL report to the LPA with a response of the “ES10b.LoadBoundProfilePackage” function including the Profile Installation Result.
    Before delivering the Profile Installation Result to the LPA, the eUICC SHALL store them in its non-volatile memory.

22. The LPA calls the “ES9+.HandleProfileInstallationResult " function with input data including the profile Installation Result which was given by the eUICC in the step 20.
23. Optional: The SM-DP+ call the “ES2+.HandleProfileInstallationResult” with input data including eid, iccid, ProfileType, completionTimestamp, resultCode and operationStatus.
24. If the LPA receives an Acknowledge message from the SM-DP+ in the step 21, then the LPA SHALL call “ES10b.DeleteProfileInstallationResult”.
25. The eUICC SHALL delete the Profile Installation Result from its non-volatile memory.

3.1.3 Limitation for Profile Installation

This Limitation is for Phase 1 ONLY.

The number of Profiles installed on an eUICC SHALL be restricted to 1.

If there is a Profile already installed in the eUICC and if the LPA initiates a download operation then eUICC SHALL reject the operation.

3.1.4 Error Handling Within the Profile Download Procedure

The Profile download and installation procedure comprises a sequence of operations between the SM-DP+, the LPA, and the eUICC over a period of time. In addition to errors reported by ES9+ and ES10b functions, other conditions MAY impact the successful execution of this procedure. The LPA SHOULD indicate such failures to the user; however, the specific presentation of these errors is out of the scope of this document.

The LPA SHOULD NOT initiate a new Profile download and installation procedure while there is an active download session. However, in the event that this does occur, the eUICC SHALL discard its session state (including any downloaded metadata, Profile contents, and Profile Installation Report) when a new session is started with ES10b.GetEUICCChallenge.

If an eUICC memory reset is initiated during a Profile download, the download will be terminated.

The Profile download and installation procedure MAY fail because of a communications failure between the LPA and the SM-DP+. The LPA MAY retry for a period of time. The LPA SHALL reset its own Profile download session state when all retry attempts have failed.

The Profile download and installation procedure could fail while the LPA is sending BPP TLVs to the eUICC using ES10b.LoadBoundProfilePackage for reasons other than an error status reported by the eUICC. Examples of such failures during the download process include:

- In the case of a removable eUICC card, the End User could remove the card.
- The End User could switch off the power or remove the battery.
- A software fault could cause a crash of the LPA, host Device, and/or baseband processor.

The LPA SHOULD provide an appropriate error indication to the End User when possible (e.g., when power is restored). The specific presentation of such an error notification is out of scope of this document.

3.2 Local Profile Management

3.2.1 Enable Profile

Normal Case:

This procedure is used to enable a Profile already downloaded and installed on an eUICC. The procedure is initiated by the End-user using the LUI of the LPA. It SHOULD be noted that in contrast to remote controlled use-cases as defined in SGP.02 [2] no post processing such as fall-back and server notifications are required as this control scheme is given to the end-user: when LPA exists, eUICC SHALL ignore fall-back attribute setting.

NOTE: This sequence illustrates the enablement of a Profile with a single call of the "ES10c.EnableProfile" function as an example. Another way to achieve the same result would be for the LUI to call first the "ES10c.DisableProfile" function and then the "ES10c.EnableProfile". That's a matter of implementation choice from the LUI provider.

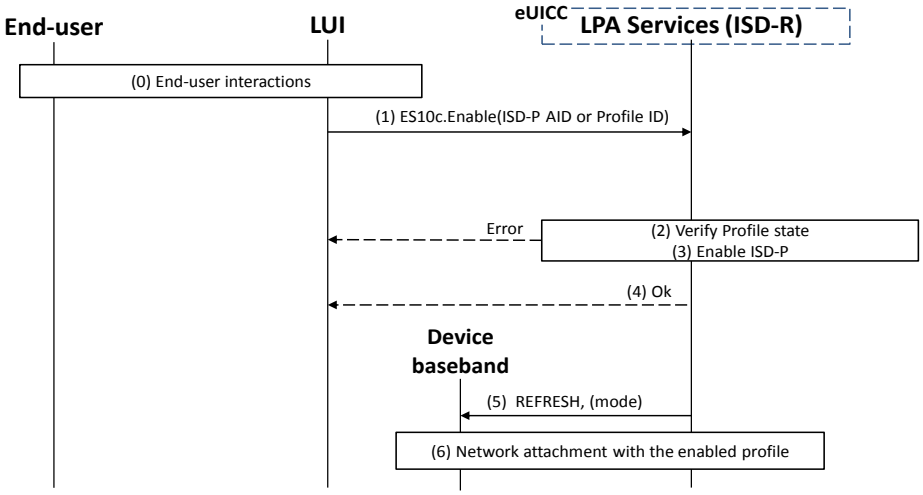

Figure 8: Enable Profile

Start Conditions:

- The LPA is authenticated to the eUICC as legitimate for performing Local Profile Management.
- End User Intent is verified as defined in SGP.21 [4] Simple Confirmation.

Procedure:

1. The End-user is presented a user interface that displays the list of installed Profiles, with their current states (Enabled or Disabled), within the eUICC as described in "List Profiles" procedure (see section 3.2.4) The End User selects the Profile to be enabled.
2. LPA SHALL call the "ES10c.EnableProfile" (see section 5.2.4.2) function of the ISD-R with its relevant input data.
3. ISD-R SHALL verify the state of the target Profile. If the target Profile is not in Disabled state, ISD-R SHALL return a response indicating a failure, and the procedure SHALL end.

4. The ISD-R SHALL disable the currently Enabled Profile (if any) and then enable the targeted Profile. This SHALL be processed as an atomic operation by the eUICC. In case of any error, the eUICC SHALL recover the previous Profile state, ISD-R SHALL return a response indicating a failure, and the procedure SHALL end.
5. The ISD-R SHALL return a response indicating result OK back to the LUI.

NOTE: At this stage only the state of the ISD-P(s) is (are) changed. The change will be effective only when the Device will have taken into account this change of Enabled Profile (any Device cache refreshing, network attachment with the new Enabled Profile…) triggered by the execution of the REFRESH command.

6. The ISD-R SHALL send a REFRESH proactive command to trigger the execution of a network attach procedure with the newly Enabled Profile.

NOTE: An additional REFRESH mode for the specific case of eUICC called “Profile changed” mode may be use.

NOTE: In case of any error after this step, indicating that the currently Enabled Profile cannot provide connectivity, there SHALL not be any fall-back to the previously Enabled Profile. This action SHALL remain under the responsibility of the End-user.

3.2.2 Disable Profile

Normal Case:

This procedure is used to disable an Enabled Profile already downloaded and installed on an eUICC. The procedure is initiated by the End-user using the LUI of the LPA. It SHOULD be noted that in contrast to remote controlled use-cases as defined in SGP.02 [2] no post processing such as fall-back and server notifications are required as this control scheme is given to the end-user.

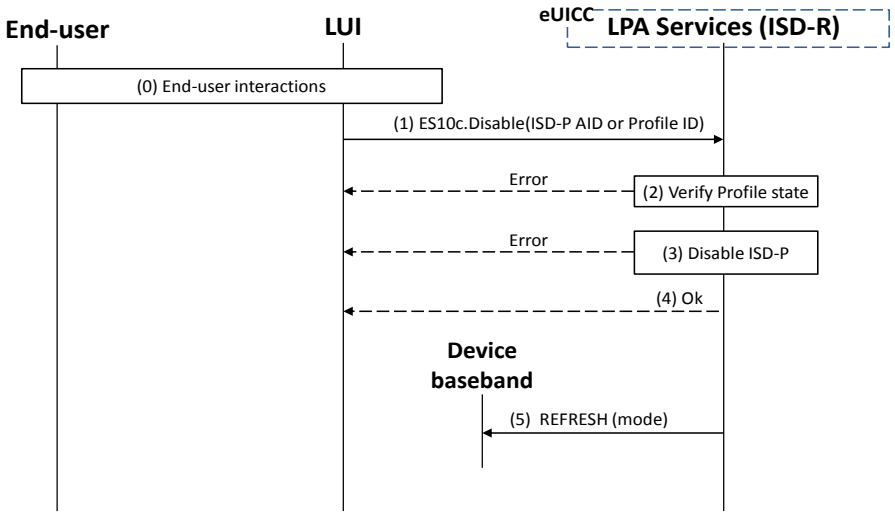

Figure 9: Disable Profile

Start Conditions:

- The LPA is authenticated to the eUICC as legitimate for performing Local Profile Management.
- End User Intent is verified as defined in SGP.21 [4] Simple Confirmation.

Procedure:

0. The End-user is presented a user interface that displays the list of installed Profiles, with their current states (Enabled or Disabled), within the eUICC as described in “List Profiles” procedure (see section 3.2.4) The End User selects the Profile to be Disabled. (1) LPA SHALL call the “ES10c.DisableProfile” (see section 5.2.4.3) function of the ISD-R with its relevant input data.
7. ISD-R SHALL verify the state of the target Profile. If the target Profile is not in Enabled state, ISD-R SHALL return a response indicating a failure, and the procedure SHALL end.
8. The ISD-R SHALL disable the currently Enabled Profile.
9. The ISD-R SHALL return a response indicating result OK back to the LUI.
10. The ISD-R SHALL send a REFRESH proactive command to instruct the Device of the change of the Enabled Profile (see section 3.2.1).

3.2.3 Delete Profile

This procedure is used to delete a Profile already downloaded and installed on an eUICC. The procedure is initiated by the End-user using the LUI of the LPA.

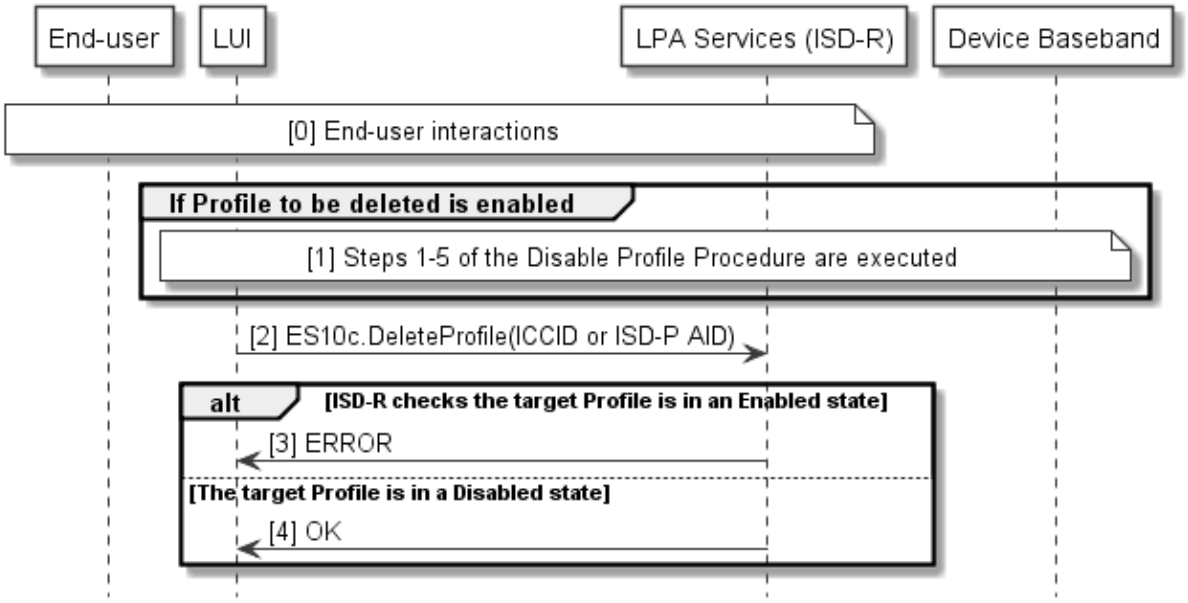

Figure 10: Delete Profile

Start Conditions:

- The LPA is authenticated to the eUICC as legitimate for performing Local Profile Management.
- End User Intent is verified as defined in SGP.21 [4] Authenticated Confirmation.

Procedure:

0. The End-user is presented a user interface that displays the list of installed Profiles, with their current states (Enabled or Disabled), within the eUICC as described in “List Profiles” procedure (see section 3.2.4**Error! Reference source not found.**) The End User selects the Profile to be deleted and acknowledges the consequences.
1. If the identified Profile to be deleted is Enabled then steps 1-5 of the disable profile procedure SHALL be executed as defined in section 3.2.2.
2. The LPA SHALL call the ES10c.DeleteProfile function of the ISD-R with its relevant input data.
3. ISD-R SHALL verify the state of the target Profile. If the target Profile is in an Enabled state, ISD-R SHALL return a response indicating a failure, and the procedure SHALL end.
4. The ISD-R SHALL return a response indicating result OK back to the LPA.

3.2.4 List Profiles

This procedure is used by the LPA to list the Profiles and their current state pre-installed or previously downloaded and installed on an eUICC in human readable format. The procedure is initiated by the LPA either implicitly (e.g. at 1st Device boot up) or explicitly (e.g. through LUI user interface options).

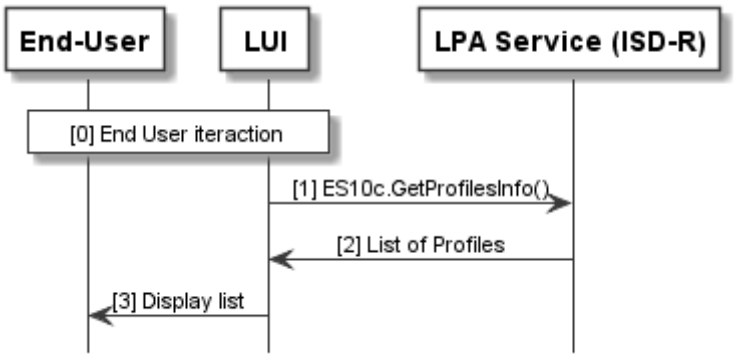

Figure 11: List Profile

Start Conditions:

- The LPA is authenticated to the eUICC as legitimate for performing Local Profile Management.
- End User Intent is verified as defined in SGP.21 [4] Simple Confirmation.

Procedure:

0. The LPA is started on the Device. The user MAY be presented with the user interface options.
1. Either as part of the LPA launch procedure or through explicit user menu selection, the LPA SHALL call ES10c.GetProfilesInfo to request the list of Profiles from the LPA Services.
2. The eUICC SHALL return with data that contains Profile metadata and status of the Profile(s) as defined in section (5.2.4.1).
3. The LPA SHALL display the list of installed Profiles along with their current states (Enabled or Disabled) to the End User in human readable format. If there are no Profiles installed on the eUICC, then the Profile list on LPA SHALL remain empty.

End Conditions:

Any Profile information presented to the user SHALL always be in human readable format.

3.2.5 Add Profile

This procedure will allow the End User to add a single Profile. This procedure will not enable the downloaded Profile, nor disable an Enabled Profile. Network connectivity is assumed. The download can be initiated by the input of an Activation Code.

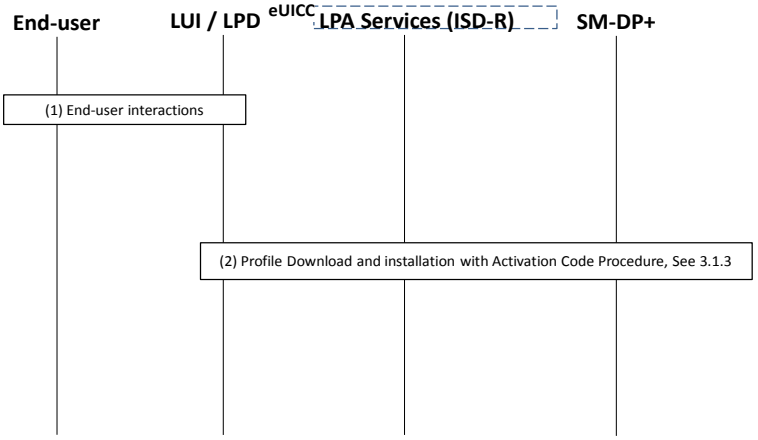

Figure 12: Add Profile

Start conditions:

- The LPA is authenticated to the eUICC as legitimate for performing Local Profile Management.
- The Download of a new Profile is allowed on the eUICC.
- End User Intent is verified as defined in SGP.21 [4] Authenticated Confirmation.

Procedure:

1. The End User initiates the Add Profile operation through explicit user menu selection within the LUI and the input of the Activation Code.
2. A Profile is downloaded via the Profile Download and Installation Procedure with Activation Code as defined in (4.1).

End conditions:

11. The Profile has been installed on the End User's Device.
12. The Profile Metadata within the eUICC is updated with the Profile Metadata from the Installed Profile.

3.2.6 Add/Update Profile Nickname

This procedure is used to add or change a nickname to a Profile already downloaded and installed on an eUICC. The procedure is initiated by the End-user using the LUI of the LPA.

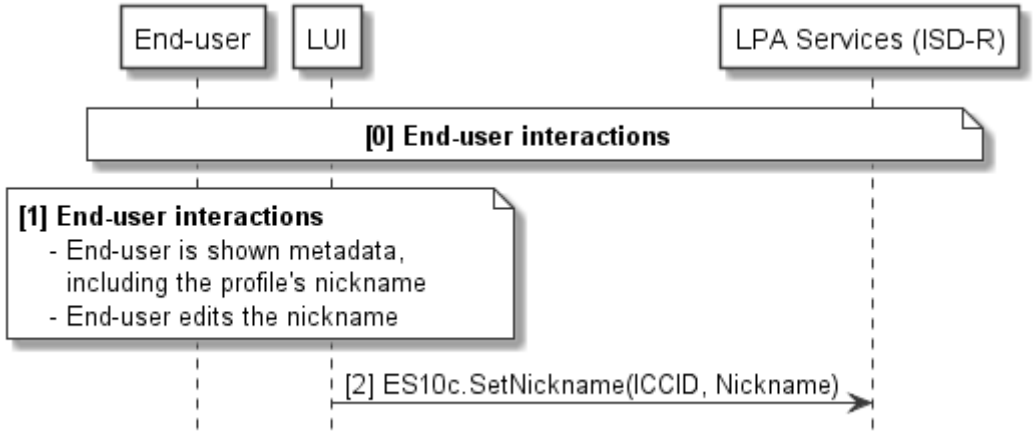

Figure 13: Add/Update Profile Nickname

Start Conditions:

- The LPA is authenticated to the eUICC as legitimate for performing Local Profile Management.
- End User Intent is verified as defined in SGP.21 [4] Simple Confirmation.

Procedure:

4. The End-user is presented a user interface that displays the list of installed Profiles, with their current states (Enabled or Disabled), within the eUICC as described in “List Profiles” procedure (see section 3.2.4) The End User selects the Profile to be modified.
13. Through the LUI, the End-user:

    a) Is shown the relevant metadata, including the associated nickname (if any), along with the Profile status.
    b) Edits the Nickname.

14. The LUI calls the function “ES10c.SetNickname” with the relevant ICCID and edited Nickname.

End Conditions:

The new Nickname is stored in the metadata of the relevant Profile.

3.3 Local eUICC Management

3.3.1 Retrieve EID

The EID SHALL be made available for the End User on the user interface by the LPA. The EID is retrieved by the LPA over the ES10c interface using the function ES10c.GetEID as described in section (5.2.4.6).

