



**TMCOS 4.0 0.3 with eUICC
extension on THN31 1.0.2 Secure Element
Security Target Lite**

Revision History

Revision number	Date	Description
1.0	2025-10-30	- ST Lite Public Release
1.1	2025-11-20	- Re-add SPD for eUICC and some tables
1.2	2025-12-20	- Align with ST release V1.13

Content

1	ST Introduction	6
1.1	ST Reference	6
1.2	TOE Reference	6
1.3	TOE Identification	6
1.4	TOE Overview	7
1.4.1	TOE Type	8
1.4.2	Usage and Major Security Features of the TOE	8
1.4.3	Non-TOE Hardware/Software/Firmware Required	10
1.5	TOE Description	11
1.5.1	Physical scope	11
1.5.2	Logical Scope	12
1.5.3	Life Cycle	19
1.5.4	TOE Delivery	23
2	Conformance Claims	24
2.1	CC Conformance Claim	24
2.2	PP Claim	24
2.2.1	GSMA SGP.25- Embedded UICC for Consumer Devices Protection Profile	24
2.3	Conformance Claim Rationale	25
2.3.1	Rationale for added elements	25
2.3.2	Rationale for eUICC component	26
3	Security Aspects	29
3.1	Confidentiality	29
3.2	Integrity	29
3.3	Unauthorized Executions	30
3.4	Bytecode Verification	31
3.5	Card Management	31
3.6	Services	34
3.7	OS Update	35
4	Security Problem Definition	36

4.1 SPD for Java Card System	36
4.1.1 Assets	36
4.1.2 Threats	37
4.1.3 Organizational Security Policies	42
4.1.4 Assumptions	42
4.2 SPD for eUICC	42
4.2.1 Assets for eUICC	42
4.2.2 Users / Subjects for eUICC	45
4.2.3 Threats for eUICC	46
4.2.4 Organizational Security Policies for eUICC	51
4.2.5 Assumptions for eUICC	52
5 Security Objectives	54
5.1 Security Objectives for the TOE	54
5.1.1 Security Objectives for Java Card System	54
5.1.2 Security Objectives for eUICC	60
5.2 Security Objectives for the Operational Environment	60
5.2.1 Operational Environment of Java Card System	60
5.2.2 Operational Environment of eUICC	61
5.3 Security Objectives Rationale	63
5.3.1 Java Card System	63
5.3.2 Security Objective Rational related to eUICC	78
6 Extended Components Definition	87
7 Security Requirements	88
7.1 Security Requirements for Java Card System	88
7.1.1 Security Functional Requirements	88
7.1.2 Security Requirements Rationale	130
7.1.3 Rationale Tables of Security Objectives and SFRs	135
7.1.4 Dependencies for Java Card System	140
7.2 Security Requirements for eUICC	144
7.2.1 Security Policies	144
7.2.2 Security attributes used in SFRs	144

7.2.3 Security Functional Requirements for eUICC	146
7.2.4 Security Requirements Rationale for eUICC	165
7.2.5 Dependencies for eUICC	165
8 Security Assurance Requirements	170
8.1 SARs Dependencies	170
8.2 Rationale for the Security Assurance Requirements	171
8.2.1 ALC_DVS.2 Sufficiency of security measures	171
8.2.2 AVA_VAN.5 Advanced methodical vulnerability analysis	171
9 TOE Summary Specification	172
9.1 TOE Security Functionality	172
9.1.1 SFs For Java Card System	173
9.1.2 SFs for eUICC	181
10 Composition with IC	183
10.1 Statement of compatibility - Threats part	183
10.2 Statement of compatibility - OSPs part	184
10.3 Statement of compatibility - Assumptions part	184
10.4 Statement of compatibility - Security objectives for the environment part	185
10.5 Statement of compatibility - Security objectives part	185
10.6 Statement of compatibility - SFRs part	186
11 References	187
11.1 Glossary of vocabulary	187
11.2 References	190