3.3.2 eUICC Memory Reset

This procedure is used to delete all the Profiles and their associated Profile Metadata stored on the eUICC regardless of their status. The procedure is initiated by the End User using the LUI of the LPA.

```
@startuml
skinparam monochrome true
skinparam ArrowColor Black

hide footbox

participant "End User" as EndUser #FFFFFF
participant "LUI" as LPA #FFFFFF
participant "LPA Services (ISD-R)" as LPAServices #FFFFFF
participant "Device Baseband" as Baseband #FFFFFF

note over EndUser, LPA #FFFFFF
     [0] End User interactions
end note
LPA -> LPAServices: [1] ES10c.eUICCMemoryReset

note over LPAServices #FFFFFF
     [2] Delete all the ISD-P with its Profile
     and associated Profile Metadata
end note

LPAServices --> LPA : [3] OK

LPAServices --> Baseband : [4] [REFRESH (mode)]


@enduml
```

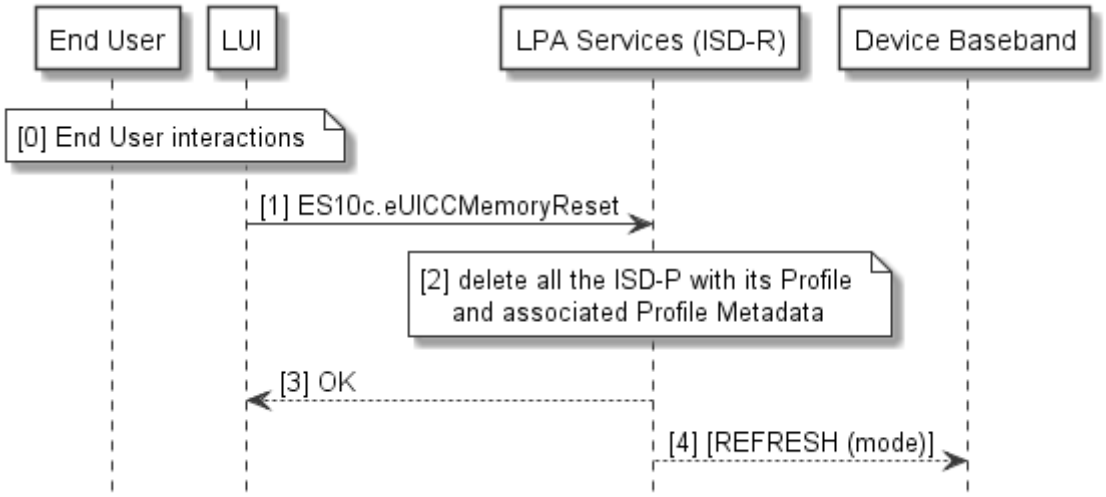

Figure 14: eUICC Memory Reset

Start Conditions:

- The LPA is authenticated to the eUICC as legitimate for performing Local Profile Management.
- End User Intent is verified as defined in SGP.21 [4] Authenticated Confirmation.

Procedure:

0. The End User initiates the eUICC Memory Reset and acknowledges the consequences.
1. The LPA SHALL call the “ES10c.eUICCMemoryReset” function of the ISD-R with its relevant input data.
2. The ISD-R SHALL delete all the ISD-Ps with their Profiles and their associated data and Profile Metadata.
3. The ISD-R SHALL return a response indicating result OK back to the LUI.
4. If there was an Enabled Profile, the ISD-R SHALL send a REFRESH proactive command to instruct the Device.

End conditions:

The Profiles and their associated Profile Metadata are deleted from the eUICC.

3.4 Device and eUICC initialisation

3.4.1 eUICC initialisation

The eUICC SHALL indicate its support of eUICC functionality in ATR Global Interface byte as defined in ETSI TS 102 221 [6]. If the indication is received by the LPA, the LPA MAY obtain additional eUICC information, such as SVN.

The eUICC initialisation SHALL follow the procedure as defined in ETSI TS 102 221 [6]. If the eUICC contains an Enabled Profile, the eUICC initialisation procedure SHALL be complete.

3.4.2 RSP Terminal Services

The Device SHALL report its support of LPA functions using the Terminal Capability command data defined in ETSI TS 102 221 [6].

Within the Terminal Capability template (tag 'A9'), the tag ‘83’ is used for indicating the Device’s support for eUICC related functions.

The LPA support is indicated in the first byte within the TLV object under tag '83':

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
-	-	-	-	-	-	-	1	Local User Interface (LUI) for eUICC supported
-	-	-	-	-	-	-	0	Local User Interface (LUI) for eUICC not supported
-	-	-	-	-	-	1	-	Local Profile Download (LPD) for eUICC supported
-	-	-	-	-	-	0	-	Local Profile Download (LPD) for eUICC not supported
x	x	x	x	x	x	-	-	RFU

Table 7: RSP Terminal Services objects

Subsequent bytes are RFU.

3.4.3 eUICC file structure

If there is no Enabled Profile on the eUICC, the eUICC SHALL ensure a default file system is available to the Device. This file system SHALL contain at least the MF and MAY contain the MF-level EFs shown below.

- EFENV-CLASSES
- EFUMPC

These 2 files SHALL never be present in any Profile.

When a Profile is enabled, the eUICC SHALL present a file system comprising the Profile file system and the EFs listed above if existing.

4 Data Elements

4.1 Activation Code

The Activation Code SHALL be coded to be the concatenation of the following strings listed in the following table:

Name	MOC	Description
AC_Format	M	Format of the Activation Code. SHALL be set to “1” for this Format of the Activation Code and any subsequent backward compatible Format
Delimiter	M	SHALL be set to “$”
SM-DP+ Address	M	FQDN (Fully Qualified Domain Name) of the SM-DP+ (e.g., smdp.gsma.com) restricted to the Alphanumeric mode character set defined in table 5 of ISO/IEC 18004 [15]
Delimiter	M	SHALL be set to “$”
AC_Token	M	MatchingID as described in section (4.1.1)
Delimiter	C	SHALL be present and set to “$” if any of the following optional parameters is present
SMDPid	O	Subject Identifier of the SM-DP+ certificate
Delimiter	C	SHALL be present and set to “$” if any of the following optional parameters is present
Confirmation Code Required Flag	O	SHALL be present and set to “1” if Confirmation Code is required; else it SHALL be absent

Table 8: Activation Code Structure

The maximum length of the Activation Code SHALL be 255 characters, but in practise it is recommended to consider the user experience when choosing the length.

A Phase 1 Device SHALL ignore a delimiter and any further parameters beyond the “Confirmation Code Required Flag”.

If a Phase 1 Device encounters a AC_Format other than “1” the Activation Code SHALL be treated as invalid.

Examples of the Activation Code are as follows:

- 1$SMDP.GSMA.COM $04386-AGYFT-A74Y8-3F815
  (if SMDPid and Confirmation Code required flag are not present)

- 1$ SMDP.GSMA.COM $04386-AGYFT-A74Y8-3F815$$1
  (if SMDPid is not present and Confirmation Code required flag is present)
- 1$ SMDP.GSMA.COM $04386-AGYFT-A74Y8-3F815$1.3.6.1.4.1.31746$1
  (if SMDPid Confirmation Code required flag are present)
- 1$ SMDP.GSMA.COM $04386-AGYFT-A74Y8-3F815$1.3.6.1.4.1.31746
  (If SMDPid is present and Confirmation Code required flag is not present)
- 1$ SMDP.GSMA.COM $$1.3.6.1.4.1.31746
  (If SMDPid is present, Activation token is left blank and Confirmation Code required flag is not present)

When entered manually, the Activation Code SHALL be used as defined above.

When provided in a QR code according to ISO/IEC 18004 [15], the Activation Code SHALL be prefixed with "LPA:"

4.1.1 Matching ID

The MatchingID is a mandatory information (but MAY be zero-length) that SHALL be set-up between the Operator and the SM-DP, to identify the context of a specific management order given to the SM-DP+. The MatchingID is generated during the Download Initiation procedure, (see section 3.1.2).

The MatchingID is equivalent to the “Activation Code Token” as defined is SGP.21 [4].

The format and content of the MatchingID is subject to the following constraints:

It SHALL be a unique context identifier in the perimeter of the Operator and the SM-DP+ to:

- Match a download order initiated by the Operator with a Profile Download request coming from an LPD.
- As a protection for the SM-DP+: SM-DP+ SHALL only process requests containing a MatchingID known to the SM-DP+ (and therefore inherently valid).

It SHALL consist only of upper case alphanumeric characters (0-9, A-Z) and the “-“in any combination.

NOTE: This selection allows more compact alphanumeric QR code encoding and is expected to be supported for manual entry.

4.2 Device Information

During the Profile Download and Installation procedure, any Device Information provided by the LPA to the eUICC SHALL be signed by the eUICC, and then provided by the eUICC to the SM-DP+ for the purpose of Device eligibility check. The SM-DP+/Operator is free to use or ignore this information at their discretion.

Device Information includes:

- Device type allocation code: TAC
- Device capabilities: The Device SHALL set all the capabilities it supports.
  - Supported radio access technologies, including release.

- Contactless: support of the SWP and HCI interfaces as well as the associated APIs.

Device Information

DeviceInfo is coded using ASN.1 DER as following:

```
DeviceInfo ::= SEQUENCE {
   tac Octet8,
   deviceCapabilities DeviceCapabilities
}

DeviceCapabilities ::= SEQUENCE { -- Highest supported release for each definition
  -- The device SHALL set all the capabilities it supports
   gsmSupportedRelease Octet1 OPTIONAL,
   utranSupportedRelease Octet1 OPTIONAL,
   cdma2000onexSupportedRelease Octet1 OPTIONAL,
   cdma2000hrpdSupportedRelease Octet1 OPTIONAL,
   cdma2000ehrpdSupportedRelease Octet1 OPTIONAL,
   eutranSupportedRelease Octet1 OPTIONAL,
   contactlessSupportedRelease Octet1 OPTIONAL
}
```

4.3 eUICC Information

During Profile the Download and installation procedure, eUICC information needs to be provided to the LPA and forwarded to SM-DP+. eUICC information SHALL be generated by the eUICC and is intended to be signed:

- EID prefix: First 9 bytes of EID as defined in SGP.02 [2].
- Profile Package Version: Indicates the highest version number of the “SIMalliance eUICC Profile Package: Interoperable Format Technical Specification” [5] supported by the eUICC.
- SVN: Indicates the highest Specification Version Number of the “SGP.22 Remote Provisioning Architecture for Embedded UICC Technical Specification for consumer Devices” supported by the eUICC [This document].
- Firmware version: indicates the current version information of the eUICC’s platform and the OS, defined as for the EID in SGP.02 [2]. This value is issuer specific.
- Available amount of Non-volatile memory: Indicates the current total available memory (whatever the underlying technology, flash or eeprom) for Profile download and installation. The value is expressed in Bytes.
- eUICC capabilities: Contains the capabilities supported by the eUICC.
- eUICC Certificate Info describing the supported certificate formats.
- Supported curves for signature creation.
- Supported curves for signature verification.

The eUICC information comprising eUICCInfo1 and eUICCInfo2 are defined in Annex H.

4.4 Profile Metadata

During Profile Download and installation procedure, Profile Metadata needs to be provided to the LPA for display and to the eUICC. Profile Metadata are generated by the SM-DP+ in plain text to be readable by the LPA. Profile Metadata are also contained protected in BPP to be loaded into the eUICC, so that the LPA will be able to access the same information any

time after the Profile has been successfully loaded into the eUICC, using the ES10c.GetProfilesInfoMetadata function.

Profile Metadata values, like any other Profile data, are under the responsibility of, and defined by, the Profile owner. Profile Metadata is communicated to the SM-DP+ by means which are out of scope of this specification.

Profile Metadata includes:

- ICCID of the Profile
- Profile Name (corresponds to “Short description” in SGP.21 [4]) as a plain text information: content free information defined by the Operator/Service Provider
- Operator/Service Provider name, as a plain text information: content free information defined by the Service Provider (e.g. ‘Orange’, ‘AT&T’…)
- End user’s Profile Nickname
- Icon

Profile Metadata are merely to be displayed to the End-user to provide information about the Profile to be installed. But it is out of scope of this implementation what LPA does exactly with these Profile Metadata, e.g. LPA can display all or only part of this information.

4.5 Keys and Certificates

This section describes Keys and Certificates used in this specifications.

4.5.1 Cryptographic Keys

Key name	Key name	Nature	Length
PK.EUICC.ECDSA	Public key of the eUICC used to verify a eUICC signature. This key is included inside the eUICC Certificate CERT.EUICC.ECDSA.	ECC	See below
SK.EUICC.ECDSA	Private key of the eUICC used to provide signatures. This key is loaded inside ECASD.	ECC	See below
PK.DP.ECDSA	Public key of the SM-DP+ used to verify a SM-DP+ signature. This key is included inside the SM-DP+ certificate CERT.DP.ECDSA.	ECC	See below
SK.DP.ECDSA	Private key of the SM-DP+ used to provide signatures.	ECC	See below
PK.EUM.ECDSA	Public key of the EUM used for verifying EUICC Certificates. This key is included inside the EUM Certificate CERT.EUM.ECDSA.	ECC	See below
SK.EUM.ECDSA	Private key of the EUM used for signing EUICC Certificates.	ECC	See below
PK.CI.ECDSA	Public key of the CI used for verifying EUM and SM-DP+ certificates.	ECC	See below

SK.CI.ECDSA	Private key of the CI used for signing EUM and SM-DP+ certificates.	ECC	See below
otPK.EUICC.ECKA	One-time public key of the EUICC used for key agreement.	ECC	See below
otSK.EUICC.ECKA	One-time private key of the EUICC used for key agreement.	ECC	See below
otPK.DP.ECKA	One-time public key of the SM-DP+ used for key agreement.	ECC	See below
otSK.DP.ECKA	One-time private key of the SM-DP+ used for key agreement.	ECC	See below

Table 9: Cryptographic Keys

All ECC keys used within the same secure channel SHALL reference the same curve parameters. Thus all keys have the same length.

This specification supports the three following curves (similar as SGP.02 [2]):

- NIST P-256, defined in Digital Signature Standard [29] (recommended by NIST)
- brainpoolP256r1, defined in RFC 5639 [18] (recommended by BSI)
- FRP256V1, defined in ANSSI ECC [20] (recommended by ANSSI)

An eUICC SHALL have at least one set of elliptic curve parameters preloaded by the EUM during eUICC manufacturing.

Capabilities of each party are exchanged during the Profile download and installation procedure. The SM-DP+ SHALL select the most suitable curve supported by both eUICC and SM-DP+.

In the event that no common curve is supported by both parties then the procedure SHALL stop.

4.5.2 Certificates

Certificate Issuer issues certificates for Remote SIM Provisioning system entities and acts as a trusted root for the purpose of authentication of the entities of the system.

The following certificates SHALL be signed and issued by a GSMA CI:

- GSMA CI Certificate (CERT.CI.ECDSA)
- EUM Certificates (CERT.EUM.ECDSA)
- SM-DP+ Certificate (CERT.DP.ECDSA)

The following certificate SHALL be signed and issued by the EUM:

- eUICC Certificate (CERT.EUICC.ECDSA)

The following generic certificate SHALL be signed and issued by a globally trusted root CA:

- SM-DP+ TLS Certificate (CERT.DP.TLS) stored in the SM-DP+

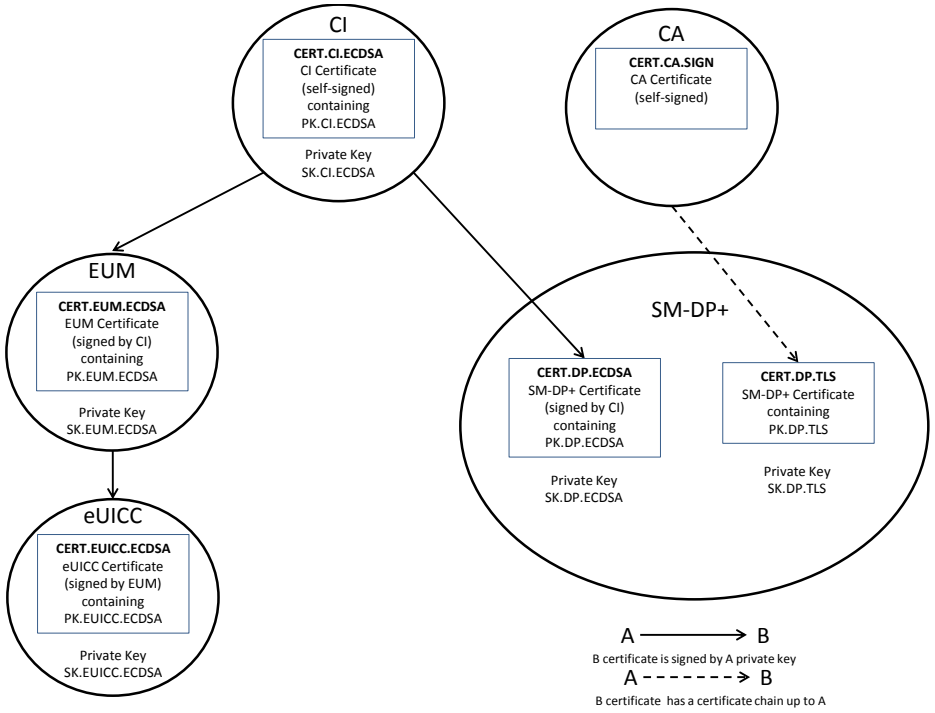

Figure 15: Certificate Chains

The Certificates CERT.CI.ECDSA, CERT.EUICC.ECDSA, CERT.EUM.ECDSA, CERT.DP.ECDSA and CERT.DP.TLS exchanged over ES9+, ES10b and ES8+ are described in the next sections.

4.5.2.1 GP Certificate profile

This section describes the GlobalPlatform Certificate profile. Those Certificates SHALL follow GlobalPlatform Card Specification [8], with the specific coding given in this section.

Only these Certificates MAY follow GP Certificate profile:

- CERT.EUICC.ECDSA
- CERT.EUM.ECDSA
- CERT.DP.ECDSA

Tag	Length	Value Description			MOC
'7F21'	Var	Certificate data object			M
		Tag	**Length**	**Value Description**	**MOC**
		'93'	1-16	Certificate Serial Number	M
		'42'	1-16	Issuer identifier: EUM OID	M
		'5F20'	1-16	Subject Identifier: eUICC Identifier (aka EID)	M
		'95'	1	Key Usage ‘82’: Signature verification	M
		'5F25'	4	Effective Date (YYYYMMDD, BCD format)	O

		'5F24'	4	Expiration Date (YYYYMMDD, BCD format) There is no life time defined. It is recommended to set a value representing this condition using the value defined in RFC 5280 [17]: 99991231	M
		'45'	1-16	ECASD Image Number	M
		'73'	Var	Discretionary Data: 'C2' var eUICC Extended GSMA SAS Accreditation Serial Number 'C9' 1-16 Authority Key Identifier of the EUM other TLVs may follow	M
		'7F49'	Var	Public Key	M
		'5F37'	Var	Signature (to be computed as described in GlobalPlatform Card Specification Amendment E [AmdE] and the signature SHALL include all the field starting from tag '93' to tag '7F49')	M

Table 10: CERT.EUICC.ECDSA

Tag	Length	Value Description			MOC
'7F21'	Var	Certificate data object			M
		Tag	Length	Value Description	MOC
		'93'	1-16	Certificate Serial Number	M
		'42'	1-16	Issuer identifier: GSMA CI OID	M
		'5F20'	1-16	Subject Identifier: EUM OID	M
		'95'	1	Key Usage '82': Signature verification	M
		'5F25'	4	Effective Date (YYYYMMDD, BCD format)	O
		'5F24'	4	Expiration Date (YYYYMMDD, BCD format)	M
		'73'	Var	Discretionary Data: 'C8 01 03': EUM Certificate 'C9' 1-16 Authority Key Identifier of the GSMA CI 'CA' 1-16 Subject Key Identifier other TLVs MAY follow	M
		'7F49'	Var	Public Key	M
		'5F37'	Var	Signature (to be computed as described in GlobalPlatform Card Specification Amendment E [12] and the signature SHALL include all the field starting from tag '93' to tag '7F49')	M

Table 11: CERT.EUM.ECDSA

Tag	Length	Value Description	MOC
'7F21'	Var	Certificate data object	M

		Tag	Length	Value Description	MOC
		'93'	1-16	Certificate Serial Number	M
		'42'	1-16	Issuer identifier: GSMA CI OID	M
		'5F20'	1-16	Subject Identifier: SM-DP+ OID	M
		'95'	1	Key Usage ‘82’: Signature verification	M
		'5F25'	4	Effective Date (YYYYMMDD, BCD format)	O
		'5F24'	4	Expiration Date (YYYYMMDD, BCD format)	M
		'73'	Var	Discretionary Data 'C8 01 01': SM-DP+ certificate for profile binding 'C9' 1-16 Authority Key Identifier of the GSMA CI other TLVs MAY follow	M
		'7F49'	Var	Public Key	M
		'5F37'	Var	Signature (to be computed as described in GlobalPlatform Card Specification Amendment E [12] and the signature SHALL include all the field starting from tag '93' to tag '7F49')	M

Table 12: CERT.DP.ECDSA

The tag 'C9' identifies the issuer authority's public key associated to the private key used for creating the Certificate signature. This tag SHALL be present.

The tag 'CA' identifies the public key bound in the Certificate. This information, jointly with the Authority Key Identifier, is used to verify the Certificate trust chain.

The public key data object (‘7F49’) contains an elliptic curves (EC) public key and the corresponding domain parameters as defined in **Error! Reference source not found.**.

Tag	Length	Value Description			MOC
'7F49'	Var	Public Key Data Object			M
		Tag	Length	Value Description	MOC
		'B0'	Var	Public key – Q	M
		'F0'	'01'	Key Parameter Reference: ‘00’: NIST P-256 [29] ‘03’: brainpoolP256r1 [18] ‘40’: FRP256V1 [20]	M

Table 13: Public Key Data Object Data Field

4.5.2.2 X.509 Certificate profile

This section describes the X.509 Certificate profile. Those Certificates SHALL follow RFC 5280 [17], with the specific coding given in this section.

In particular:

- 'Issuer' and 'Subject' fields SHALL be limited to standard attributes defined in ITU-T X.520 [24] and RFC 4519 [28].
- Certificates SHALL contain all 'critical' marked extensions defined in their respective profile; non-critical extension MAY be included but will be ignored by the using system.

These Certificates MAY follow X.509 Certificate profile:

- CERT.EUICC.ECDSA
- CERT.EUM.ECDSA
- CERT.DP.ECDSA

These Certificates SHALL follow X.509 Certificate profile:

- CERT.CI.ECDSA

NOTE: Certificates are described using table representation for easiness, but conform to the ASN.1 format given in RFC 5280 [17].

Field	Value Description	
tbsCertificate	Data to be signed	
	Field	**Value Description**
	version	Version SHALL be 3 (value is 2) as extensions are used in this specification.
	serialNumber	Certificate serial number
	signature	Contains the algorithm identifier and parameters used by the issuer to compute the value of the field 'signatureValue'. See section 4.5.2.2.1 NOTE: The algorithm identifier value and parameters values SHALL be the same as the one of the field ‘signatureAlgorithm’.
	issuer	This SHALL be identical to 'subject' field value.
	validity	Validity period of the Certificate. Period where the CI is allowed to issue Certificates.
	subject	Distinguished Name of the CI. Example of CI DN: cn = Symantec Class 3 Public Primary Certification Authority - G4 ou = Symantec Trust Network o = Symantec Corporation c = US
	subjectPublicKeyInfo	Contains the algorithm identifier, parameters and public key value. Algorithm identifier and parameters SHALL be set according to section 4.5.2.2.1. subjectPublicKeyInfo.subjectPublicKey contains the public key value and SHALL be coded as defined in RFC 5480 [27].

	extensions	Extension for Subject Key Identifier (see RFC 5280 [17]): section 4.2.1.2): extnID = id-ce- subjectKeyIdentifier critical = true extnValue = Public Key Identifier
		Extension for Key usage (see RFC 5280 [17] section 4.2.1.3): extnID = id-ce-keyUsage critical = true extnValue = { keyCertSign (5), --[Mandatory] cRLSign(6) –[Optional] }
		Extension for Certificate Policies (see RFC 5280 [17] section 4.2.1.4): extnID = id-ce-certificatePolicies critical = true extnValue = id-rspRole-ci (see Annex H) To indicate the GSMA CI role.
		Extension for Basic Constraints (see RFC 5280 [17] section 4.2.1.9): extnID = id-ce- basicConstraints critical = true extnValue = { cA = true }
signatureAlgorithm	See section 4.5.2.2.1	
signatureValue	Signature computed accordingly to one of the possible algorithm as listed in section 4.5.2.2.1	

Table 14: CERT.CI ECDSA

NOTE: The CERT.CI.ECDSA is a self-signed certificate, there is no need to include the Extension for Authority Key Identifier.

Field	Value Description	
tbsCertificate	Data to be signed	
	Field	**Value Description**
	version	Version SHALL be 3 (value is 2) as extensions are used in this specification.
	serialNumber	Serial number that SHALL be unique for each Certificate issued with a given CERT.EUM.ECDSA.
	signature	Contains the algorithm identifier and parameters used by the EUM to compute the value of the field 'signatureValue'. Apply rules defined in Table 53.

	issuer	Distinguished Name of the EUM that has signed the EUICC Certificate. It SHALL match the ‘subject’ field of the EUM Certificate CERT.EUM.ECDSA.
	validity	Validity period of the Certificate. There is no life time defined for this Certificate. eUICC certificates never expire. Expiration Date to be set to 99991231235959Z as stated in RFC 5280 [17]
	subject	Distinguished Name of the EUICC. It shall include, at least, 'organization' and 'serialNumber' attributes. Others attributes MAY be included for information. It is RECOMMENDED that 'organization' attribute has the same value as the 'organization' attribute of the EUM. 'serialNumber' SHALL contain the EID as an hexadecimal PrintableString. Example of eUICC DN: o = CompanyName serialNumber = 8900112233445566778899AABBCCDDEE
	subjectPublicKeyInfo	Contains the algorithm identifier, parameters and public key value. Apply rules defined in Table 53.
	extensions	Extension for Authority Key Identifier (see RFC 5280 [17] section 4.2.1.1): extnID = id-ce- authorityKeyIdentifier critical = true extnValue = keyIdentifier [0] To identify the PK.EUM.ECDSA that has to be used to verify this Certificate.
		Extension for Subject Key Identifier (see RFC 5280 [17] section 4.2.1.2): extnID = id-ce- subjectKeyIdentifier critical = true extnValue = keyIdentifier [0] Contains the identifier of the PK.EUICC.ECDSA bound in this Certificate.
		Extension for Key usage (see RFC 5280 [17] section 4.2.1.3): extnID = id-ce-keyUsage critical = true extnValue = digitalSignature (0)
		Extension for Certificate Policies (see RFC 5280 [17] section 4.2.1.4): extnID = id-ce-certificatePolicies critical = true extnValue = id-rspRole-euicc (see Annex H) To indicate that this is an eUICC Certificate.

		Extension for SAS accreditation serial number (this extension is defined in section 4.5.2.2.2): extnID = id-rsp-sasSn critical = false extnValue = SAS accreditation serial number value This is an optional extension that SHALL be present if this eUICC has been delivered a SAS accreditation.
signatureAlgorithm	See section 4.5.2.2.1	
signatureValue	Signature computed accordingly to one of the possible algorithm as listed in section 4.5.2.2.1	

Table 15: CERT.EUICC.ECDSA

NOTE: The ECASD Image number is not mapped in the X.509 Profile.

Field	Value Description	
tbsCertificate	Data to be signed	
	Field	**Value Description**
	version	Version SHALL be 3 (value is 2) as extensions are used in this specification.
	serialNumber	Serial number that SHALL be unique for each Certificate issued with a given CERT.CI.ECDSA.
	signature	Contains the algorithm identifier and parameters used by the issuer to compute the value of the field 'signatureValue'. Apply rules defined in Table 53.
	issuer	Distinguished Name of the GSMA CI that has signed the EUM Certificate. Example of CI DN: cn = Company Name Class 3 Public Primary Certification Authority - G4 ou = Company Name Trust Network o = Company Name c = US
	validity	Validity period of the Certificate. Period where the EUM is allowed to issue eUICC Certificates.
	subject	Distinguished Name of the EUM. It SHALL include at least 'organization' and 'commonName' attributes. Example of EUM DN: c = UK l = London o = Company Name cn = Company Name CA e = admin.pki@companyname.com
	subjectPublicKeyInfo	Contains the algorithm identifier, parameters and public key value. Apply rules defined in Table 53.

	extensions	Extension for Authority Key Identifier (see RFC 5280 [17]): section 4.2.1.1): extnID = id-ce-authorityKeyIdentifier critical = true extnValue = keyIdentifier [0] To identify the PK.CI.ECDSA that has to be used to verify this Certificate.
		Extension for Subject Key Identifier (see RFC 5280 [17] section 4.2.1.2): extnID = id-ce- subjectKeyIdentifier critical = true extnValue = keyIdentifier [0] Contains the identifier of the PK.EUM.ECDSA bound in this Certificate.
		Extension for Key usage (see RFC 5280 [17] section 4.2.1.3): extnID = id-ce-keyUsage critical = true extnValue = extnValue = { keyCertSign (5), --[Mandatory] cRLSign(6) –[Optional] }
		Extension for Certificate Policies (see RFC 5280 [17] section 4.2.1.4): extnID = id-ce-certificatePolicies critical = true extnValue = id-rspRole-eum (see Annex H) To indicate that this is an EUM Certificate.
		Extension for subjectAltName (see RFC 5280 [17] section 4.2.1.6): extnID = id-ce-subjectAltName critical = false extnValue = { registeredID (8) = EUM OID }
		Extension for Basic Constraints (see RFC 5280 [17] section 4.2.1.9): extnID = id-ce- basicConstraints critical = true extnValue = { cA = true pathLenConstraint = 0 } To indicate that this Certificate is a sub-ca limited to issue only "leaf" Certificate for eUICC.
signatureAlgorithm	See section 4.5.2.2.1	
signatureValue	Signature computed accordingly to one of the possible algorithm as listed in section 4.5.2.2.1	

Table 16: CERT.EUM.ECDSA

Field	Value Description	
tbsCertificate	Data to be signed	
	Field	**Value Description**
	version	Version SHALL be 3 (value is 2) as extensions are used in this specification.
	serialNumber	Serial number that SHALL be unique for each Certificate issued with a given CERT.CI.ECDSA.
	signature	Contains the algorithm identifier and parameters used by the issuer to compute the value of the field 'signatureValue'. Apply rules defined in Table 53.
	issuer	Distinguished Name of the CA that has signed the SM-DP+ Certificate.
	validity	Validity period of the Certificate.
	subject	Distinguished Name of the SM-DP+. It SHALL include at least 'organization' and 'commonName' attributes. Example of SM-DP+ DN: c = UK l = London o = Company Name cn = Company Name e = dp@companyName.com
	subjectPublicKeyInfo	Contains the algorithm identifier, parameters and public key value. Apply rules defined in Table 53.
	extensions	Extension for Authority Key Identifier (see RFC 5280 [17] section 4.2.1.1): extnID = id-ce- authorityKeyIdentifier critical = true extnValue = keyIdentifier [0] To identify the PK.CI.ECDSA that has to be used to verify this Certificate.
		Extension for Subject Key Identifier (see RFC 5280 [17] section 4.2.1.2): extnID = id-ce- subjectKeyIdentifier critical = true extnValue = keyIdentifier [0] Contains the identifier of the PK.DP.ECDSA bound in this Certificate.

		Extension for Key usage (see RFC 5280 [17] section 4.2.1.3): extnID = id-ce-keyUsage critical = true extnValue = digitalSignature (0)
		Extension for Certificate Policies (see RFC 5280 [17] section 4.2.1.4): extnID = id-ce-certificatePolicies critical = true extnValue = id-rspRole-dp-pb (see Annex H) To indicate that this is an SM-DP+ Certificate for profile binding.
		Extension for subjectAltName (see RFC 5280 [17] section 4.2.1.6): extnID = id-ce-subjectAltName critical = false extnValue = { registeredID (8) = SM-DP+ OID }
signatureAlgorithm	See section 4.5.2.2.1	
signatureValue	Signature computed according to one of the possible algorithm as listed in section 4.5.2.2.1	

Table 17: CERT.DP.ECDSA

4.5.2.2.1 Algorithm identifiers and parameters

This section provides the values to be set in 'AlgorithmIdentifier.algorithm' and 'AlgorithmIdentifier.parameters' fields of the Certificate for each of the algorithms used in this specification.

For BrainpoolP256r1:

- 'AlgorithmIdentifier.algorithm' field SHALL be set to: "iso(1) identified-organization(3) teletrust(36) algorithm(3) signaturealgorithm(3) ecSign(2) ecStdCurvesAndGeneration (8) ellipticCurve(1) versionOne(1) brainpoolP256r1(7)" as defined in RFC 5639 [18]
- 'AlgorithmIdentifier.parameters' field SHALL be omitted.

For NIST P-256:

- 'AlgorithmIdentifier.algorithm' field SHALL be set to: "iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) ecdsa-with-SHA256(2)" as defined in RFC 5758 [25] and RFC 5759 [26]
- 'AlgorithmIdentifier.parameters' field SHALL be omitted as defined in RFC 5758 [25] section 3.2.

For FRP256V1:

- 'AlgorithmIdentifier.algorithm' field SHALL be set to: '' iso(1) member-body(2) fr(250) type-org(1) 223 101 256 1''.
- 'AlgorithmIdentifier.parameters' field SHALL be omitted.

4.5.2.2.2 Specific Certificate extensions

This section provides the definition of specific Certificate extensions used in this specification.

```
-- Definition of the extension for Extended GSMA SAS Accreditation Serial Number
id-rsp-sasSn OBJECT IDENTIFIER ::= { id-rspExt 0}
SasSn::= UTF8String (SIZE(0..64))
```

4.5.2.3 Certificate verification

Any of the Certificate described in section **Error! Reference source not found.** SHALL be verified according to the description given in this section.

If any of these verifications fail, the Certificate SHALL be considered as invalid and the operation where it was used SHALL be rejected.

The Certificate SHALL:

- have a valid signature.
- be signed by a GSMA CI, or a trusted chain up to a GSMA CI with the exception of CERT.DP.TLS which may be signed by any globally trusted root CA. Certificate Path validation SHALL follow the process defined in RFC 5280 [17], using the Subject Key Identifier ('CA' tag for GP Certificates) and Authority Key Identifier fields ('C9' tag for GP Certificates).
- not have been revoked, and no Certificate in the trust chain has been revoked.
- not have expired (It should be noted that eUICC and the Device might not have reliable access to the current time to perform this verification).
- have all the 'critical' extension defined for its profile.

In addition to those verifications, and specifically to those Certificates:

CERT.EUICC.ECDSA:

- (GP) The 'issuer identifier' field has the same value as the 'subject identifier' field of the CERT.EUM.ECDSA.
- (GP) The 'ECASD image number' has the value as defined in SGP.02 [2].
- (GP) The 'Key usage' field SHALL be set with value '82'.
- (GP) The discretionary data field SHALL contain the tag 'C9' set with an Authority Key Identifier.
- (X.509) The extension 'Key usage' SHALL be set with the value ''digitalSignature''.
- (X.509) The extension 'Certificate Policies' SHALL be set with the OID id-rspRole-euicc to indicate an eUICC Certificate.

CERT.EUM.ECDSA:

- (GP) The 'Key usage' field SHALL be set with value '82' (signature verification).

- (GP) The discretionary data field SHALL contain the tag 'C8' set with value "EUM Certificate".
- (GP) The discretionary data field SHALL contain the tag 'C9' set with an Authority Key Identifier.
- (GP) The discretionary data field SHALL contain the tag 'CA' set with a Subject Key Identifier.
- (X.509) The extension 'Key usage' SHALL be set with the value "keyCertSign".
- (X.509) The extension 'Certificate Policies' SHALL be set with the OID id-rspRole-eum to indicate an EUM Certificate.
- (X.509) The extension 'Basic Constraints' SHALL be set to cA=true. The path length restriction must be set to 0.

CERT.DP.ECDSA:

- (GP) The 'Key usage' field SHALL be set with value '82' (signature verification).
- (GP) The discretionary data field SHALL contain the tag 'C8' set with value "SM-DP+ Certificate".
- (GP) The discretionary data field SHALL contain the tag 'C9' set with an Authority Key Identifier.
- (X.509) The extension 'Key usage' SHALL be set with the value "digitalSignature".
- (X.509) The extension 'Certificate Policies' SHALL be set with the OID id-rspRole-dp-pb to indicate an SM-DP+ Certificate for profile binding.

5 Functions

This section specifies the Functions associated with the Remote SIM Provisioning and Management of the eUICC for consumer Devices.

5.1 Overview of Functions per Interface

The following table presents the normative list of all the functions that are defined in this section.

Request-response functions:

Interface	Functions	Function provider Role
ES2+	DownloadOrder	SM-DP+
	ConfirmOrder	SM-DP+
ES8+	InitialiseSecureChannel ReplaceSessionKeys ConfigureISDP StoreMetadata LoadProfileElements	ISD-R/ISD-P
ES9+	InitiateAuthentication GetBoundProfilePackage HandleProfileInstallationResult	SM-DP+

ES10b	GetEUICCChallenge GetEUICCInfo PrepareDownload LoadBoundProfilePackage GetProfileInstallationResult DeleteProfileInstallationResult	ISD-R
ES10c	GetProfilesInfo EnableProfile DisableProfile DeleteProfile eUICCMemoryReset GetEID	ISD-R

Table 18: Request-response functions

Notification handler functions:

Interface	Notification handler functions	Function handler/recipient
ES2+	HandleProfileInstallationResult	Operator
ES9+	HandleProfileInstallationResult	SM-DP+

Table 19: Notification handler functions

5.2 eUICC Interfaces

This section provides the description of the interfaces and functions within the Remote SIM Provisioning and Management system involving the eUICC, including the following:

- ES6: The interface used by the Operator to manage the content of their Profile.
- ES8+: Provides a secure end-to-end channel between the SM-DP+ and the eUICC for the administration of the ISD-P and the associated Profile during download and installation.
- ES10b: Used by the LPA to transfer a Profile Package to the eUICC.
- ES10c: Used for local End User management of Profiles installed on the eUICC (e.g. Enable, Disable, Delete).
- ESeum: The interface between the EUM and the eUICC.

5.2.1 ES6 (Operator -- eUICC)

This interface is present between the Operator and their Enabled Profile in eUICC. It allows the Operator to make modifications on their Profile in the eUICC using legacy OTA mechanisms.

Currently there are no defined functions for this interface within the scope of this specification.

5.2.2 ES8+ (SM-DP+ -- eUICC)

The ES8+ is an interface defined between the Profile Package Binding function of the SM-DP+ and eUICC. This interface is intended to be tunnelled over the ES9+ and ES10b interfaces.

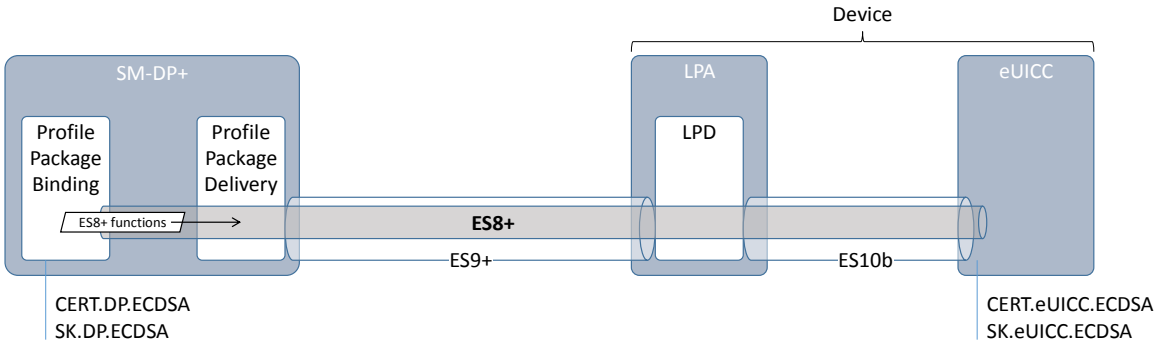

Figure 16: ES8+

The ES8+ functions are addressed to the eUICC through a secure channel established between the Profile Package Binding function of the SM-DP+ and the eUICC.

The secure channel is established by:

- Mutual authentication of the eUICC and the SM-DP+ using SK.DP.ECDSA /CERT.DP.ECDSA and SK.eUICC.ECDSA/CERT.eUICC.ECDSA.
- Session keys agreement based on exchanged one-time public keys of both parties during mutual authentication (see section Annex G).

The SM-DP+ authenticates the eUICC by:

- Verifying the CERT.eUICC.ECDSA with PK.EUM.ECDSA extracted from CERT.EUM.ECDSA, itself verified with PK.CI.ECDSA extracted from CERT.CI.ECDSA.
- Verifying the signature of the eUICC over a SM-DP+ challenge with PK.eUICC.ECDSA extracted from the verified CERT.eUICC.ECDSA.

The eUICC authenticates the SM-DP+ by:

- Verifying the CERT.DP.ECDSA with PK.CI.ECDSA.
- Verifying the signature of the SM-DP+ with the PK.DP.ECDSA extracted from the verified CERT.DP.ECDSA.

The data exchanged after channel establishment are secured using SCP03t as defined in SGP.02 [2]. The eUICC SHALL support the SCP03t with:

- AES in CBC mode with key length of 128 bits, referred as AES-128 (key length as defined in SGP.02 [2]).
- Use of C-MAC, C-DECRYPTION, R-MAC and R-ENCRYPTION.

As a result the SM-DP+ and eUICC are mutually authenticated, all data sent from the Profile Package Binding function of the SM-DP+ to the eUICC are MACed and encrypted, and all responses generated by the eUICC for the Profile Package Binding function of the SM-DP+ are also MACed and encrypted.