1 ST Introduction

1.1 ST Reference

Content	Information
Title:	TMCOS 4.0 0.3 with eUICC extension on THN31 1.0.2 Secure Element Security Target Lite
Version:	1.2
Author:	TSINGTENG Microsystem Co., Ltd
Publication Date:	2025/12/20

Table 1:ST Reference

1.2 TOE Reference

Content	Version
Product Type:	eUICC
TOE Name:	TMCOS 4.0 0.3 with eUICC extension on THN31 1.0.2 Secure Element
TOE Version:	V4.0 0.3

Table 2:TOE Reference

1.3 TOE Identification

Content	Version
TOE Version	V4.0 0.3
COS Name	TMCOS 4.0 0.3
Hardware	THN31 1.0.2 Secure Element version 1.0

Table 3:TOE Identification

1.4 TOE Overview

The TMCOS 4.0 0.3 with eUICC extension on THN31 1.0.2 [\[THN31P\]](#) Secure Element combines a Java Card eUICC Operating System and a Java Card Secure Element (eSE) Operating System with a NFC Controller on a single die.

These two OSs have isolated independent Java Card Systems and Global Platform (both in terms of code and data), and they also have shared code components consisting of

- Memory Management,
- Cryptographic Primitives,
- Communication.

[Figure 1](#) provides an overview of the TOE and its place in the system.

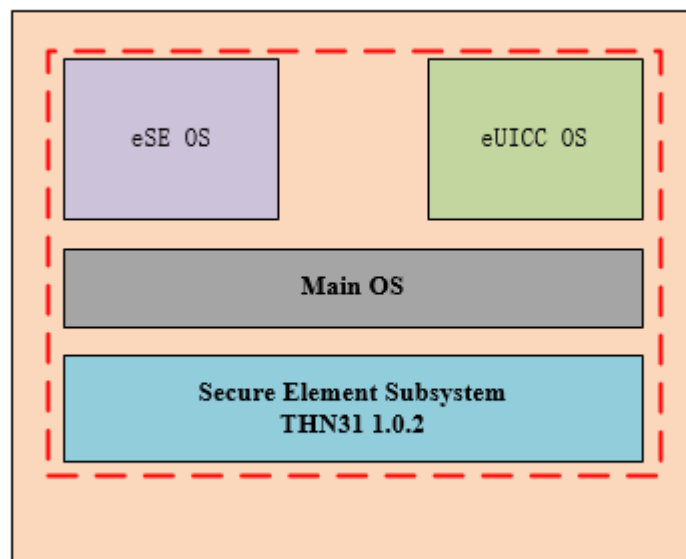


Figure 1: TOE Boundary

The Main OS creates separate and independent code and data space for eUICC OS and eSE OS, and assigns separate, independent communications channels for them.

1.4.1 TOE Type

The TOE is the embedded UICC (eUICC) software that implements the GSMA Remote SIM Provisioning (RSP) Architecture for Consumer Devices and is deployed on a certified IC hardware.

The eUICC is a component in a consumer device. The eUICC is connected to a given mobile network, by the means of its currently enabled MNO Profile.

The TOE includes the following components:

- The RSP Technical Specification V3.1 [\[SGP 22\]](#)
- eUICC Profile Package: Interoperable Format Technical Specification [\[Profile PKG\]](#)
- The Java Card System Platform 3.1 – Open Configuration
- The Card Manager according to Global Platform 2.3.1 [\[GP\]](#)
- The Native Operating System which is the native part of the TMCOS
- The underlying IC Hardware THN31 1.0.2 Secure Element

This TOE communicates with external entities over ISO-7816 interface.

Note: The TMCOS 4.0 0.3 with eUICC extension on THN31 1.0.2 [\[THN31P\]](#) Secure Element, supports MEP feature defined [\[SGP 22\]](#) v3.0, and is within the scope of TOE.