5.2.2.1 Function: InitialiseSecureChannel

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R

Description:

This function is used by the SM-DP+ to open a new remote provisioning session with the target eUICC. The function carries the identifier of the remote operation type to be performed by the eUICC (e.g. installation of a new Bound Profile Package) and the necessary material for key agreement with PFS, allowing a secure end-to-end communication between the SM-DP+ and the eUICC:

- Transaction ID.
- Description of the keys to generate.
- One-time public key for key agreement generated by SM-DP+ (otPK.DP.ECKA).
- Signature upon material (including the previously generated otPK.eUICC.ECKA, also acting as an eUICC challenge) to ensure its integrity and authenticity.

The level of security is implicitly deduced from the remote operation type to execute.

This function assumes that the eUICC has already got the SM-DP+ certificate and verified its validity.

The reception of the InitialiseSecureChannel function SHALL be rejected if a secure channel session is already ongoing.

On reception of this command the eUICC SHALL:

- Verify the SM-DP+ signature using the PK.DP.ECDSA; if the signature is invalid the command SHALL be rejected, an error SHALL be returned, Profile installation SHALL be aborted, and any contextual data associated to its Profile installation (like the SM-DP+ certificate) SHALL be discarded.
- Generate the session keys (S-ENC, S-MAC and S-RMAC) and the initial MAC chaining value from received otPK.DP.ECKA and previously generated otSK.eUICC.ECKA.

Command Message

The command message for this function is identified by the tag 'BF23' containing the data structure described in **Error! Reference source not found.**.

Tag	Length	Value Description			MOC
'BF23'	Var	InitialiseSecureChannel Command			M
		Tag	**Length**	**Value Description**	
		'82'	01	Remote operation type identifier	M
		'80'	1-16	Transaction ID	M

		'A6'	Var	Control Reference Template (Key Agreement). This specification only considers a subset of the CRT specified in [1213] section 6.4.2.3 for the MUTUAL AUTHENTICATE Data Field			M
				Tag	**Length**	**Value Description**	
				'80'	1	Key Type according to GlobalPlatform Card Specification [8] Table 11-16 • '88' (AES)	M
				'81'	1	Key length in bytes. In this release of the specification the key length SHALL be '10' (16 bytes)	M
				'84'	1-n	HostID	M
		'5F49'	Var	otPK.DP.ECKA as specified in GlobalPlatform Card Specification Amendment F [13] section 6.4.2.3 for the ePK.OCE.ECKA			M
		'5F37'	Var	SM-DP's signature			M

Table 20: InitialiseSecureChannel Command References

NOTE: The tag '90' for ‘SCP identifiers and parameters’ is not used. This specification only uses one SCP type derived from SCP11a defined in GlobalPlatform Card Specification Amendment F [13]. The tag '95' for 'Key Usage Qualifier' is also not used. This is determined by the for ‘Remote operation type identifier’ (see hereunder). Since only AES key type is specified currently, key values from GlobalPlatform Card Specification [8] Table 11-16 are not used.

The eUICC SHALL verify the values provided for key type and key length.

The tag ‘5F37’ SM-DP+ signature is computed as described in GlobalPlatform Card Specification Amendment E [12], using the SM-DP+ private key SK.DP.ECDSA across the following data:

Tag	Length	Value Description	MOC
‘82’	01	Remote operation type identifier	M
‘80’	1-16	Transaction ID	M
‘A6’	Var	Control Reference Template	M
'5F49'	Var	otPK.DP.ECKA	M
'5F49'	Var	otPK.eUICC.ECKA	M

Table 21: InitialiseSecureChannel Key Type and Length

As the signature includes the otPK.eUICC.ECKA, the eUICC can authenticate the SM-DP+.

Remote operation type identifier are defined in **Error! Reference source not found.**.

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
-	-	-	-	-	-	-	1	Install Bound Profile Package
X	X	X	X	X	X	X	-	RFU

Table 22: InitialiseSecureChannel Operation Types

When type is ‘Install Bound Profile Package’, the implicit Key Usage Qualifier SHALL be set to MAC and ENCRYPTION.

EUICC SHALL return an error '05' for any other Remote operation type identifier value.

If all checking are valid, the eUICC SHALL process the key derivation as described (Annex G).

Response Message

The Response data field for this function is identified by the tag ‘BF23’ and SHALL contain the following data structure:

Tag	Length	Value Description			MOC
'BF23'	Var	InitialiseSecureChannel response			M
		Tag	**Length**	**Value Description**	**MOC**
		'80'	1-16	Transaction ID, as received in the command message.	M
		'81'	1	Error code. This is an optional field. This field SHALL be empty if no error.	O
‘5F37’	Var	eUICC’s signature. Computed as described in GlobalPlatform Card Specification Amendment E [12], using the eUICC private key SK.EUICC.ECDSA upon the data object ‘InitialiseSecureChannel response’ (tag ‘BF23’)			M

Table 23: InitialiseSecureChannel Response Message

Possible error codes:

'01': error in length or structure of command data

‘02’: invalid signature

‘03’: Invalid Transaction ID

‘04’: Unsupported CRT values.

5.2.2.2 Function: ConfigureISDP

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R

Description:

This function is used by the SM-DP+ to provide data to the eUICC for configuring the ISD-P. For this version of the specification, this data element is empty.

NOTE: Even if empty it seems useful to define the data element already in this specification. It is intended to carry information like the amount of assigned memory in future versions. Defining it now already creates a future proof structure.

On reception of this command the eUICC SHALL:

- Create the ISD-P for the Profile and assign an AID value from the range reserved for ISD-Ps in SGP.02 [2].

Command Message

The command message for this function is identified by the tag 'BF24' containing the data structure described in Table 2: : Bound Package Profile Data Structure

Tag	Length	Value Description	MOC
‘BF24’	0	ConfigureISDP command	M

Table 24: ConfigureISDP Command Message

Response Message

The Response data field for this function is identified by the tag 'BF24' and SHALL contain the following data structure.

Tag	Length	Value Description			MOC
'BF24'	Var	ConfigureISDP response			M
		Tag	Length	Value Description	MOC
		‘4F’	5-16	AID of ISD-P.	M
		'81'	1	Error code. This is an optional field. This field SHALL be empty if no error.	O

Table 25: ConfigureISDP Response Message

Possible error codes:

'01': error in length or structure of command data

5.2.2.3 Function: StoreMetadata

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R

Description:

This function is used by the SM-DP+ to provide metadata of the Profile to the eUICC.

On reception of this command the eUICC SHALL:

- Store the data elements for future use by the LPA.

NOTE: This function provides information not already present in the Profile (e.g. ICCID).

Command Message

The command message for this function is identified by the data structure defined hereunder.

```
StoreMetadataRequest ::= [37] SEQUENCE {  -- Tag 'BF25'
       serviceProviderName [17] UTF8String (SIZE(0..32)), -- Tag '91'
       profileName [18] UTF8String (SIZE(0..64)), -- Tag '92' (corresponds to 'Short
Description' defined in SGP.21 [ref])
       iconType [19] IconType OPTIONAL, -- Tag '93' (JPG or PNG)
       icon [20] OCTET STRING (SIZE(0..1024)) OPTIONAL -- Tag '94' (Data of the icon. Size 64
x 64 pixel. This field shall only be present if iconType is present)
}
```

Response Message

The Response data field for this function is identified by the tag ‘BF25’ and SHALL contain the following data structure.

```
StoreMetadataResponse ::= [37] SEQUENCE {  -- Tag 'BF25'
   errorCode[1] Octet1 OPTIONAL -- tag '81'
}
```

Table 26: Void

Possible error codes:

'01': error in length or structure of command data

5.2.2.4 Function: ReplaceSessionKeys

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R

Description:

This function is used to replace the SCP03t session keys (S-ENC, S-MAC and S-RMAC) during the loading of a Bound Profile Package by a new set of session keys (typically the PPK-ENC, PPK-CMAC and PPK-RMAC (see section2.5).

On reception of this function the eUICC SHALL:

- Verify that the new keys are of same length as the old keys
- Replace the current session keys with the new set of keys

Once the function 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.

Command Message

The command message for this function is identified by the tag ‘9F26’ containing the data structure described in Table 27: ReplaceSessionKeys Command Message

Name	Length	Value Description	MOC
InitialMACChainingValue	‘10’	The new initial MAC chaining value	M
Key enc	Var(1)	New value of the session key for encryption/decryption. This typically contains the PPK-ENC	M
Key cmac	Var(1)	New value of the session key for C-MAC computation/verification. This typically contains the PPK-MAC	M
Key rmac	Var(1)	New value of the session key for R-MAC computation/verification. This typically contains the PPK-RMAC	M

Table 27: ReplaceSessionKeys Command Message

NOTE: In this release of the specification the length of the three keys SHALL be 16 bytes each.

Response Message

The Response data field for this function is identified by the tag 'BF26' and SHALL contain a length of '00' in case of execution without error.

Tag	Length	Value Description			MOC
'BF26'	Var	ReplaceSessionKeys response			M
		Tag	Length	Value Description	MOC
		'81'	1	Error code	O

Table 28: ReplaceSessionKeys Response Message

Possible error codes:

'01': error in length or structure of command data

5.2.2.5 Function: LoadProfileElements

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R

Description:

This function is used by the SM-DP+ to provide the Profile Elements defined by SIMalliance [5] to the eUICC.

Command messages, response messages and the processing on the eUICC are defined in SIMalliance specification [5].

After processing all Profile Elements, the eUICC SHALL perform the following:

- Check if the ICCID value in EF_{ICCID} is different from the ICCIDs of all other Profiles on the eUICC. If this check fails, the Profile Download and Installation SHALL fail.
- Else the value in EF_{ICCID} SHALL be used as the ICCID value in the Profile's Meta Data.

5.2.3 ES10b (LPD -- eUICC)

The ES10b is an interface defined between the LPD and ISD-R (LPA Services).

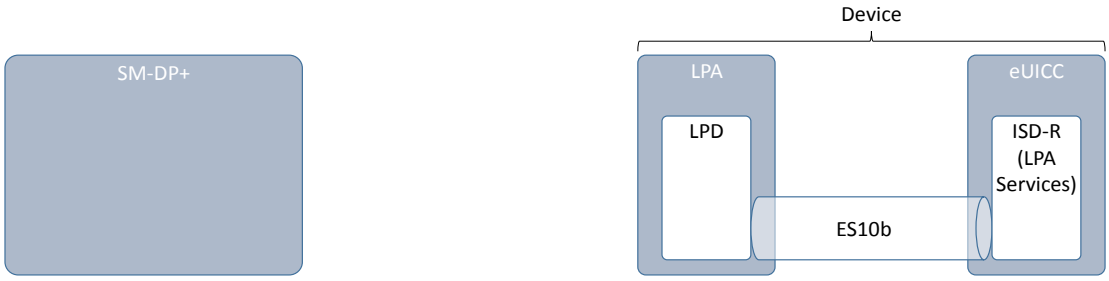

Figure 17: ES10b

Before sending any command to the eUICC, the LPA SHALL establish a logical channel and select the ISD-R.

The opening of the logical channel and the selection of the ISD-R SHALL be done explicitly using, respectively, the MANAGE CHANNEL command and the SELECT command defined in GlobalPlatform Card Specification [8]. This MANAGE CHANNEL and SELECT command can be intrinsically used via a dedicated Device OS API (e.g. OMAPI defined by SIMalliance [OMAPI] if provided).

The Device SHALL ensure that only the LPA, but no other application on the Device, is permitted to select the ISD-R.

This interface is defined with command functions that are mostly handled with a single APDU command and response pair. When multiple commands are required (e.g. STORE DATA), it is indicated by the use of the 'more commands' bit in the P1 byte as defined in GlobalPlatform Card Specification [8], and status bytes controlling the return of additional data (e.g. '61 XX'). In particular if the size of the response is bigger than 255 bytes, the chaining of the commands must be done as defined in ISO/IEC 7816-4 [14]. The responses SHALL be retrieved by the Device using several GET RESPONSE commands.

5.2.3.1 Function: PrepareDownload

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R (LPA Services)

Description:

This function is used by the LPA to transfer to eUICC the SM-DP+ response of the previous authentication request.

This function allows the eUICC to authenticate the SM-DP+, send back the one-time public key (otPK.EUICC.ECKA) and authenticate itself in response to the SM-DP+.

On reception of this command, the eUICC SHALL:

- Verify that a session context exists (i.e. ES10b.GetUICCChallenge function has been previously called).
- Verify the validity of the CERT.DP.ECDSA (using the ECASD service), using the public key PK.CI.ECDSA.
- Verify the signature (DP_Sign1) of the SM-DP+ done upon euiccChallenge (retrieved from the eUICC session context), DP_Challenge, TransactionID and SMD-DP+ address.
- (Optional) Verify that given SM-DP+ OID matches the SM-DP+ OID (tag '5F20') of the CERT.DP.ECDSA.
- Verify that eUICC has the Certificate corresponding to requested Certificate format and Key Parameter Reference value.

If any of these verifications fails, the eUICC SHALL return an error.

If these verifications are successful, the eUICC SHALL:

- Store the TransactionID in the session context.
- Generate an ephemeral key pair otSK.EUICC.ECKA and otPK.EUICC.ECKA, and store otSK.EUICC.ECKA in the session context
- Extract the public key of the CERT.DP.ECDSA and store it in the session context.
- Generate the eUICC_Sign1, as defined in hereunder command description, with the SK.EUICC.ECDSA related to the CERT.EUICC.ECDSA as requested by SM-DP+.

Command Message

This function uses the command message STORE DATA as defined in GlobalPlatform Card Specification [8] with the specific coding defined in this section.

Code	Value	Meaning
CLA	'80'-'83' or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.1.4
INS	'E2'	STORE DATA
P1	'11' or '91'	See below
P2	'xx'	Block number
Lc	Var	Length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 29: PrepareDownload P1

Parameter P1

The P1 SHALL be coded as defined in **Error! Reference source not found.**.

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0/1	-	-	-	-	-	-	-	More blocks/Last block
-	0	0	-	-	-	-	-	No general encryption information or non-encrypted data
-	-	-	1	0	-	-	-	BER-TLV format of the command data field
-	-	-	-	-	-	-	1	Case 4 command as defined in GlobalPlatform AmdA [9]
-	-	-	-	-	X	X	-	RFU

Table 30: PrepareDownload P2

Data field

The data field SHALL be coded as defined hereunder.

```
PrepareDownloadRequest ::= [33] SEQUENCE { -- Tag 'BF21'
   dpSigned1 DPSigned1,                    -- Signed information
   smdpSignature1 [APPLICATION 55] OCTET STRING,      -- DP_Sign1, tag '5F37'
   activationCodeToken UTF8String,
   deviceInfo DeviceInfo,                  -- The Device information
   smdpOid OBJECT IDENTIFIER OPTIONAL,   -- SM-DP+ OID (same value as in
CERT.DP.ECDSA)
   hashCc Octet32 OPTIONAL, -- Hash of confirmation code
   certFormatTobeUsed CertToBeUsed,       -- Certificate Format to be used by eUICC
for signing
   curveTobeUsed Octet1,          -- Curve to be used, coded as a Key Parameter
Reference value as defined in GlobalPlatform Amendment E [AmendE] and {section
2.11}
   smdpCertificate RSPCertificate -- CERT.DP.ECDSA in (one of) the format(s)
requested by eUICC for signature verification (GP or X.509).
}

DPSigned1 ::= SEQUENCE {
   euiccChallenge Octet16, -- The eUICC Challenge
   smdpChallenge Octet16,  -- The SM-DP+ Challenge
   transactionId OctetTo16,-- The TransactionID generated by the SM-DP+
   smdpAddress UTF8String         -- SM-DP+ address
}

CertToBeUsed ::= INTEGER {gp(1), x509(2)}
```

smdpSignature1 SHALL be created using the SK.DP.ECDSA and verified by the eUICC using the PK.DP.ECDSA as described in GlobalPlatform Card Specification Amendment E [12]. smdpSignature1 SHALL apply on the full DPSigned1 data object after encoding (i.e. on DER representation).

Response Message

Data field

The data field SHALL be coded as defined hereunder.

```
PrepareDownloadResponse ::= [33] SEQUENCE { -- Tag 'BF21'
   euiccSigned1 EUICCSigned1,              -- Signed information
   euiccSignature1 [APPLICATION 55] OCTET STRING,     --EUICC_Sign1, tag 5F37
   euiccCertificate RSPCertificate,      -- eUICC Certificate (CERT.EUICC.ECDSA) in
the requested format with the requested Key Parameter Reference value
```

```
   eumCertificate RSPCertificate        -- EUM Certificate (CERT.EUM.ECDSA) in the
requested format with the requested Key Parameter Reference value
}

EUICCSigned1 ::= SEQUENCE {
   smdpChallenge Octet16,  -- The SM-DP+ Challenge
   transactionId OctetTo16,
   smdpAddress UTF8String,
   activationCodeToken UTF8String,
   deviceInfo DeviceInfo,
   smdpOid OBJECT IDENTIFIER OPTIONAL,
   hashCc Octet32 OPTIONAL,             -- Hash of confirmation code
   otpk OCTET STRING,             -- otPK.EUICC.ECKA
   euiccInfo2 EUICCInfo2
}
```

euiccSignature1 SHALL be created using the SK.EUICC.ECDSA and verified using the PK.EUICC.ECDSA as described in GlobalPlatform Card Specification Amendment E [12]. euiccSignature1 SHALL apply on the full EUICCSigned1 data object after encoding (i.e. on DER representation).

Processing State Returned in the Response Message

As defined in GlobalPlatform Card Specification [8] section 11.11.3.2 with the following meaning:

SW1	SW2	Meaning
'6A'	'80'	Invalid Certificate Invalid Signature Unsupported requested Certificate format or curve
'6A'	'88'	No session context exists

Table 31: PrepareDownload Response

5.2.3.2 Function: LoadBoundProfilePackage

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R (LPA Services)

Description:

This function is used to transfer a Bound Profile Package to the eUICC. The transfer is done by calling repeatedly this function with blocks of 255 bytes or less according to the structure of the Bound Profile Package, i.e. each TLV of the BPP that is up to 255 bytes is transported in one APDU. Larger TLVs are sent in blocks of 255 bytes for the first blocks and a last block that MAY be shorter.

Command Message

This function uses the command message STORE DATA as defined in GlobalPlatform Card Specification [8] with the specific coding defined in this section.

Code	Value	Meaning
CLA	‘80’-‘83’ or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.11
INS	'E2'	STORE DATA
P1	'11’ or’ 91 ’	See below
P2	xx	Block number
Lc	xx	Length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 32: LoadBoundProfilePackage Command Message

Parameter P1

The P1 SHALL be coded as defined in **Error! Reference source not found.**.

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0/1	-	-	-	-	-	-	-	More blocks / Last block
-	0	0	-	-	-	-	-	No general encryption information or non-encrypted data
-	-	-	1	0	-	-	-	BER-TLV format of the command data field
-	-	-	-	-	-	-	1	Case 4 command as defined in GlobalPlatform AmdA [9]
-	-	-	-	-	X	X	-	RFU

Table 33: LoadBoundProfileP1

A transfer of an intermediate block of a TLV SHALL be done by indicating “More blocks”. A last block of a TLV SHALL be transferred indicating “Last block”.

Data field

The data field SHALL contain a block of data of the BPP. The transfer and slicing in blocks of data SHALL follow description given in section (2.5.4).

Response Message

Data field

The data field presence in the response message depends on the block status:

- For an intermediate block of data of a BPP TLV, response message SHALL not contain data field
- For the last block of data of a BPP TLV, a response message containing a Profile Installation Result SHALL be present or absent as specified in section 2.5.6.

Processing State Returned in the Response Message

As defined in GlobalPlatform Card Specification [8] section (11.11.3.2).

5.2.3.3 Function: GetEUICCChallenge

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R (LPA Services)

Description:

This function is used by the LPA to initiate a remote management session between an SM-DP+ and the ISD-R. The initiation of the remote management session is materialized on the eUICC by the creation of a context containing an eUICC challenge.

Only one remote management session can be managed by the ISD-R at a time. So an on-going remote management session SHALL be completed before requesting the opening of a new one. If a new remote management session would be requested while a remote management session is on-going, the ISD-R SHALL automatically discard the on-going one.

On reception of this function, the eUICC SHALL:

- Discard the on-going session context, if any.
- Create a new session context and generate a new random challenge attached to this session.

NOTE: This behaviour is specific to ISD-R. This SHALL NOT affect, and SHALL NOT be affected, by a GET CHALLENGE requested to another Security Domain.

Command Message

This function uses the command message GET CHALLENGE as defined in ETSI TS 102 221 [6].

Response Message

Response message SHALL be coded as defined in ETSI TS 102 221 [6]. The response data field SHALL only contain the eUICC random challenge (euiccChallenge) as a raw data of 16 bytes length.

5.2.3.4 Function: GetEUICCInfo

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R (LPA Services)

Description:

This function is used by the LPA to get the eUICC Information as defined in section (4.3). This function MAY be called at any time.

Command Message

This function uses the command message GET DATA as defined in GlobalPlatform Card Specification [8] with the specific coding defined in this section.

Code	Value	Meaning
CLA	‘80’-‘83’ or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.11
INS	'CA'	GET DATA
P1	'xx’	See below
P2	‘xx’	See below
Lc	‘xx’	Not present
Data	'xx xx…'	Empty
Le	'00'	

Table 34: GetEUICC info Command

Parameter P1 and P2

The P1 and P2 parameters define the tag of the data object to be read.
This function SHALL be used with P1='BF' and P2='20' to indicate 'eUICC_Info1' data object.

Data field

The data field SHALL be empty.

Response Message

Data field

The data field SHALL contain the ‘eUICC Information’ ASN.1 data object (as identified by tag contained in P1 and P2) as defined hereunder.

```
EUICCInfo1 ::= [32] SEQUENCE { -- Tag 'BF20'
   eUICCVerSupport [2] VersionType,      -- GSMA SGP.22 version supported
   certificateInfo [5] CertificateInfo,
   curveSigningSupport [6] CurveSigningSupport, -- coded as a sequence of Key
parameter reference values  as defined in GlobalPlatform Amendment E [AmdE] and
Table 13
   curveVerificationSupport [7] CurveVerificationSupport -- coded as a sequence of
Key parameter reference values  as defined in GlobalPlatform Amendment E [AmdE] and
Table 13
}
```

Processing State returned in the Response Message

As defined in GlobalPlatform Card Specification [8] section 11.3.3.2.

5.2.3.5 Function: GetProfileInstallationResult

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R

Description:

This function is used by the LPA to retrieve a Profile Installation Result from eUICC.

A Profile Installation Result is composed of a Profile Installation Report and a Profile Installation Receipt as defined in section (2.5.6).

Command Message

This function uses the command message GET DATA as defined in GlobalPlatform Card Specification [8] with the specific coding defined in this section.

Code	Value	Meaning
CLA	'80'-'83' or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.11
INS	'CA'	GET DATA
P1	'xx'	See below
P2	'xx'	See below
Lc	'xx'	Not present if no command data, otherwise length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 35: GetProfileInstallationResult Comand

Parameter P1 and P2

The P1 and P2 parameters define the tag of the data object to be read.

This function SHALL be used with P1='BF' and P2='28' to indicate 'Profile Installation Result' data object.

Data field

The data field SHALL be empty, indicating to return the 'Profile Installation Result' stored in the eUICC.

Response Message

Data field

The data field SHALL contain the 'Profile Installation Result' data object (as identified by tag contained in P1 and P2) if available.

Tag	Length	Value Description			MOC
'BF28'	Var	Profile Installation Result			M
		Tag	**Length**	**Value Description**	
		'BF23'	Var	Tag for 'ES8+.InitialiseSecureChannel' function response (first tag of a Profile Installation Report)	M
		'5F37'	Var	eUICC's signature for 'ES8+.InitialiseSecureChannel' function response	M

		'87'	Var	Remaining part of the Profile Installation Report (depending of error case) as defined in section 2.5.6	O
		…			
		'BF27'	Var	Profile Installation Receipt	M
		'5F37'	Var	eUICC's signature upon the data object 'Profile Installation Receipt'	M

Table 36: GetProfileInstallationResult Response

Processing State returned in the Response Message

As defined in GlobalPlatform Card Specification [8] section 11.3.3.2.

If no 'Profile Installation Result' data object is available, the error status '6A88' (Referenced data not found) SHALL be returned.

5.2.3.6 Function: DeleteProfileInstallationResult

Related Procedures: Profile Download and Installation

Function Provider entity: ISD-R

Description:

This function is used by the LPA to delete the Profile Installation Result from eUICC.

This function is typically used by the LPA when the LPA has successfully returned the Profile Installation Receipt, and optionally the Profile Installation Report, to the SM-DP+.

On reception of this command the eUICC SHALL delete the pending Profile Installation Result, if any.

Command Message

This function uses the command message DELETE as defined in GlobalPlatform Card Specification [8] with the specific coding defined in this section.

Code	Value	Meaning
CLA	'80'-'83' or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.1.4
INS	'E4'	DELETE
P1	'00'	See below
P2	'00'	See below
Lc	'xx'	Length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 37: DeleteProfileInstallationResult Command

Parameter P1 and P2

P1 parameter SHALL be coded to indicate ‘Last (or only) command’.

P2 parameter SHALL be coded to indicate ‘Delete object’.

Data field

The data field SHALL contain the TL (with no value) identifying the 'Profile Installation Result' data object.

NOTE: As the eUICC will retain at maximum one Profile Installation Result, there is no need to identify a particular 'Profile Installation result' by its Transaction ID.

Tag	Length	Meaning	Presence
'BF28'	0	Profile Installation Result	M

Table 38: DeleteProfileInstallationResult Response

Response Message

Data field

A single byte of '00' SHALL be returned when a pending Profile Installation Result has been deleted, indicating that no additional data is present.

Processing State returned in the Response Message

As defined in GlobalPlatform Card Specification [8] section 11.2.3.2.

If no 'Profile Installation Result' data object was available, the error status '6A88' (Referenced data not found) SHALL be returned.

5.2.4 ES10c (LUI -- eUICC)

The ES10c is an interface defined between the LUI and ISD-R (LPA Services).

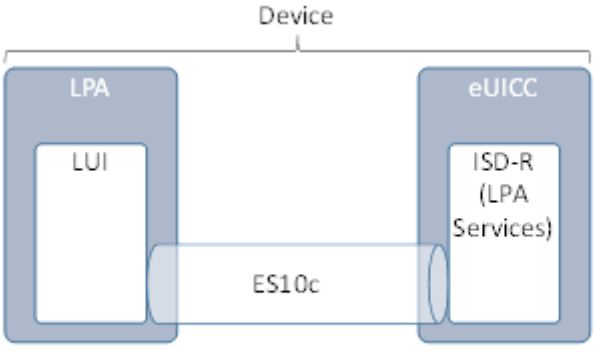

Figure 18: ES10c

Before sending any command defined in this interface to the eUICC, the LPA SHALL establish a logical channel and select the ISD-R. This SHALL be done as described for ES10b.

5.2.4.1 Function: GetProfilesInfo

Related Procedures: Local Profile Management – Enable Profile

Function Provider entity: ISD-R (LPA Services)

Description:

This function is used by the LPA to retrieve the list of Profile information for installed Profiles including their current state (Enabled/Disabled) and their associated Profile Metadata. This function MAY also be used to retrieve this information for a particular Profile.

Command Message

This function uses the command message GET STATUS as defined in GlobalPlatform Card Specification [8] with the specific coding defined in this section.

Code	Value	Meaning
CLA	'80'-'83' or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.1.4
INS	'F2'	GET STATUS
P1	'xx'	See below
P2	'xx'	See below
Lc	'xx'	Length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 39: GetProfilesInfo Command

Parameter P1 and P2

The P1 SHALL be coded to indicate ‘Applications and Supplementary Security Domains only’ (P1=’40’).

The P2 SHALL be coded as per table 2 to retrieve the Profiles Info. The command SHALL use the general chaining mechanism defined for ES10b in case the response data is larger than 255 bytes. Bit b2 SHALL be set to 1. Coding is given in the following : GetProfilesInfo Response".

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
X	X	X	X	X	X	-	-	RFU
-	-	-	-	-	-	-	0	Get first or all occurrence(s)
-	-	-	-	-	-	1	-	Response data structure coded as in GlobalPlatform Card Specification [8] Table 11.36

Table 40: GetProfilesInfo Response

The search is limited to the ISD-P instances.

Data field sent in the Command Message

The GET STATUS command message data field SHALL contain at least one TLV coding search qualifier: the AID (tag '4F').

It SHALL be possible to search for all the Profiles that match the selection criteria according to the reference control parameter P1 using a search criteria of '4F 00'.

It SHALL be possible to search for a particular Profile using the search criteria '4F XX XX …' where 'XX XX …' is the AID of the target ISD-P or by ICCID using the tag '5A' with a wildcard '4F' followed by the 10 bytes of the ICCID value, i.e. '4F 00 5A 0A XX XX XX XX XX XX XX XX XX XX'.

No other search criteria are defined.

The tag list (tag '5C') indicates to the eUICC how to construct the response data (i.e. which data to include in the response data field) for each Profile matching the search criteria.

An eUICC SHALL support the following filtering criteria:

- ICCID, tag '5A'.
- ISD-P AID, tag '4F'.
- Profile state, tag '9F70'.
- Profile Nickname, tag '90'.
- Service provider name, tag '91'.
- Profile name, tag '92'.
- Icon type, tag '93'.
- Icon, tag '94'.

If no filtering criteria is present, eUICC SHALL return the full ProfileInfo data object as defined in the response data filed section hereunder.

Example of use:

- Retrieve the list of ProfileInfo for all installed Profiles with all information. The data field SHALL be coded as '4F 00'
- Retrieve all information of a particular Profiles/ISD-P having the following A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00. The data field SHALL be coded as '4F 10 A0 00 00 05 59 10 10 FF FF FF FF 89 00 00 10 00'
- Retrieve ICCID and Profile state for all installed Profiles. The data field SHALL be coded as '4F 00 5C 03 5A 9F70'

Response Message

Data Field Returned in the Response Message:

The data field SHALL contain the list of data objects as required by the filtering criteria. The list MAY be empty if no Profile is installed or if no Profile matches the given search criteria.

The following is the definition of the ProfileInfo data object:

```
ProfileInfo ::= [PRIVATE 3] SEQUENCE {
       iccid [APPLICATION 26] OCTET STRING (SIZE(10)) OPTIONAL, -- ICCID as coded in EFiccid,
corresponding tag is '5A'
       isdpAid [APPLICATION 15] OctetTo16 OPTIONAL, -- AID of the AISD-P containing the
Profile, tag '4F'
       profileState [112] ProfileState OPTIONAL, -- Tag '9F70'
       profileNickname [APPLICATION 16] UTF8String (SIZE(0..64)) OPTIONAL, -- Tag '50'
       serviceProviderName [APPLICATION 17] UTF8String (SIZE(0..32)) OPTIONAL, -- Tag '51'
       profileName [APPLICATION 18] UTF8String (SIZE(0..64)) OPTIONAL, -- Tag '52'
       iconType [APPLICATION 19] IconType OPTIONAL, -- Tag '53'
       icon [APPLICATION 20] OCTET STRING (SIZE(0..1024)) OPTIONAL -- Tag '54'
}

IconType ::= INTEGER {jpg(0), png(1)}
ProfileState ::= INTEGER {disabled(0), enabled(1)}
```

Processing State returned in the Response Message:

As defined in GlobalPlatform Card Specification [8] section 11.4.3.2, with the exception that the status '6310' SHALL NOT be used; responses larger than 256 bytes are indicated by the status '61xx', see section 5.2.3.

5.2.4.2 Function: EnableProfile

Related Procedures: Local Profile Management – Enable Profile

Function Provider entity: LPA Services

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, if any. This SHALL be performed in an atomic way, meaning that in case of any error during the command execution, the command SHALL stop and SHALL leave the involved Profiles in their original states prior to command execution.

Upon reception of the EnableProfile function, the eUICC SHALL:

- Verify that the target Profile is in the Disabled state
- If any of these verifications fail, terminate the command with an error status word.
- Disable the currently Enabled Profile (if any) and Enable the target Profile
- Send the REFRESH command in “Profile State changed” mode (if supported by the Device) or “UICC Reset” mode to the Device according to ETSI TS 102 223 [31].

Command Message

This function uses the command message STORE DATA as defined in SGP.02 [2] section 4.1.1.2 with extension described hereunder.

Data field sent in the Command Message

In addition, the command provides the possibility to identify the target Profile using the ICCID. In that case the data field of the command SHALL be coded according to : Enable Profile Command".

DGI	Length	Value Description			MOC
'3A03'	Var	Enable Profile			M
		Tag	Length	Value Description	MOC
		'5A'	10	ICCID	M

Table 41: Enable Profile Command

Response Message

The response message SHALL be coded as defined in SGP.02 [2].

5.2.4.3 Function: DisableProfile

Related Procedures: Local Profile Management – Disable Profile

Function Provider entity: LPA Services

Description:

This function is used to disable a Profile on the eUICC.

Upon reception of the “DisableProfile” function, the eUICC SHALL:

- Verify that the target Profile is in the Enabled state
- If any of these verifications fail, terminate the command with an error status word.
- Disable the target Profile
- Send the REFRESH command in “Profile State Changed” mode (if supported by the Device) or “UICC Reset” mode to the Device according to ETSI TS 102 223 [31].

Command Message

This function uses the command message STORE DATA as defined in SGP.02 [2] section 4.1.1.3 with extension described hereunder.

Data field sent in the Command Message

In addition, the command provides the possibility to identify the target Profile using the ICCID. In that case the data field of the command SHALL be coded according to : Disable Profile Command"

DGI	Length	Value Description			MOC
'3A04'	Var	Disable Profile			M
		Tag	Length	Value Description	MOC
		'5A'	10	ICCID	M

Table 42: Disable Profile Command

Response Message

The response message SHALL be coded as defined in SGP.02 [2].

5.2.4.4 Function: DeleteProfile

Related Procedures: Delete Profile

Function Provider entity: ISD-R (LPA Services)

Description:

This function is used by the LPA to delete a Profile from eUICC. This function can be used at any time by the LPA. The Profile to be deleted can be identified by ISD-P AID or ICCID.

On reception of this command the eUICC SHALL check the state of the targeted Profile. If Profile is in Enabled state, the eUICC SHALL return an error; else the eUICC SHALL delete the ISD-P containing the target Profile and its related metadata.

Command Message

This function uses the command message DELETE as defined in GlobalPlatform Card Specification Amendment C [10] for cumulative delete with the specific coding defined in this section.

Code	Value	Meaning
CLA	'80'-'83' or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.1.4
INS	'E4'	DELETE
P1	'00'	See below
P2	'40'	See below
Lc	'xx'	Length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 43: Delete Profile Command

Parameter P1 and P2

P1 parameter SHALL be coded to indicate 'Last (or only) command' ('00').

P2 parameter SHALL be coded to indicate 'Delete a root Security Domain and all associated' Applications' ('40'). In that case it SHALL also delete associated Profile Metadata.

Data field

The data field SHALL contain the TLV identifying the Profile to be deleted. Profile MAY be identified by its ISD-P AID or ICCID, but exactly one of the two TLV SHALL be provided.

Tag	Length	Meaning	Presence
'4F'	5-16	AID of the ISD-P containing the profile to be deleted	C
'5A'	10	ICCID of the Profile to be deleted	C

Table 44: Delete Profile Response

Response Message

Data field

A single byte of '00' SHALL be returned when the Profile has been deleted, indicating that no additional data is present.

Processing State returned in the Response Message

As defined in GlobalPlatform Card Specification [8] section 11.2.3.2.

The following status words SHALL be returned by the ISD-R:

- If the ISD-P to be deleted does not exist: '6A 82' (application not found).
- If the ISD-P to be deleted is in Enabled state: '69 85' (conditions of use not satisfied).

5.2.4.5 Function: eUICCMemoryReset

Related Procedures: eUICC Memory Reset

Function Provider entity: LPA Services

Description:

This function deletes all the Profiles stored on the eUICC regardless of their status.

This SHALL be performed in an atomic and non-reversible way in case of external interruptions (e.g. power loss): the eUICC SHALL continue the processing of that command upon the next eUICC power on. In case of any other error during the command execution, the command SHALL stop and SHALL leave the eUICC and the involved Profiles in their original states prior to command execution.

Upon reception of the eUICCMemoryReset function, the eUICC SHALL:

- Delete all the ISD-P with their Profiles regardless of their status
- Delete all the Profiles Metadata stored in the ISD-R

Command Message

This function uses the command message STORE DATA as defined in section as defined in GlobalPlatform Card Specification [8] with extension described hereunder.

Code	Value	Meaning
CLA	'80'-'83' or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.11
INS	'E2'	STORE DATA
P1	'88'	See below
P2	'00'	
Lc	xx	Length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 40a: eUICCMemoryReset Command Message

Parameter P1

The P1 SHALL be coded as defined in the following table.

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	-	-	-	-	-	-	-	Last block
-	0	0	-	-	-	-	-	No general encryption information or non-encrypted data
-	-	-	0	1	-	-	-	DGI format of the command data field
-	-	-	-	-	-	-	0	Case 3 command as defined in GlobalPlatform AmdA [9]
-	-	-	-	-	X	X	-	RFU

Table 40b: P1 for eUICCMemoryReset command

Data field sent in the Command Message

The data field of the command SHALL be coded according to : eUICC Memory Reset Attribute Data Field.

Data Field Sent in the Command Message

DGI	Length	Value Description			MOC
'3A09'	'03'	eUICCMemoryReset			M
		Tag	Length	Value Description	MOC
		'82'	'01'	eUICC Memory Reset options	M

Table 45: eUICC Memory Reset Attribute Data Field

eUICC Memory Reset options coding:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
-	-	-	-	-	-	-	1	All Profiles SHALL be deleted
X	X	X	X	X	X	X	-	RFU

Table 46: eUICC Memory Reset options

Response Message

The data field of the response message SHALL NOT be present.

Processing State Returned in the Response Message

See GlobalPlatform Card Specification [8] section 11.11.3.2.

5.2.4.6 Function: GetEID

Related Procedures: Profile Download Initiation

Function Provider entity: ISD-R (LPA Services)

Description:

This function is used by the LPA to get the EID from the eUICC. This function can be used at any time by the LPA, and for instance during the Profile Download Initiation when the End user MAY have to provide the EID to the contracting Service Provider/Operator, and when the EID is not available by another mean, e.g. the End user MAY have lost the physical material where it was printed on.

NOTE: This function is coded with a command message GET DATA on the ISD-R (LPA Services). It MAY also be possible to use the GET DATA as defined in SGP.02 [2] targeting directly the ECASD.

Command Message

This function uses the GET DATA command defined in GlobalPlatform Card Specification [8], section 11.3, with the specific coding defined in this section.

Code	Value	Meaning
CLA	'80'-'83' or 'C0'-'CF'	See GlobalPlatform Card Specification [8] section 11.3
INS	'CA'	GET DATA
P1	'BF'	See below
P2	'30'	See below
Lc	'03'	Length of data field
Data	'xx xx…'	See below
Le	'00'	

Table 47: GetEID Command

Parameter P1 and P2

The P1 and P2 parameters define the tag of the data object to be read. Tag P1 and P2 SHALL be set to 'BF 30' (forwarded CASD Data mechanism as defined in GlobalPlatform Card Specification Amendment C [10]) to enable retrieval of a data object defined in ECASD.

Data field

The data field SHALL indicate EID data object ‘5C 01 5A’ (tag ‘5A’ identifies the EID).

Response Message

Data field

The data field of the response message SHALL contain data structure as defined in : GetEID Response".

Tag	Length	Value Description			MOC
'BF30'	18	Forwarded ECASD Data tag			M
		Tag	**Length**	**Value Description**	**MOC**
		'5A'	16	EID value	M

Table 48: GetEID Response

Processing State Returned in the Response Message

As defined in GlobalPlatform Card Specification Amendment C [10].

5.2.4.7 Function: Add/Update Profile Nickname

Related Procedures: Add/Update Profile Nickname – LUI out of the eUICC

Function Provider entity: LPA Services

Description:

This function is used to add or update a Profile Nickname associated to one Profile present on-card.

Upon reception of the SetNickname function, the eUICC SHALL:

- Verify that the target Profile is present on the eUICC
- Update the target Profile nickname with the provided data

Command Message

This function is based on a STORE DATA as defined in GlobalPlatform Card Specification [8] with the following extension.

In case a Profile Nickname already exists for the indicated Profile, the Profile Nickname SHALL be updated with the new value. In case the Profile Nickname TLV has a Length set to 0, the nickname SHALL be removed. Removing a non-existing Profile Nickname SHALL NOT be considered an error.