1.4.2 Usage and Major Security Features of the TOE

The TOE provides a variety of security features. These security features encompass encryption, decryption, signature generation, signature verification, key generation, secure management of PINs and secure storage of confidential data. In addition, the certified IC hardware also provides security features in hardware level to prevent physical attacks.

The TOE includes the following features:

- Cryptographic algorithms and functionality
 - AES for encryption/decryption
 - Triple-DES (3DES) for encryption/decryption
 - MAC generation and verification
 - ECC for signature, verification, point multiplication and key pair generation
 - DRNG (Pseudo Random Number Generator) / TRNG (Random Number Generator)
 - ECDSA signature generation and verification
 - ECDSA key generation
 - SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE 256 algorithms
 - HMAC-SHA (224,256,384,512)

-
- eUICC authentication functions (MILENAGE, TUAK)
 - ECDH key exchange
 - Diffie Hellman Key Agreement
 - Data Protection Module for a secure storage of the sensitive data
- Java Card 3.1.0 functionality:
 - Executing Java Card bytecodes
 - Managing memory allocation of code and data of applets
 - Enforcing access rules between applets and the JCRE
 - Mapping of Java method calls to native implementations of e.g. cryptographic operation
 - Garbage Collection fully implemented with complete memory reclamation including compaction
 - Persistent Memory Management and Transaction Mechanism
 - Static Resources in CAP files
- GlobalPlatform 2.3.1 functionality:
 - Loading of Java Card packages
 - Instantiating applet instances
 - Java package deletion
 - Java applet instance deletion
 - Creating Supplementary Security Domains
 - Associating applets to Security Domains
 - Installation of keys
 - Verification of signatures of signed applets
 - CVM Management (Global PIN) fully implemented
 - Secure Channel Protocol is supported (SCP02/SCP03)
 - Delegated Management, DAP (ECC 256)
 - Supported Amendments A (scenario 1 and 3), B, D, E, F
- RSP 3.1 functionality:
 - Remote Provisioning
 - Local Profile Management
 - Local eUICC Management
 - Remote Profile Management
 - Profile Content management
 - eUICC Root Public Key Update
 - Device Change and Profile Recovery
- TMS Proprietary Functionality:

1.4.3 Non-TOE Hardware/Software/Firmware Required

Non-TOE is same as the ones mentioned in the [\[PP eUICC\], except for the Embedded Software, IC and RTE that are part of the TOE.](#)

Since this TOE uses the Java Card System as the RTE(Runtime Environment) for eUICC functionalities, the ‘bytecode verifier’ mentioned in the Non-TOE related chapter of the [PP JCS] is also a required off card Non-TOE component to this TOE.

1.4.3.1 Non-Evaluated Security Functionality

The following security functionalities are supported by the TOE but not in the evaluation scope:

- SHA-1, SHA-224, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE 256 algorithms
- HMAC-SHA(224,384,512)
- Diffie Hellman KeyAgreement operations
- DRNG (Pseudo Random Number Generator)