Data field sent in the Command Message

Profile Nickname TLV SHALL be the DER encoding of the ProfileNicknameInformation defined as follows:

```
-- Definition of Profile Nickname Information
SetNicknameRequest ::= [41] SEQUENCE {
   iccid [APPLICATION 26] OCTET STRING (SIZE(10)),
   profileNickname [16] UTF8String (SIZE(0..64))
}
```

Response Message

The data field of the response message SHALL NOT be present.

Specific Processing State returned in response Message:

'6A 88';: Profile not found.

5.3 Off-Card Interfaces

This section provides the description of the interfaces and functions within the Remote SIM Provisioning and Management system that do not terminate at the eUICC, including the following:

- Server to Server Interface:
  - ES2+: Interface between the Operator and the SM-DP+ used by the Operator to order Profile Package preparation.
- Device to Server:
  - ES9+: Used to provide a secure transport between the SM-DP+ and the LPA (LPD) for the delivery of the Profile Package.

ESop (Interface between the End User and the Operator) and ESeu (Interface between the End User and the LUI) are out of scope of this document.

5.3.1 Function commonalities

Each functions represents an entry points that is provided by a Role (function provider), and that can be called by other Roles (function requester).

5.3.1.1 Common data types

The functions provided in this section deal with management of eUICC and Profile, so that the common data defined in this section need to be used in most of the functions.

Type name	Description	Type definition
Hexadecimal String	String of even length composed of characters between '0' to '9' and 'A' to 'F' or 'a' to 'f'.	
AID	The AID (Application Identifier) of an Executable Load File, an Executable Module, a security domain, or an Application.	Hexadecimal string representation of 5 to 16 bytes.
DATETIME	Any date and time used within any interface of this specification.	String format as specified by W3C: YYYY-MM-DDThh:mm:ssTZD Where: YYYY = four-digit year MM = two-digit month (01=Jan, etc.) DD = two-digit day of month (01-31) hh = two digits of hour (00 -23) mm = two digits of minute (00 - 59) ss = two digits of second (00 - 59) TZD = time zone designator (Z, +hh:mm or -hh:mm) Ex: 2001-12-17T09:30:47Z

EID	The EID type is for representing an eUICC-ID. An eUICC-ID is primarily used in the “Embedded UICC Remote Provisioning and Management System” to identify an eUICC. See SGP.02 [2] section 5.2.4.6 for EID description.	Hexadecimal string
ICCID	The ICCID type is for representing an ICCID (Integrated Circuit Card Identifier). The ICCID is primarily used to identify a Profile. ICCID is defined according to ITU-T recommendation E.118 [21].	String representation of up to 20 decimal digits, and padded with F. Ex: 8947010000123456784F
KCV	The KCV stands for "Key Check Value". It provides the material for receiving entity to ensure that it uses the same key value as the sending entity. See Annex G for detail of KCV computation.	Hexadecimal string
MSISDN	The Mobile Station ISDN (Integrated Services Digital Network) Number.	String representation of up to 15 decimal digits,
IMSI	The IMSI (International Mobile Subscriber Identity) used to identify the Subscriber of a Mobile Subscription.	String representation of up to 15 decimal digits including MCC (3 digits) and MNC (2 or 3 digits), as defined in ITU E.212 [30]
OID	An Object Identifier.	String representation of an OID, i.e. of integers separated with dots (e.g.: '1.2', '3.4.5')
TAR	The TAR (Toolkit Application Reference) of a security domain or an Application.	String - Hexadecimal string representation of exactly 3 bytes
VERSION	The Version type is for indicating a version of any entity used within this specification. A version is defined by its major, minor and revision number.	String representation of three integers separated with dots (e.g.: ‘1.15.3’).

Table 49: Common data types

5.3.1.2 Request-Response Function

As defined in SGP.02 [2].

5.3.1.3 Notification Handler Function

As defined in SGP.02 [2].

5.3.1.4 Functions Input header

As defined in SGP.02 [2].

5.3.1.5 Functions Output header

As defined in SGP.02 [2].

5.3.1.6 Status Code

This specification relies on Subject Codes and Reason Codes as defined in SGP.02 [2]. In addition this specification defines the following codes:

Subject Code:

8.1 eUICC

8.1.2 EUM Certificate

8.1.3 eUICC Certificate

8.2 Profile

8.2.6 Matching ID

8.2.7 Confirmation Code

8.8 SM-DP+

8.8.1 SM-DP+ Address

8.8.2 Security configuration

Reason Code:

6. Security Error

6.3 Expired

5.3.2 ES2+ (Operator -- SM-DP+)

The ES2+ interface is used by the Operator to order the Profile Package preparation for specific eUICC(s) and the delivery of the Profile Package from the SM-DP+.

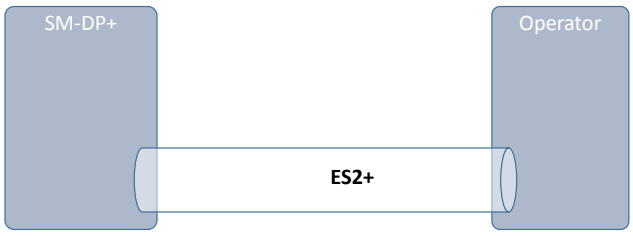

Figure 19: ES2+

The Operator communicates with the SM-DP+ through a secure connection. The level of security requested on this interface and the level of data encryption is defined in GSMA SAS SM specification [23].

5.3.2.1 Function: DownloadOrder

Related Procedures: Download initiation procedure

Function Provider entity: SM-DP+

Description:

This function is used to instruct the SM-DP+ of a new Profile download request.

The EID is optional and MAY not be known at this stage. In this case the SM-DP+, with the Operator, MAY verify if the EID acquired during the Download and installation procedure is compatible with the requested Profile Type (see also Annex H).

Upon reception of this function call, the SM-DP+ SHALL:

- Reserve an ICCID in its inventory. If the ICCID was provided as input data, the reservation SHALL use this value. Else the reservation SHALL be done corresponding to the requested Profile Type with a value available in the SM-DP+'s inventory.
- Optionally, if not already done, the SM-DP+ performs the ‘Profile generation’ and ‘Profile protection’ steps, as described in section (2.5.3), for the Profile identified by its ICCID.
- If the EID is known, the ICCID is allocated to this EID.

The SM-DP+ MAY perform additional operations, which are out of scope of this specification.

This function SHALL return one of the following:

- A ‘Function execution status’ with ‘Executed-success’ indicating that the ICCID has been reserved.
- A ‘Function execution status’ indicating ‘Failed’ with a status code as defined] [: DownloadOrder Specific status codes] or a specific status code as defined in the table here after.

Additional input data:

Input data name	Description	Type	No.	MOC
eid	Identification of the targeted eUICC.	EID	1	O
iccid	Identification of the Profile to download and install in the eUICC.	ICCID	1	C
profileType	Identification of the Profile Type to download and install in the eUICC.	String	1	C

Table 50: DownloadOrder Command

NOTE: Operator can provide the ICCID and/or the Profile Type. In case where the Profile Type is provided, the SM-DP+ is free to select one of the Profiles that matches the Profile Type.

Additional output data:

Output data name	Description	Type	No.	MOC
iccid	Identification of the Profile to download and install in the eUICC. If iccid was provided as an input data, the returned value SHALL be the same. If not	ICCID	1	M

	provided as an input data the returned value SHALL be one of the values available in the SM-DP+ inventory and corresponding to the Profile Type.			

Table 51: DownloadOrder Response

Specific status codes

Subject Code	Subject	Reason code	Reason	Description
8.2.1	Profile ICCID	3.9	Unknown	Indicates that the Profile, identified by this iccid is unknown to the SM-DP+.
8.2.1	Profile ICCID	1.2	Not Allowed (Authorisation)	Indicates that the function caller is not allowed to perform this function on the target Profile.
8.2.5	Profile Type	3.9	Unknown	Indicates that the Profile Type identified by this Profile Type is unknown to the SM-DP+.
8.2.5	Profile Type	1.2	Not Allowed (Authorisation)	Indicates that the function caller is not allowed to perform this function on the Profile Type.
8.2.5	Profile Type	3.7	Unavailable	No more Profile available for the requested Profile Type.

Table 52: DownloadOrder Specific status codes

5.3.2.2 Function: ConfirmOrder

Related Procedures: Download initiation procedure

Function Provider entity: SM-DP+

Description:

This function is used to confirm a previously requested download order.

On reception of this function call, the SM-DP+ SHALL:

- Confirm the allocation of an ICCID in its inventory.
- Generate a MatchingID (see section 4.1.1) if it is not provided by the Operator.
- Store the MatchingID.
- If the Confirmation Code is provided by the Operator, calculate the hash value of the Confirmation Code and store the hash value together with the MatchingID.

The SM-DP+ MAY perform additional operations.

This function SHALL return one of the following:

- A ‘Function execution status’ with ‘Executed-success’ indicating that the ICCID has been reserved.
- A ‘Function execution status’ indicating ‘Failed’ with a status code as defined (5.3.1.6) or a specific status code as defined in the table here after.

Additional input data:

Input data name	Description	Type	No.	MOC
iccid	Identification of the Profile to download and install in the eUICC.	ICCID	1	M
matchingId	The MatchingID as defined in section (4.1.1), when generated by the Operator.	String	1	O
confirmationCode	A code used to authorise the usage of the MatchingID to confirm the download and installation of the Profile.	String	1	O

Table 53: ConfirmOrder Additional Input

Additional output data:

Output data name	Description	Type	No.	MOC
matchingId	The MatchingID as defined in section (4.1.1). If MatchingID was provided as an input data, the returned value SHALL be the same.	String	1	M
smdpAddress	The SM-DP+ address to be used for this specific download order. The valid format of an SM-DP+ address is described in section 4.1.	String	1	O

Table 54: ConfirmOrder Additional Output

Specific status codes

Subject Code	Subject	Reason code	Reason	Description
8.2.1	Profile ICCID	3.9	Unknown	Indicates that the Profile, identified by this iccid is unknown to the SM-DP+.
8.2.1	Profile ICCID	1.2	Not Allowed (Authorisation)	Indicates that the function caller is not allowed to perform this function on the target Profile.
8.2.6	Matching ID	3.3	Already in Use (Uniqueness)	Conflicting MatchingID value.

Table 55: ConfirmOrder Specific Status Codes

5.3.2.3 Function: HandleProfileInstallationResult

Related Procedures: Profile download procedure

Function Provider entity: Operator

Description:

This function SHALL be called to notify the download and installation operation status of the Profile identified by its ICCID on 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 been performed.

What is performed by the Operator receiving this notification is out of scope of this specification.

Additional input data:

Input data name	Description	Type	No.	MOC
eid	Identification of the targeted eUICC.	EID	1	M
iccid	Identification of the Profile to download and install in the eUICC.	ICCID	1	M
profileType	Identification of the Profile Type to download and install in the eUICC.	String	1	M
completionTimestamp	Indication of the date/time when the operation has been performed.	DATETIME	1	M
resultCode	The result code can be extracted from the receipt received from the SM-DP+, this information is only available if the transaction has not expired.	Hexadecimal string	1	C
operationStatus	The status of the operation performed on Profile identified by its ICCID on the eUICC identified by its EID. The Execution Status type is re-used to specify the result of processing of the operation, and optionally to provide information on any encountered problem (status code, data/object that causes the status code, and message to provide textual and human readable explanation of the status code).	EXECUTION STATUS	1	M

Table 56: HandleProfileInstallationResult Additional Input Data

5.3.3 ES9+ (LPA -- SM-DP+)

ES9+ is the interface between:

- The LPA entity and more precisely with the LPD endpoint.
- And the SM-DP+ and more precisely with the Profile Package Delivery endpoint.

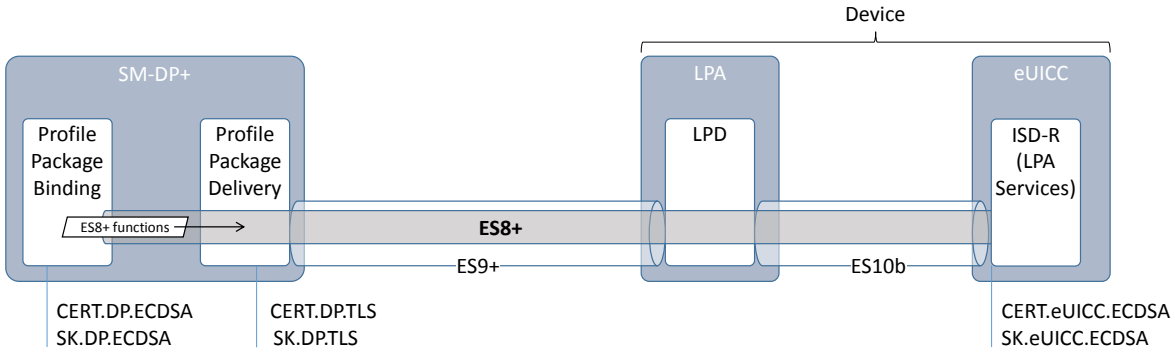

Figure 20: ES9+

The LPA SHALL communicates with the SM-DP+ secured by HTTPS in server authentication mode.

TLS version 1.2 is the minimum required for the connection.

The format of the TLS Certificates (CERT.DP.TLS) used for TLS connections is described in section 4.5.2.2.

During TLS establishment, LPA SHALL verify the received CERT.DP.TLS according to section 4.5.2.3. If any of these verifications fail, the TLS connection SHALL be rejected, and on-going procedure SHALL fail.

5.3.3.1 Function: InitiateAuthentication

Related Procedures: Download and Installation Procedure

Function Provider entity: SM-DP+

Description:

This function is used by the LPA to request the SM-DP+ authentication. This is following the "GetEUICCChallenge" between the eUICC and the LPA where the LPA retrieves material from the eUICC to be provided to the SM-DP+.

On reception of this function call, the SM-DP+ SHALL:

- Check if the received address matches the own SM-DP+ address.
- Check if among the Certificate Format and Key Parameter Reference Value supported by the eUICC, there is a common configuration that is actually supported by the SM-DP+ for signature and Bound Profile Package generation.

If any of these verifications fails, the SM-DP+ SHALL return an error.

Else the SM-DP+ SHALL:

- Generate a TransactionID which is used to uniquely identify the Profile download and installation transaction.
- Generate a DP_Challenge for eUICC authentication.
- Generate a signature (called smdpSignature1) as described in (section 5.2.3.1 ES10b.PrepareDownload).

The SM-DP+ MAY perform additional operations, which are out of scope of this specification.

Input data:

Input data name	Description	Type	MOC
euiccChallenge	(5.2.3.3 Function: GetEUICCChallenge) eUICCchallenge is retrieved from the eUICC by calling ES10b.GetEUICCChallenge	Binary[16]	M
svn	SVN is part of euiccInfo1 that can be retrieved by calling ES10b.GetEUICCInfo (see section 5.2.3.4)	VERSION	M

euiccInfo1	euiccInfo1 can be retrieved by calling ES10b.GetEUICCInfo (see section 5.2.3.4)	Binary	M
smdpAddress	Provided by the Activation Code (see section 4.1)	String	M

Table 57: InitiateAuthentication Input Data

Output data:

Output data name	Description	Type	MOC
transactionId	Transaction ID as generated by the SM-DP+ (see section 3.1.2)	Binary[1-16]	M
dpSigned1	The data object as defined in (section 5.2.3.1 ES10b.PrepareDownload)	DPsigned1	M
smdpSignature1	SM-DP+ signature as defined in (section 5.2.3.1 ES10b.PrepareDownload)	Binary	M
certFormatTobeUsed	The certificate format to be used by the eUICC, as defined in (section 5.2.3.1 ES10b.PrepareDownload)	CertToBeUsed	M
curveTobeUsed	The curve to be used by the eUICC, as defined in (section 5.2.3.1 ES10b.PrepareDownload)	Binary[1]	M
smdpCertificate	SM-DP+ Certificate (CERT.DP.ECDSA) in the format requested by the eUICC	Binary	M

Table 58: InitiateAuthentication Output Data

Specific status codes

Subject Code	Subject	Reason code	Reason	Description
8.8.1	SM-DP+ Address	3.8	Refused	Invalid SM-DP+ Address
8.8.2	Security configuration	3.1	Unsupported	The required security configuration (certificate format and curves references) not supported by the SM-DP+

Table 59: Status codes

5.3.3.2 Function: GetBoundProfilePackage

Related Procedures: Download and Installation Procedure

Function Provider entity: SM-DP+

Description:

This function SHALL be called to request the delivery and the binding of a Profile Package for the eUICC.

End point of this function on SM-DP+ side is the Profile Package Delivery which is in charge to deliver the input data to the Profile Package Binding.

The Profile Package Binding output data is delivered to the LPA through the Profile Package Delivery.

This function is correlated to a previous normal execution of an “ES9+.InitiateAuthentication” function through a TransactionID delivered by the SM-DP+.

On reception of this function call, the SM-DP+ SHALL:

- Verify the validity of the CERT.EUM.ECDSA, using the public key PK.CI.ECDSA.
- Verify the validity of the CERT.EUICC.ECDSA, using the public key PK.EUM.ECDSA.
- Verify the eUICC signature (euiccSignature1) using the PK.EUICC.ECDSA as described in (section 5.2.3.1 ES10b.PrepareDownload).
- Verify if there is a related pending Profile download order for the MatchingID (AC_Token) provided.
- Verify if this order is already linked to an EID; if yes, the SM-DP+ SHALL also check that the EID matches.
- Verify if this order requires a Confirmation Code verification; if yes, SM-DP+ SHALL verify that the received Hashed Confirmation Code matches the value known by the SM-DP+.
- Perform an eligibility check as described in Annex F.

If any of these verifications fails, the SM-DP+ SHALL return an error.

Else the SM-DP+ SHALL:

- Generate one time ECKA key pair (otPK.DP.ECKA, otSK.DP.ECKA) for key agreement.
- Generate the session keys (S-ENC, S-MAC and S-RMAC) and the initial MAC chaining value from received otPK.EUICC.ECKA and previously generated otSK.DP.ECKA.
- Generate the ProfileMetada of the Profile.
- Generate the Bound Profile Package as described in (section 2.5.4).

The SM-DP+ MAY perform additional operations, which are out of scope of this specification.

Input data:

Input data name	Description	Type	MOC
transactionId	Transaction ID as generated by the SM-DP+ (see section 3.1.2).	Binary[1-16]	M
prepareDownloadResponse	Defined in (see section 5.2.3.1).	Binary	M

Table 60: GetBoundProfilePackage Input Data

Output data:

Output data name	Description	Type	MOC
transactionId	Transaction ID as generated by the SM-DP+ (see section 3.1.2).	Binary[1-16]	M
profileMetadata	Profile Metadata for the purpose of display by the LPA	Profile Metadata	M
boundProfilePackage	This is the data structure as defined in (2.5.4) to be transferred to the eUICC using ES10b.LoadBoundProfilePackage. (see section 5.2.3.2)	Binary	M

Table 61: GetBoundProfilePackage Output Data

The **PROFILE_METADATA** type is defined by the following data structure:

Data name	Description	Type	No.	MOC
iccid	ICCID of the Profile	ICCID	1	M
serviceProviderName	Service Provider name	String	1	M
profileName	Name of the Profile as set by the Operator	String	1	M
iconType	JPG (0), PNG (1)	Integer	1	O
icon	Contains the binary payload of the icon. SHALL be present if iconType is present	Binary	1	C

Table 62: PROFILE METADATA

Specific status codes

Subject Code	Subject	Reason code	Reason	Description
8.1.2	EUM Certificate	6.1	Verification Failed	Certificate is invalid
8.1.2	EUM Certificate	6.3	Expired	Certificate has expired
8.1.3	eUICC Certificate	6.1	Verification Failed	Certificate is invalid
8.1.3	eUICC Certificate	6.3	Expired	Certificate has expired
8.1	eUICC	6.1	Verification Failed	eUICC signature is invalid
8.2.6	MatchingID	3.8	Refused	MatchingID (AC_Token) is refused
8.1.1	EID	3.8	Refused	EID doesn't match the expected value
8.2.7	Confirmation Code	8.2	Mandatory Element Missing	Confirmation Code is missing
8.2.7	Confirmation Code	3.8	Refused	Confirmation Code is refused
8.2.5	Profile Type	4.3	Stopped on warning	No eligible Profile for this eUICC/Device

Table 63: Specific status codes

5.3.3.3 Function: HandleProfileInstallationResult

Related Procedures: Download and Installation Procedure

Function Provider entity: SM-DP+

Description:

This function SHALL be called by the LPA to notify the download and installation operation status of the session identified by its TransactionID.

The SM-DP+ SHALL notify the Operator about the status of the Profile Download and Installation using the ES2+. HandleProfileInstallationResult.

The SM-DP+ MAY perform additional operations which are out of scope of this specification.

Input data:

Input data name	Description	Type	MOC
result	The result SHALL contain the Profile Installation Result Data Object as defined in (2.5.6)	Binary	M

Table 64: HandleProfileInstallationResult Input Data

Output data:

No output data.

5.3.4 Function Binding in JSON

5.3.4.1 Security

To secure the messages being sent between Function requester and Function provider; Transport Layer Security (TLS) with mutual authentication on ES2+ and server authentication on ES9+.

This specification mandates usage of TLS v1.2 defined in RFC 5246 [16] to allow appropriate algorithm and key length.

5.3.4.1.1 Secure Channel Set-Up

The process of setting up secure channel is out of scope of this document. This process includes the exchange of the following information:

- Function requester (on ES2+) and Function provider (on ES2+ and ES9+) OIDs SHALL be registered and respective values have been communicated to each party.
- Function requester and Function provider URL SHALL have been communicated to each party.
- Function requester and Function provider SHALL agree on the Message Exchange Pattern (MEP) for response handling of asynchronous function: Asynchronous Request-Response with call back or Asynchronous with polling as defined in SGP.02 [2].
- Function requester and Function provider parties' trust must have been established on a X-509 certificate chain basis.

5.3.4.1.2 Identification/Authentication/Authorisation

If applicable on the interface, authentication of the sending party of a JSON message SHALL rely on the Transport layer security (using TLS certificate of the sending party).

5.3.4.1.3 Integrity

The integrity of the message SHALL exclusively rely on the Transport Layer Security (TLS).

5.3.4.1.4 Confidentiality

The confidentiality of the message SHALL exclusively rely on the Transport Layer Security (TLS).

5.3.4.2 Introduction to JSON

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is based on a subset of the JavaScript Programming Language. JSON is a text format that is completely language independent.

5.3.4.3 JSON message definition

The Function Requester and the Function Provider SHALL exchange the JSON objects in HTTP messages as follows:

- HTTP Request SHALL have the following format:

```
HTTP POST <HTTP Path> HTTP/1.1
Host: <Server Address>
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: <Length of the JSON requestMessage>

<JSON requestMessage>
```

- HTTP Response SHALL have the following format:

```
HTTP/1.1 <HTTP Status Code>
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: <Length of the JSON responseMessage>

<JSON responseMessage>
```

Standard HTTP errors SHALL apply to this section.

5.3.4.3.1 Definition of <JSON requestMessage>

<JSON requestMessage> is the combination of:

- <JSON requestHeader>
- <JSON body> which depends on the function called

ES9+ SHALL not contain the <JSON requestHeader>.

5.3.4.3.2 Definition of <JSON responseMessage>

<JSON responseMessage> is the combination of:

- <JSON responseHeader>
- <JSON body> which depends on the function called

The <JSON responseMessage> SHALL be empty for MEP notification message.

5.3.4.3.3 Definition of <JSON requestHeader>

The <JSON requestHeader> maps the function input header.

```
{
   "header" : {
     "type" : "object",
     "properties" : {
       "functionRequesterIdentifier" : {
         "type" : "string",
         "description" : "identification of the function requester"
       },
       "functionCallIdentifier" : {
         "type" : "string",
         "description" : "identification of the function call"
       }
     },
     "required" : ["functionRequesterIdentifier", "functionCallIdentifier"]
   }
}
```

NOTE: The Validity Period defined in SGP.02 [2] is not mapped in this specification.

5.3.4.3.4 Definition of <JSON responseHeader>

The <JSON responseHeader> maps the function output header.

```
{
   "header" : {
     "type" : "object",
     "properties" : {
       "functionExecutionStatus" : {
         "type" : "object",
         "description" : "Whether the function has been processed correctly or
not",
         "properties" : {
           "status" : {
             "type" : "string",
             "description" : " Executed-Success, Executed-WithWarning, Failed,
Expired"
           },
           "statusCodeData" : {
             "type" : "object",
             "properties" : {
               "subjectCode" : {
                 "type" : "string",
                 "description" : "OID of the subject code"
               },
               "reasonCode" : {
                 "type" : "string",
                 "description" : "OID of the reason code"
               },
               "subjectIdentifier" : {
                 "type" : "string",
```

```
                  "description" : "Identifier of the subject "
               },
               "message" : {
                  "type" : "string",
                  "description" : "Textual and human readable explanation"
               }
            },
            "required" : ["subjectCode", "reasonCode"]
         },
      },
      "required" : ["status"]
   }
},
"required" : ["functionExecutionStatus"]
}
}
```

5.3.4.4 List of functions

	Function	Path	MEP
ES2+	DownloadOrder	/gsma/rsp1/es2plus/downloadOrder	Synchronous
	ConfirmOrder	/gsma/rsp1/es2plus/confirmOrder	Synchronous
	Handle Profile Installation Result	/gsma/rsp1/es2plus/handleProfileInstallationResult	Notification
ES9+	InitiateAuthentication	/gsma/rsp1/es9plus/initiateAuthentication	Synchronous
	GetBoundProfile Package	/gsma/rsp1/es9plus/getBoundProfilePackage	Synchronous
	Handle Profile Installation Result	/gsma/rsp1/es9plus/handleProfileInstallationResult	Notification

Table 65: List of Functions

5.3.4.4.1 "ES2+.DownloadOrder" Function

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON requestMessage> corresponding to the ES2+.DownloadOrder function:

```
{
  "type" : "object",
  "properties" : {
    "eid" : {
      "type" : "string",
      "pattern" : "^[0-9]{32}$",
      "description" : "EID as desc in SGP.02"
    },
    "iccid" : {
      "type" : "string",
      "pattern" : "^[0-9]{19,20}$",
      "description" : "ICCID as desc in ITU-T E.118"
    },
    "profileType" : {
      "type" : "string",
      "description" : "content free information defined by the Operator"
    }
  }
}
```

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON responseMessage> corresponding to the ES2+.DownloadOrder function:

```
{
   "type" : "object",
   "properties" : {
     "iccid" : {
       "type" : "string",
       "pattern" : "^[0-9]{19,20}$",
       "description" : "ICCID as desc in ITU-T E.118"
     }
   },
   "required" : ["iccid"]
}
```

5.3.4.4.2 "ES2+.ConfirmOrder" Function

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON requestMessage> corresponding to the ES2+.ConfirmOrder function:

```
{
   "type" : "object",
   "properties" : {
     "iccid" : {
       "type" : "string",
       "pattern" : "^[0-9]{19,20}$",
       "description" : "ICCID as desc in ITU-T E.118"
     },
     "matchingId" : {
       "type" : "string",
       "description" : "as defined in section {5.3.2.2}"
     },
     "confirmationCode" : {
       "type" : "string",
       "description" : "as defined in section {5.3.2.2}"
     }
   },
   "required" : ["iccid"]
}
```

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON responseMessage> corresponding to the ES2+.ConfirmOrder function:

```
{
   "type" : "object",
   "properties" : {
     "matchingId" : {
       "type" : "string",
       "description" : "as defined in section {5.3.2.2}"
     },
     "smdpAddress" : {
       "type" : "string",
       "description" : "as defined in section {5.3.2.2}"
     }
   },
   "required" : ["matchingID"]
}
```

5.3.4.4.3 "ES2+.HandleProfileInstallationResult" Function

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON requestMessage> corresponding to the ES2+.HandleProfileInstallationResult function:

```
{
   "type" : "object",
   "properties" : {
     "eid" : {
        "type" : "string",
        "pattern" : "^[0-9]{32}$",
        "description" : "EID as described in SGP.02"
     },
     "iccid" : {
        "type" : "string",
        "pattern" : "^[0-9]{19,20}$",
        "description" : "ICCID as described in ITU-T E.118"
     },
     "profileType" : {
        "type" : "string",
        "description" : "content free information defined by the Operator
(e.g.‘P9054-2’)"
     },
     "completionTimestamp" : {
        "type" : "string",
        "pattern" : "$[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-
9]{2}[T,D,Z]{1}$",
        "description" : " String format as specified by W3C: YYYY-MM-
DDThh:mm:ssTZD (E.g. 2001-12-17T09:30:47Z)"
     },
     "operationStatus" : {
        "type" : "string",
        "description" : " EXECUTION STATUS Common Data Type"
     }
   },
   "required" : ["eid", "iccid", "profileType", "completionTimestamp",
"operationStatus"]
}
```

5.3.4.4.4 "ES9+.InitiateAuthentication" Function

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON requestMessage> corresponding to the ES9+.InitiateAuthentication function:

```
{
   "type" : "object",
   "properties" : {
     "euiccChallenge" : {
        "type" : "string",
        "format" : "base64",
        "description" : "base64 encoded binary data containing eUICC Challenge
defined in Section 5.3.3.1"
     },
     "svn" : {
        "type" : "string",
        "description" : "SVN in string, e.g., 1.0.0"
     },
     "euiccnfo1" : {
        "type" : "string",
```

```
        "format" : "base64",
        "description" : "base64 encoded binary data containing euiccInfo1 defined
in Section 5.2.3.4"
      },
      "smdpAddress" : {
        "type" : "string",
        "description" : "SM-DP+ Address"
      }
   },
   "required" : ["euiccChallenge", "euiccInfo1", "smdpAddress"]
}
```

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON responseMessage> corresponding to the ES9+.InitiateAuthentication function:

```
{
   "type" : "object",
   "properties" : {
      "transactionId" : {
        "type" : "string",
        "pattern" : "^[0-9,A-F]{2,32}$",
        "description" : "Hexadecimal representation of the TransactionID defined
in section 5.3.3.1"
      },
      "dpSigned1" : {
        "type" : "string",
        "format" : "base64",
        "description" : "The data object as required by ES10b.PrepareDownload"
      },
      "smdpSignature1" : {
        "type" : "string",
        "format" : "base64",
        "description" : "The signature as required by ES10b.PrepareDownload"
      },
      "certFormatTobeUsed" : {
        "type" : "string",
        "format" : "base64",
        "description" : "The certificate format to be used as required by
ES10b.PrepareDownload"
      },
      "curveTobeUsed" : {
        "type" : "string",
        "format" : "base64",
        "description" : "The curve to be used as required by
ES10b.PrepareDownload"
      },
      "smdpCertificate" : {
        "type" : "string",
        "format" : "base64",
        "description" : "The SM-DP+ Certificate as required by
ES10b.PrepareDownload"
      }
   },
   "required" : ["transactionId", "dpSigned1", "smdpSignature1",
"certFormatTobeUsed, "curveTobeUsed", "smdpCertificate"]
}
```

NOTE: LPA is in charge of transcoding the transactionId.

5.3.4.4.5 "ES9+.GetBoundProfilePackage" Function

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON requestMessage> corresponding to the ES9+.GetBoundProfilePackage function:

```
{
   "type" : "object",
   "properties" : {
     "transactionId" : {
        "type" : "string",
        "pattern" : "^[0-9,A-F]{2,32}$",
        "description" : "Hexadecimal representation of the TransactionID defined
in Section 5.3.3.2"
     },
     "prepareDownloadResponse" : {
        "type" : "string",
        "format" : "base64",
        "description" : "base64 encoded binary data containing
prepareDownloadResponse defined in Section 5.3.3.2"
     }
   },
   "required" : ["transactionId", "prepareDownloadResponse"]
}
```

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON responseMessage> corresponding to the ES9+.GetBoundProfilePackage function:

```
{
   "type" : "object",
   "properties" : {
     "transactionId" : {
        "type" : "string",
        "pattern" : "^[0-9,A-F]{2,32}$",
        "description" : "Hexadecimal representation of the TransactionID defined
in Section 5.3.3.2"
     },
     "profileMetadata" : {
        "type" : "object",
        "description" : "data containing Profile Metadata defined in section
5.3.3.2",
        "properties" : {
          "iccid" : {
             "type" : "string",
             "pattern" : "^[0-9]{19,20}$",
             "description" : "ICCID in decimal string"
          },
          "serviceProviderName" : {
             "type" : "string",
             "pattern" : "^[0-9,A-F]{0,64}$",
             "description" : "Hexadecimal representation of Service provider name
(in UTF-8 string) defined in section 4.4"
          },
          "profileName" : {
             "type" : "string",
             "pattern" : "^[0-9,A-F]{0,128}$",
             "description" : "Hexadecimal representation of Profile name (in UTF-8
string) defined in Section 4.4"
          },
          "iconType" : {
```

```
            "enum" : [0,1],
            "description" : "JPG (0), PNG (1) as defined in Section 4.4"
          },
          "icon" : {
            "type" : "string",
            "format" : "base64",
            "description" : "base64 encoded binary data containing the icon
defined in section 4.4. This SHALL be present if iconType is present"
          }
        }
     },
     "boundProfilePackage" : {
        "type" : "string",
        "format" : "base64",
        "description" : "base64 encoded binary data containing Bound Profile
Package defined in Section 5.3.3.2"
     }
   },
   "required" : ["transactionId", "profileMetadata", "boundProfilePackage"]
}
```

6 "ES9+.HandleProfileInstallationResult" Function

Hereunder is the definition of the JSON schema for the <JSON body> part of the <JSON requestMessage> corresponding to the ES9+.HandleProfileInstallationResult function:

```
{
   "type" : "object",
   "properties" : {
     "result" : {
        "type" : "string",
        "format" : "base64",
        "description" : "base64 encoded binary data containing the Result defined
in section 5.3.3.3"
     }
   },
   "required" : ["result"]
}
```

Annex A Use of GlobalPlatform Privileges (Normative)

The eUICC architecture defined in this specification relies on the ISD-R, ISD-P, MNO-SD and ECASD Security Domains defined in SGP.02 [2].

The GlobalPlatform privileges allocation defined in SGP.02 [2] are applicable for those ISD-R, ISD-P, MNO-SD and ECASD.

Annex B Data Definitions (Normative)

- Coding of the IMEI

The value of the IMEI SHALL be directly copied from Terminal Response of the Provide Local Information command (see ETSI TS 102 223 [31] and ETSI TS 124 008 [32]).

Annex C Device Requirements (Normative)

Functional Device Requirements No.	Requirement
DEV1	For connectivity the Companion Device SHALL support at least one of the network access technologies defined by 3GPP or 3GPP2. UDP over IP as defined in RFC 768 [34] (subject to the right support of access network technology) TCP over IP as defined in RFC 793 [19].
DEV2	For Network connection control the Companion Device SHALL support: • RPLMN details (LAC/TAC, NMR). • QoS (failures, duration, power, location). • New network selection after SIM/USIM update.
DEV3	The Companion Device SHALL contain a unique IMEI (International Mobile Equipment Identity) value compliant with the format defined in ETSI TS 123 003 [35] and/or a unique MEID as defined in 3GPP2 S.R0048-A [36].
DEV4	The Companion Device SHALL support, as a minimum, the following set of commands: • PROVIDE LOCAL INFORMATION (location information, IMEI, NMR, date and time, access technology, at least). • POLL INTERVAL, POLLING OFF, TIMER MANAGEMENT [at least one timer], ENVELOPE (TIMER EXPIRATION). • SET UP EVENT LIST and ENVELOPE (EVENT DOWNLOAD). REFRESH Command (At least mode 4 - "UICC reset").
DEV5	The Companion Device SHALL comply with the IMEI security requirements defined in the GSMA-EICTA document "Security Principles Related to Handset Theft" [22].
DEV6	A Companion Device SHALL be able to handle an eUICC without any installed Profiles.
DEV7	If a Companion Device does not have the capability itself to communicate directly with the SM-DP+, it SHALL use a Primary Device as a conduit, allowing it to communicate with the SM-DP+.
DEV8	At least one of the Primary or Companion Device SHALL have a UI that allows the secure capture of User Intent.
DEV9	At least one of the Primary or Companion Device SHALL have a UI that allows the user to initiate a Profile Download or Local Profile Management.

Table 66: Device requirements

Secure interaction between the Primary Device and the Companion Device

The LPA of the Companion Device SHALL support secure capture of the End User intent for the purpose of Local Profile Management through the Primary Device when the Companion Device has to rely on the Primary Device for UI function. This SHALL further include a

secure pairing and secured communication between the Primary and Companion Device, the implementation of which is the responsibility of the Primary and Companion Devices' OEM(s). The End User MAY perform local Profile download and management towards the eUICC in the Companion Device using the Primary Device.

LPA functions

The LPA SHALL support all the functions related to Profile download and loading via the LPA's Local Profile Download (LPD) functions as defined in section 3.1.2.

The LPA SHALL support retrieving Profile installation Report and Receipt from the eUICC as defined in section 5.2.3.5.

The LPA SHALL support the Local Profile Management functions via LPA's Local User Interface (LUI) function as defined in section 3.2.

- Initiate a Profile download session with SM-DP+ as defined in section 3.1.2,
- Enabling a Disabled Profile as defined in section 3.2.1,
- Disabling an Enabled Profile as defined in section 3.2.2,
- Delete a Profile as defined in section 3.2.3,
- Query the Metadata and states of Profiles installed on the eUICC as defined in section 3.2.4,
- Perform eUICC memory reset, as defined in section 3.3.2,
- Process the Profile installation Result as defined in sections 2.5.6, 3.1.2 and 5.2.3.5,
- Add/Update Profile Nicknames associated with installed Profiles as defined in section 3.2.6,
- Get eUICC info as defined in section 4.3,

LPA Functions and Security Protection

The mechanism for End User intent verification and security protection is out of scope for this release.

As examples, the recommended End User Intent verification could include:

- Biometric (e.g. fingerprint) verification, or
- Device passcode verification, or
- Hard-wired End User input that bypasses Device application processor.

NOTE: End User Intent Verifications MAY be combined to simplify the User Experience and avoid repeated input steps for the End User – For instance, in the Activation Code procedure, the Verification for download and Verification for Enabling the Profile MAY be combined in one single step e.g. "Do you want to download and enable Profile XYZ – YES/NO" – In the case of combined verifications, it SHALL be clear to the End User what Operations will be performed.

The proper security level associated with LPA functions SHOULD be ensured based on industry-proven implementations of:

- A secure boot OS.
- An implementation-dependent software/hardware secure execution environment for capturing, storing and verifying the passcode or biometric input.
- Verification of proper OEM signature of LPA related software components.
- Application-level secure pairing and un-pairing methods between Primary and Companion Devices. This MAY be independent of pairing technologies and associated link layer security (e.g. Bluetooth or Wi-Fi).

Annex D Coding of the AIDs for 'Remote SIM Provisioning' (Normative)

The Coding of the AID for ISD-R, ISD-P and ECASD are defined in SGP.02 [2].

Annex E List of Identifiers (Informative)

OIDs

The following identifiers for remote provisioning are created under a dedicated OID tree under ISO branch:

- ASN.1 notation: {ISO(1) identified-organization(3) dod(6) internet(1) private(4) enterprise(1)}
- dot notation: 1.3.6.1.4.1
- IOD-IRI notation: /ISO/Identified-Organization/6/1/4/1

The private enterprise numbers may be found under the Internet Assigned Numbers Authority: http://www.iana.org/assignments/enterprise-numbers/enterprise-numbers

EUM Identifiers

Identifier	Uniqueness	Registration Entity
EUM OID	within the ecosystem	ISO 1.3.6.1.4.1
SIN	within the ecosystem	ISO 7812 [37]

Table 67: EUM Identifiers

eUICC Identifiers

Identifier	Uniqueness	Registration Entity
EID	within the ecosystem	GSMA ESIM Technical Specification SGP.02 [2]
ECASD AID	within the eUICC	GSMA ESIM Technical Specification SGP.02 [2]
ISD-R AID	within the eUICC	GSMA ESIM Technical Specification SGP.02 [2]
ISD-P AID	within the eUICC	eUICC within a range defined in GSMA ESIM Technical Specification SGP.02 [2]
ICCID	Global	ITU-T E.118 [21]
ISD-R TAR	within the eUICC	GSMA ESIM Technical Specification SGP.02 [2]
MNO-SD AID	Within the Profile	ETSI TS 101 220 [33]
MNO-SD TAR	Within the Profile	ETSI TS 101 220 (ISD TAR) [33]

Table 68: eUICC Identifiers

SM-DP+ Identifier

Identifier	Uniqueness	Registration Entity
SMDP OID	within the ecosystem	ISO 1.3.6.1.4.1

Table 69: SM-DP+ Identifier

MNO Identifiers

Identifier	Uniqueness	Registration Entity
MNO OID	within the ecosystem	ISO 1.3.6.1.4.1
MCC+MNC (IMSI)	Global	ITU-T for MCC and National Regulators for MNC

Table 70: MNO Identifiers

Annex F Profile Eligibility Check (Informative)

Prior to any Profile download, the Operator or the SM-DP+ verifies if the selected Profile Type is compatible with the targeted Device.