1.5 TOE Description

1.5.1 Physical scope

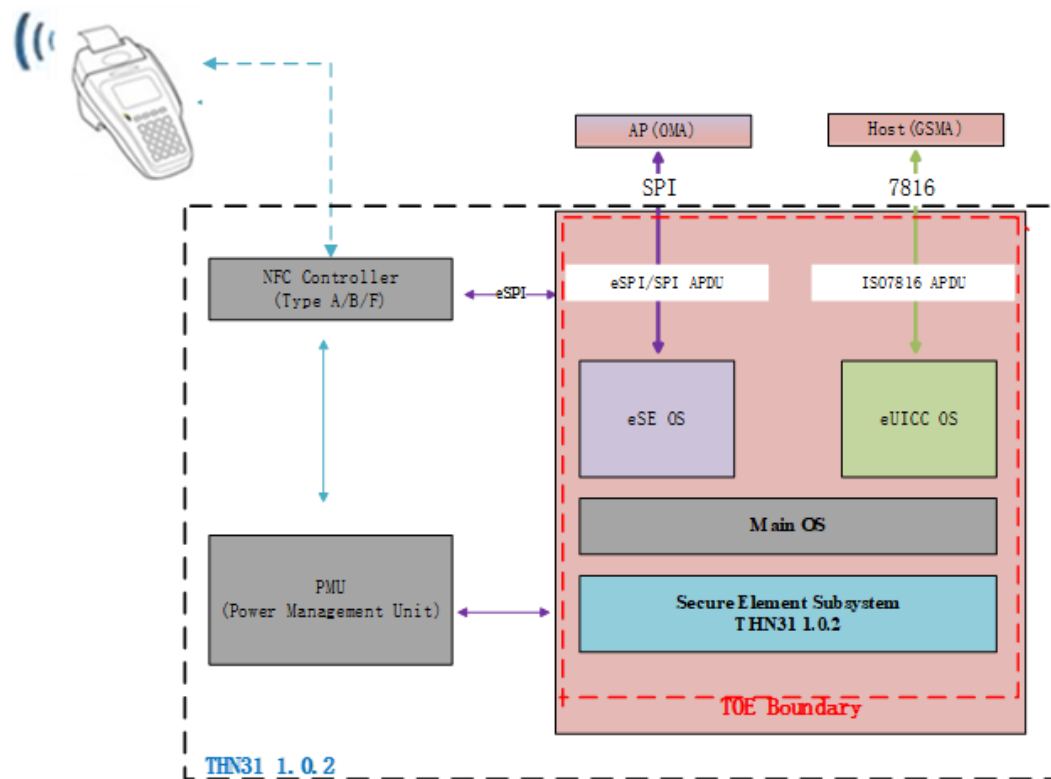


Figure 2: The TOE physical scope

The TOE of this Security Target is the embedded UICC (eUICC) software that implements the GSMA Remote SIM Provisioning (RSP) Architecture for Consumer Devices. In all, the physical scope consists of

- eUICC OS (Java Card System Platform, Global Platform and eUICC Application included)
- eSE OS (Java Card System Platform, Global Platform included)
- Main OS
- Secure Element Subsystem

Java Card RMI is not supported by the TOE.

Category	Component	Version	Delivery from
HW	THN31 1.0.2 Secure Element	1.0	Diced wafer
FW	Crypto Library	1.0.1	Masked in ROM
SW	TMCOS	4.0 0.3	Pre-Install in NVM
DOC	[AGD_OPE]	[AGD_OPE]	docx file
DOC	[AGD_PRE]	[AGD_PRE]	docx file

Table 4:Reference to physical scope's components

1.5.2 Logical Scope

The architecture of the TOE composed of various components is illustrated in the following diagram:

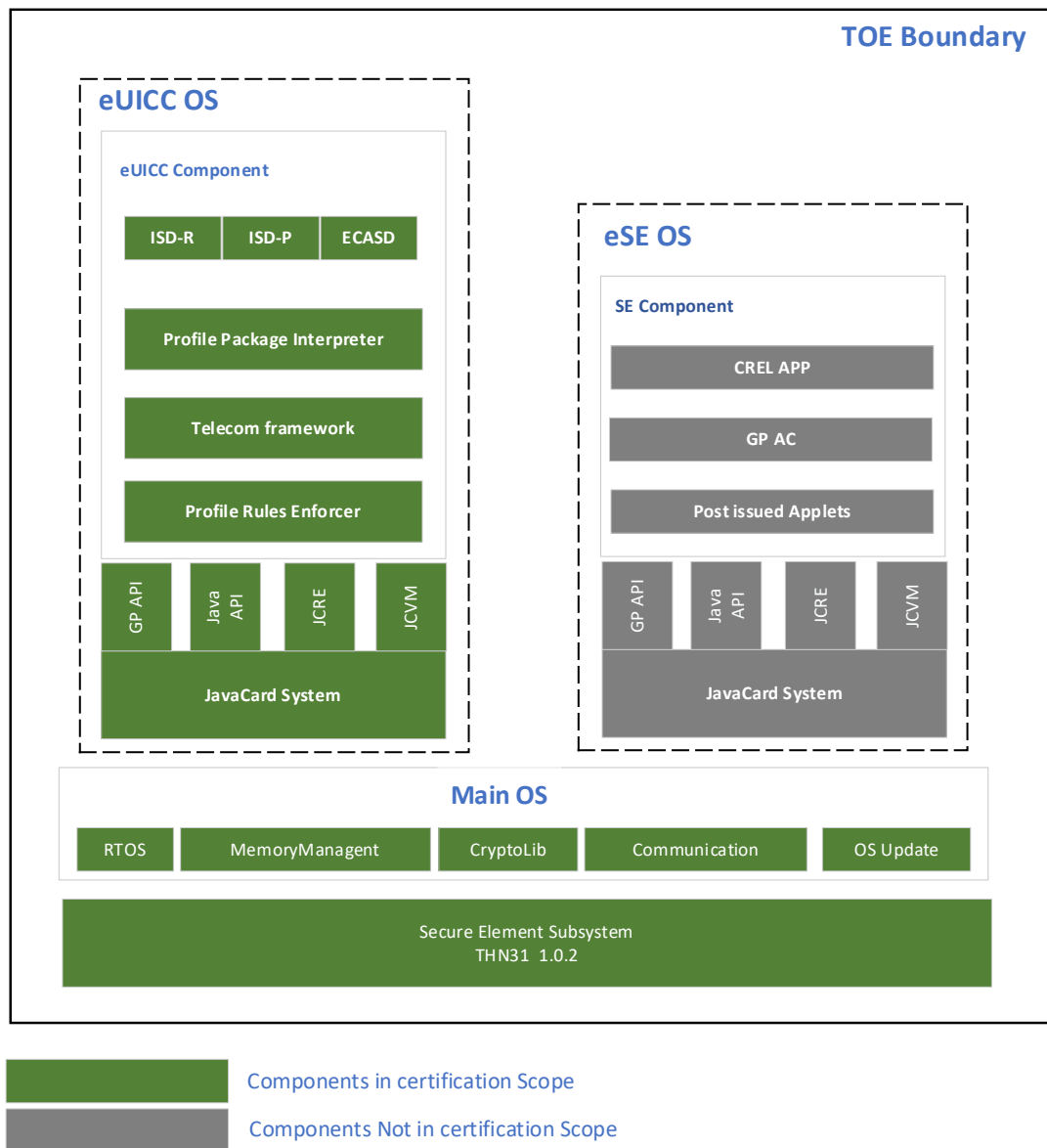


Figure 3:The TOE logical Scope

1.5.2.1 Hardware and IC dedicated software

The certified [\[THN31P\]](#) Secure Element is composed of hardware and IC dedicated software.

The hardware is based on a 32-bit secure CPU with ROM (Non-Volatile Read-Only Memory), NVM (Non-Volatile Programmable Memory) and RAM (Volatile Memory). The Hardware also incorporates communication peripherals and cryptographic coprocessor for execution and acceleration of symmetric and asymmetric cryptographic algorithms.

The features provided by the hardware are listed below:

- Hardware coprocessor for TDES;
- Hardware for RSA-CRT support;
- Hardware coprocessor for AES;
- Hardware for ECC support;
- True Random Number Generator;
- ISO/IEC 7816 contact interface;
- SPI interface;
- eSPI interface;
- Protection against power analysis;
- Protection against physical attacks;
- Protection against perturbation attacks

The IC dedicated software provides the following services:

- RSA-CRT Decryption operation;
- TDES ECB operation;
- TRNG;
- AES ECB operation;
- ECC operation

The detailed information about the reference of the certified THN31 IC Hardware is shown in the table below:

Content	Information
Hardware Name	THN31 1.0.2 Secure Element
Certified HW Version	Version 1.0
Certification ID	NSCIB-CC-2400073-01
Security Target Reference	ASE EAL5+ for TMS THN31 1.0.2

Table 5: The reference of the certified THN311.0.2 IC Hardware

1.5.2.2 Main OS

The Main OS contains the classic secure element services including memory management, I/O management and cryptographic primitives. The functionalities provided by Main OS to the upper layer operating system are listed below:

1.5.2.2.1 Crypto Library(SW)

AES

- Encryption and decryption
- ECB, CBC, CBC-MAC, and CMAC
 - All modes except ECB are implemented based on ECB

TDES

- Encryption and decryption
- ECB, CBC, CBC-MAC
 - All modes except ECB are implemented based on ECB

ECDSA

- Signature generation and signature verification
- ECC key generation
- ECDH is supported
- Key sizes 224, 256, 320, 384, 512 and 521 bits are supported

SHA/HMAC

- SHA-256

Random number generation

- TRNG (Random Number Generator)

1.5.2.2.2 OSU (OS Update)

The OS Update is a standalone subsystem and is selected by a strict process.

When OS Update is running, eUICC OS and eSE OS are in a suspended state.

OSU ensures:

- the Update Image must be kept confidential;
- the Update Image to be protected from unauthorized modification;
- the system stays in a secure state when the update fails.

1.5.2.3 eUICC OS

The eUICC OS has a separate and independent Java Card Platform[[Table 6](#)] and Global Platform[[Table 7](#)], completely isolated from the eSE OS in terms of both Code and Data spaces.

The eUICC OS also has an eUICC Component[[Section 1.5.2.3.3](#)] which implements the GSMA RSP Architecture Specification and Technical Specification for Consumer Devices.

1.5.2.3.1 Java Card Platform details

The Java Card Specification Version listed as follows:

Name	Version
JCVM and JCRE version	Version 3.1
JC API version	Version 3.1

Table 6:The Java Card Specification Version

1.5.2.3.2 Global Platform details

The Global Platform Specification Version listed as follows:

Name	Version
Global Platform	Version 2.3.1
Amendment A	Version 1.1

Amendment B	Version 1.1.3
Amendment D	Version 1.1.1
Amendment E	Version 1.1
Amendment F	Version 1.2
GP Card API	Version 1.7

Table 7:The Global Platform Specification Version

1.5.2.3.3 eUICC Component details

The eUICC component is built on top of the eUICC OS, providing a monolithic eUICC dedicated product.

It implements the Telecom commands and APIs defined in the following specifications:

- ETSI TS 102 222
- ETSI TS 102 221
- ETSI TS 131 102
- ETSI TS 131 103
- 3GPP2 C.S0065-0
- ETSI TS 143 019
- ETSI TS 102 241
- ETSI TS 131 130

The eUICC Application is connected to a given mobile network, by the means of its currently enabled MNO Profile.

1.5.2.3.3.1 eUICC Application Layer

The goal of the Application layer is to implement the eUICC functionalities described in [\[SGP 21\]](#) and [\[SGP 22\]](#), which rely on the notion of a Profile. A Profile is the combination of a file structure, data and applications to be provisioned onto, or present on, an eUICC. Each Profile, combined with the functionality of the eUICC, behaves basically as a SIM card. An eUICC may contain more than one Profile, and since this eUICC product supports the MEP(Multiple Enabled Profiles) feature, at most 2 profiles can be activated at the same time. Each Profile is controlled by a unique ISD-P.

A Profile can have several forms:

- A Provisioning Profile: A Profile that allows connectivity to a mobile network solely to provide the provisioning of Profiles.
- An Operational Profile: A Profile that allows connectivity to a mobile network.
- A Test Profile: A Profile that can only be used in Device Test Mode and cannot be used to connect to any MNO. The support of this kind of profile is not mandatory for an eUICC implementation.