Two types of checking are possible:

- Static eligibility check (SEC): a check based on the static capabilities of the Device and / or the eUICC. These capabilities could be retrieved based on the knowledge of the EID and the TAC. These eUICC capabilities MAY be acquired by various means: information contained in the EID itself, additional tables locally handled by the Operator or communication with an external entity like the EUM. Device capabilities can be retrieved by the Operator based on the TAC. This Static eligibility check is under the responsibility of the Operator; it MAY be done by the SM-DP+ on behalf of the Operator. The means to establish the compatibility of the Profile Type with a Device type and eUICC type is out of scope of this specification.
- Dynamic eligibility check (DEC): a check based on the eUICC Info and / or the Device capabilities signed by the eUICC during Profile Download and Installation procedure. This Dynamic eligibility check is under the responsibility of the SM-DP+ on behalf of the Operator.

The following : Eligibility Check" describes the global eligibility process depending on the knowledge of the target Device.

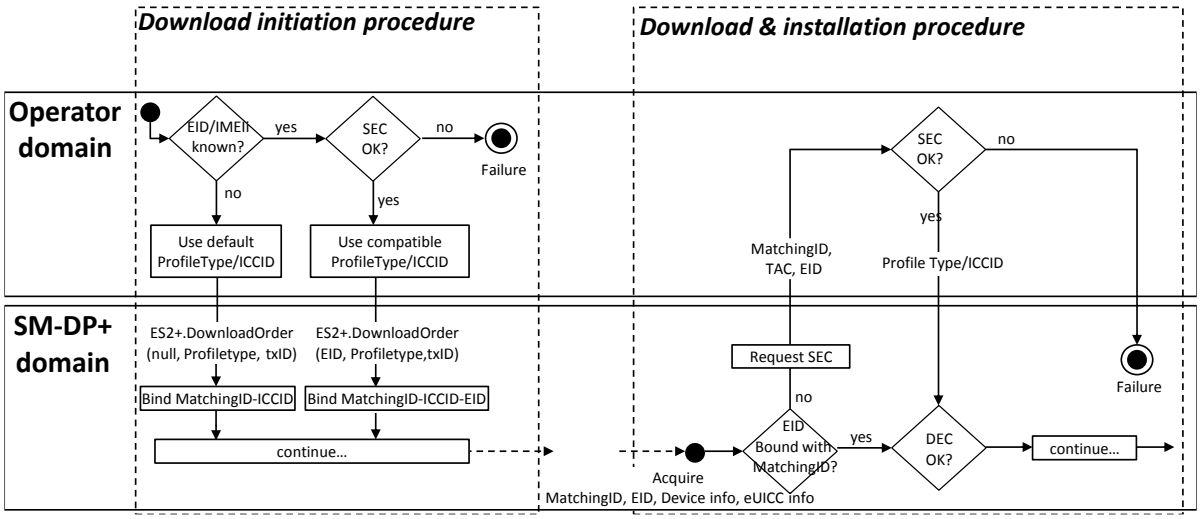

Figure 21: Eligibility Check

Annex G Key Derivation Process (Normative)

The key derivation process described in this section SHALL be executed by both off-card entity (SM-DP+) and eUICC in a symmetric way.

- Use otPK.eUICC.ECKA and otPK.DP.ECKA (with their respective one time private keys) to generate the shared secret ShS as described in GlobalPlatform Card Specification Amendment F [13] section 3.1.1 (but limited to ephemeral keys) which constitutes the input for the Key Derivation process.
- Concatenates the following values as SharedInfo as input for the Key Derivation process (these data are the one given as input data in the function ES8+.InitialiseSecureChannel):
  - Key type (1 byte)
  - Key length (1 byte)
  - HostID-LV and EID-LV. HostID-LV comprises the length and the value field of the HostID given in the input data; EID-LV comprises the length and value field of the EID.
- Initial MAC Chaining value, S-ENC, S-MAC and S-RMAC are taken from KeyData derived from the ShS as defined in [TR 03111] for the “X9.63 Key Derivation Function” (SHA-256 SHALL be used for the key derivation to calculate KeyData of sufficient length). This key derivation includes additional information, the “SharedInfo” of the key derivation algorithm. Keys are assigned following **Error! Reference source not found.**:

KeyData	Key
1 to L	Initial MAC chaining value
L+1 to 2L	S-ENC
2L+1 to 3L	S-MAC
3L+1 to 4L	S-RMAC

Table 71: Mac Chaining

The initial MAC chaining value is used for the computation of the MAC of the first SCP03t block following the ES8+.InitialiseSecureChannel command.

Annex H ASN.1 Definitions (Normative)

```
RSPDefinitions {joint-iso-itu-t(2) international-organizations(23) gsma(tbc) rsp(1)
spec-version(1) version-one(1)}
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN

IMPORTS Certificate FROM PKIX1Explicit88 {iso(1) identified-organization(3) dod(6)
internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-pkix1-explicit(18)};

-- Basic types, for size constraints
EIDPrefix ::= OCTET STRING (SIZE(9))
Octet8 ::= OCTET STRING (SIZE(8))
Octet16 ::= OCTET STRING (SIZE(16))
OctetTo16 ::= OCTET STRING (SIZE(1..16))
Octet32 ::= OCTET STRING (SIZE(32))
Octet1 ::= OCTET STRING(SIZE(1))
VersionType ::= OCTET STRING(SIZE(3)) -- major/minor/revision version are coded on
byte 1/2/3


-- Definition of EUICCInfo1 --------------------------
EUICCInfo1 ::= [32] SEQUENCE { -- Tag 'BF20'
   eUICCVerSupport [2] VersionType,      -- GSMA SGP.22 version supported
   certificateInfo [5] CertificateInfo,
   curveSigningSupport [6] CurveSigningSupport, -- coded as a sequence of Key
parameter reference values  as defined in GlobalPlatform Amendment E [AmdE] and
Table 13
   curveVandKASupport [7] CurveVandKASupport -- coded as a sequence of Key
parameter reference values  as defined in GlobalPlatform Amendment E [AmdE] and
Table 13
}

EUICCInfo2 ::= [34] SEQUENCE {-- Tag 'BF22'
   eidPrefix [0] EIDPrefix,              -- EID Prefix, first 9 Byte of EID
   profileVersion [1] VersionType,       -- Profile package version supported
   euiccVerSupport [2] VersionType,      -- GSMA SGP.22 version supported
   euiccFirmwareVer [3] VersionType,     -- eUICC Firmware version
   extCardResource [4] OCTET STRING,     -- Extended Card Resource Information
according to ETSI TS 102 226
   certificateInfo [5] CertificateInfo,
   curveSigningSupport [6] CurveSigningSupport, -- coded as a sequence of Key
parameter reference values  as defined in GlobalPlatform Amendment E [AmdE] and
Table 13
   curveVandKASupport [7] CurveVandKASupport, -- coded as a sequence of Key
parameter reference values  as defined in GlobalPlatform Amendment E [AmendE] and
Table 13
   euiccCapability [8] EUICCCapability
}

CertificateInfo ::= BIT STRING {

   certSigningGp(0),       -- eUICC has a CERT.EUICC.ECDSA in GlobalPlatform format
   certSigningX509(1),     -- eUICC has a CERT.EUICC.ECDSA in X.509 format
   rfu2(2),
   rfu3(3),
   certVerificationGp(4), -- Handling of Certificate in GlobalPlatform format
   certVerificationX509(5)-- Handling of Certificate in X.509 format
}

CurveSigningSupport ::= OCTET STRING -- Supported curves for ECDSA signature
creation

CurveVandKASupport ::= OCTET STRING -- Supported curves for ECDSA signature
verification and for ECKA key agreement
```

```
-- Definition of EUICCCapability
EUICCCapability ::= BIT STRING {
   contactlessSupport(0),  -- Contactless (SWP, HCI and associated APIs)
   usimSupport(1),         -- USIM as defined by 3GPP
   isimSupport(2),         -- ISIM as defined by 3GPP
   csimSupport(3),         -- CSIM as defined by 3GPP2

   akaMilenage(4),         -- Milenage as AKA algorithm
   akaCave(5),             -- CAVE as authentication algorithm
   akaTuak128(6),          -- TUAK as AKA algorithm with 128 bit key length
   akaTuak256(7),          -- TUAK as AKA algorithm with 256 bit key length
   rfu(8),                 -- reserved for further algorithms
   rfu(9),                 -- reserved for further algorithms

   gbaAuthenUsim(10),      -- GBA authentication in the context of USIM
   gbaAuthenIsim(11),      -- GBA authentication in the context of ISIM
   mbmsAuthenUsim(12),     -- MBMS authentication in the context of USIM
   eapClient(13),          -- EAP client

   javacard(14),           -- JavaCard support
   multos(15)              -- Multos support

   multipleUsimSupport(16), -- Multiple USIM applications are supported within the
same Profile
   multipleIsimSupport(17),  -- Multiple ISIM applications are supported within the
same Profile
   multipleCsimSupport(18)   -- Multiple CSIM applications are supported within the
same Profile
}

-- Definition of DeviceInfo
DeviceInfo ::= SEQUENCE {
   tac Octet8,
   deviceCapabilities DeviceCapabilities
}

DeviceCapabilities ::= SEQUENCE { -- Highest supported release for each definition
  -- The device SHALL set all the capabilities it supports
   gsmSupportedRelease Octet1 OPTIONAL,
   utranSupportedRelease Octet1 OPTIONAL,
   cdma2000onexSupportedRelease Octet1 OPTIONAL,
   cdma2000hrpdSupportedRelease Octet1 OPTIONAL,
   cdma2000ehrpdSupportedRelease Octet1 OPTIONAL,
   eutranSupportedRelease Octet1 OPTIONAL,
   contactlessSupportedRelease Octet1 OPTIONAL
}

-- Definition of ProfileInfo
ProfileInfo ::= [PRIVATE 3] SEQUENCE {  -- Tag 'E3'
       iccid [APPLICATION 26] OCTET STRING (SIZE(10)) OPTIONAL, -- ICCID as coded in EFiccid,
corresponding tag is ‘5A’
       isdpAid [APPLICATION 15] OctetTo16 OPTIONAL, -- AID of the ISD-P containing the
Profile, tag ‘4F’
       profileState [112] ProfileState OPTIONAL, -- Tag ‘9F70’
       profileNickname [16] UTF8String (SIZE(0..64)) OPTIONAL, -- Tag ‘90’
       serviceProviderName [17] UTF8String (SIZE(0..32)) OPTIONAL, -- Tag ‘91’
       profileName [18] UTF8String (SIZE(0..64)) OPTIONAL, -- Tag ‘92’
       iconType [19] IconType OPTIONAL, -- Tag ‘93’
       icon [20] OCTET STRING (SIZE(0..1024)) OPTIONAL -- Tag '94'
}

-- Definition of StoreMetadata request
StoreMetadataRequest ::= [37] SEQUENCE {  -- Tag 'BF25'
       serviceProviderName [17] UTF8String (SIZE(0..32)), -- Tag '91'
       profileName [18] UTF8String (SIZE(0..64)), -- Tag '92'
       iconType [19] IconType OPTIONAL, -- Tag '93'
       icon [20] OCTET STRING (SIZE(0..1024)) OPTIONAL -- Tag '94'
}

IconType ::= INTEGER {jpg(0), png(1)}
ProfileState ::= INTEGER {disabled(0), enabled(1)}
```

```
-- Definition of StoreMetadata response
StoreMetadataResponse ::= [37] SEQUENCE {  -- Tag 'BF25'
   errorCode[1] Octet1 OPTIONAL -- tag '81'
}

-- Definition of data objects for command PrepareDownload -------------------------
PrepareDownloadRequest ::= [33] SEQUENCE {  -- Tag 'BF21'
   dpSigned1 DPSigned1,                  -- Signed information
   smdpSignature1 [APPLICATION 55] OCTET STRING,      -- DP_Sign1, tag '5F37'
   activationCodeToken UTF8String,
   deviceInfo DeviceInfo,                -- The Device information
   smdpOid OBJECT IDENTIFIER OPTIONAL,   -- SM-DP+ OID (same value as in
CERT.DP.ECDSA)
   hashCc Octet32 OPTIONAL, -- Hash of confirmation code
   certFormatTobeUsed CertToBeUsed,       -- Certificate Format to be used by eUICC
for signing
   curveTobeUsed Octet1,          -- Curve to be used, coded as a Key Parameter
Reference value as defined in GlobalPlatform Amendment E [AmdE] and {section 2.11}
   smdpCertificate RSPCertificate -- CERT.DP.ECDSA in (one of) the format(s)
requested by eUICC for signature verification (GP or X.509).
}

DPSigned1 ::= SEQUENCE {
   euiccChallenge Octet16, -- The eUICC Challenge
   smdpChallenge Octet16,  -- The SM-DP+ Challenge
   transactionId OctetTo16,-- The TransactionID generated by the SM-DP+
   smdpAddress UTF8String         -- SM-DP+ address
}

CertToBeUsed ::= INTEGER {gp(1), x509(2)}

PrepareDownloadResponse ::= [33] SEQUENCE {  -- Tag 'BF21'
   euiccSigned1 EUICCSigned1,     -- Signed information
   euiccSignature1 [APPLICATION 55] OCTET STRING,     -- EUICC_Sign1, tag '5F37'
   euiccCertificate RSPCertificate, -- eUICC Certificate (CERT.EUICC.ECDSA) in the
requested format with the requested Key Parameter Reference value
   eumCertificate RSPCertificate  -- EUM Certificate (CERT.EUM.ECDSA) in the
requested format with the requested Key Parameter Reference value
}

EUICCSigned1 ::= SEQUENCE {
   smdpChallenge Octet16,  -- The SM-DP+ Challenge
   transactionId OctetTo16,
   smdpAddress UTF8String,
   activationCodeToken UTF8String,
   deviceInfo DeviceInfo,
   smdpOid OBJECT IDENTIFIER OPTIONAL,
   hashCc Octet32 OPTIONAL,              -- Hash of confirmation code
   otpk OCTET STRING,             -- otPK.EUICC.ECKA
   euiccInfo2 EUICCInfo2
}
-- Definition of Certificate
RSPCertificate ::= CHOICE{
   gp [0] OCTET STRING, -- This contain the Certificate in GlobalPlatform format
(the whole TLV structure) as an octet string starting with '7F21'
   x509 [1] Certificate
}

-- Definition of Profile Nickname Information
SetNicknameRequest ::= [41] SEQUENCE {  -- Tag 'BF29'
   iccid [APPLICATION 26] OCTET STRING (SIZE(10)),
   profileNickname [16] UTF8String (SIZE(0..64))
}

id-rsp OBJECT IDENTIFIER ::= {joint-iso-itu-t(2) international-organizations(23)
gsma(tbc) rsp(1) spec-version(1) version-one(1)}

id-rspExt OBJECT IDENTIFIER ::= {id-rsp 0}
```

```
id-rspRole OBJECT IDENTIFIER ::= {id-rsp 1}

-- Definition of the extension for Extended GSMA SAS Accreditation Serial Number
id-rsp-sasSn OBJECT IDENTIFIER ::= {id-rspExt 0}
SasSn::= UTF8String (SIZE(0..64))

-- Definition of OIDs for role identification
id-rspRole-ci OBJECT IDENTIFIER ::= {id-rspRole 0}
id-rspRole-euicc OBJECT IDENTIFIER ::= {id-rspRole 1}
id-rspRole-eum OBJECT IDENTIFIER ::= {id-rspRole 2}
id-rspRole-dp-pb OBJECT IDENTIFIER ::= {id-rspRole 4}


END
```

Annex I JSON Request Response Examples (Informative)

An example for the ES9+.InitiateAuthentication function is shown as follows:

- HTTP Request (from LPA to SM-DP+):

```
HTTP POST /gsma/rsp1/es9plus/initiateAuthentication HTTP/1.1
Host: smdp.gsma.com
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: XXX

{
   "euiccChallenge" : "ZVVpY2NDaGFsbGVuZ2VFeGFtcGxlQmFzZTY0oUFZuQnNZVE5D",
   "svn" : "1.0.0",
   "euiccInfo1" : "RmVHRnRjR3hsUW1GelpUWTBvVUZadVFuTlpWRTU",
   "smdpAddress" : "smdp.gsma.com"
}
```

- HTTP Response (from LPA to SM-DP+ to):

```
HTTP/1.1 200 OK
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: XXX

{
   "header" : {
      "functionExecutionStatus" : {
         "status" : "Executed-Success"
      }
   },
   "transactionId" : "0123456789ABCDEF",
   "dpSigned1" : "RKNFZsbFVUa05qUm14e",
   "smdpSignature1" : "RKNFZsbFVUa05qUm14e",
   "certFormatTobeUsed" : "MQ==",
   "curveTobeUsed" : "MDM=",
   "smdpCertificate" : "RUU2NTQ0ODQ5NDA0RlpSRUZERA=="
}
```

An example for the ES2+.DownloadOrder function is shown as follows:

- HTTP Request (from Operator to SM-DP+):

```
HTTP POST /gsma/rsp1/es2plus/downloadOrder HTTP/1.1
Host: smdp.gsma.com
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: XXX

{
   "header" : {
      "functionRequesterIdentifier" : "RequesterID",
      "functionCallIdentifier" : "TX-567"
   },
   "eid" : "01020300405060708090A0B0C0D0EOF",
   "iccid" : "01234567890123456789",
   "profileType" : "myProfileType"
}
```

- HTTP Response for a successful execution:

```
HTTP/1.1 200 OK
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: XXX

{
   "header" : {
     "functionExecutionStatus" : {
        "status" : "Executed-Success"
     }
   },
   "iccid" : "01234567890123456789"
}
```

- HTTP Response for a failed execution:

```
HTTP/1.1 200 OK
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: XXX

{
   "header" : {
     "functionExecutionStatus" : {
        "status" : "Failed",
        "statusCodeData” : {
           "subjectCode" : "8.2.5",
           "reasonCode" : "3.7",
           "message" : “No more Profile”
        }
     }
   }
}
```

An example for the ES2+.HandleProfileInstallationResult function is shown as follows:

- HTTP Request:

```
HTTP POST /gsma/rsp1/es2plus/handleProfileInstallationResult HTTP/1.1
Host: smdp.gsma.com
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: XXX

{
   "header" : {
     "functionRequesterIdentifier" : "RequesterID",
     "functionCallIdentifier" : "TX-567",
   },
   "eid" : "01020300405060708090A0B0C0D0EOF",
   "iccid" : "01234567890123456789",
   "profileType" : "myProfileType",
   "completionTimeStamp" : "2015-12-16T09:30:47Z",
   "operationStatus" : "Executed-Success"
}
```

- HTTP Response for a successful execution:

```
HTTP/1.1 202 ACCEPT
X-Admin-Protocol: gsma/rsp1
Content-Type: application/json
Content-Length: 0
```

Annex J Tag allocation (Normative)

This Annex recaps the tags allocated to data objects used for the definition of the eUICC functions.

Tag	Data name
'9F26'	ReplaceSessionKeyRequest
‘BF20’	EUICCInfo1
'BF21'	PrepareDownloadRequest or PrepareDownloadResponse
'BF22'	EUICCInfo2
'BF23'	InitialiseSecureChannelRequest or InitialiseSecureChannelResponse
'BF24'	ConfigureISDPRequest or ConfigureISDPResponse
'BF25'	StoreMetadataRequest or StoreMetadataResponse
'BF26'	ReplaceSessionKeyResponse
'BF27'	ProfileInstallationReceipt
'BF28'	ProfileInstallationResult
'BF29'	SetNicknameRequest
'BF2A'	UpdateMetadataRequest
'E3'	ProfileInfo

DGI	Data name
'3A03'	EnableProfile
'3A04'	DisableProfile
'3A09'	eUICCMemoryReset

Annex K Document Management

K.1 Document History

Version	Date	Brief Description of Change	Approval Authority	Editor / Company
V1.0	13 January 2016	New PRD Publication	PSMC/SGP	Duncan Macadam GSMA
V.1.1	14 April 2016	Minor Change to fix bugs Phase one	PSMC/SGP	Yolanda Sanz GSMA

Other Information

Type	Description
Document Owner	Yolanda Sanz
Editor / Company	GSMA

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