This document will use the term “Profile” to describe either Provisioning Profiles, Operational Profiles, or Test Profiles.

All Profiles include Network Access Applications and associated Parameters, but these applications rely on the algorithms stored in the platform layer of the eUICC.

In the same manner, the Profile includes policy rules (PPR), but relies on the Platform Layer to have them enforced on the eUICC. The Profile structure, composed of a set of Profile Components, is specified by, and under the full control of, the MNO. The full Profile structure shall be contained in a unique ISD-P. The Profile structure shall contain a Profile Component, called MNO-SD, which performs an identical Role as the ISD for a UICC. The Profile structure shall include:

- The MNO-SD;
- Supplementary Security Domains (SSD) and a CASD;
- Applets;
- Applications, e.g. NFC applications;
- NAAs;
- Other elements of the File System;
- Profile metadata, including Profile Policy Rules (PPR).

More details on the Profile can be found in [\[SGP 21\]](#) and [\[SGP 22\]](#).

In addition to Profile data, the eUICC itself has a Rules Authorization Table (RAT) that is used by the Profile Policy Enabler (PPE) and the Local Profile Assistant (non-TOE element LPAd) to determine whether or not a Profile containing PPRs is authorized and can be installed on the eUICC.

The RAT is initialized at eUICC manufacturing time, or during the initial Device setup provided that there is no installed Operational Profile. In particular, it cannot be affected by the Memory Reset function.

ISD-P

The ISD-P is a secure container (Security Domain) for the hosting of a Profile. The ISD-P is also used for updating the Profile Metadata on behalf of the MNO.

As defined in [\[SGP 22\]](#), the ISD-P shall ensure that:

- a) It hosts a unique Profile;
- b) Only the following Application Layer components shall have access to the profiles:
 - ISD-P
 - ISD-R, which shall only have access to the metadata of the profiles.
- c) A Profile component shall not have any visibility of, or access to, components outside its ISD-P. An ISD-P shall not have any visibility of, or access to, any other ISD-P;
- d) Deletion of a Profile shall remove the containing ISD-P and all Profile components of the Profile.

ISD-R

The ISD-R is responsible for the creation of new ISD-Ps and life-cycle management of all ISD-Ps. An ISD-R shall be created within an eUICC at the time of manufacture.

The ISD-R is used for the Profile download and installation, in collaboration with the Profile Package Interpreter for the decoding/interpretation of the received Profile Package, and with an ISD-P as a target.

As defined in [\[SGP 22\]](#):

- a) There shall be only one ISD-R on an eUICC;
- b) The ISD-R shall be installed and personalized by the EUM during eUICC manufacturing.
- c) The ISD-R shall be associated with itself;
- d) The ISD-R cannot be deleted or disabled.

LPA Services

The LPA Services is the subset of ISD-R functionalities that provide the necessary access to the services and data required by LPA (the non-TOE element LPA_d). These services are:

- Transfer Bound Profile Package from the LPA_d to the ISD-P;
- Provide list of installed Profiles;
- Retrieve EID;
- Provide Local/Remote Profile Management Operations (SGP.22);

LPA Services are mandatory even if the LPA_e is provided in the eUICC. LPA Services code is located in the ISD-R.

MNO-SD

The MNO-SD is the on-card representative of the MNO Platform. It contains the MNO Over-The-Air (OTA) keys and provides a secure OTA channel.

ECASD

The Embedded UICC Controlling Authority Security Domain (ECASD) is responsible for the secure storage of credentials used to enforce trust in the identities of Actors (eUICC, remote Actors such as SM-DS or SM-DP+) and provides security functions used during key establishment and eUICC authentication.

As defined in [\[SGP 22\]](#), the ECASD has the following properties:

- a) There can only be one ECASD on an eUICC;
- b) It is installed and personalized by the EUM during the eUICC manufacturing as described in [\[GP\]](#);
- c) It has eUICC private key(s) for creating signatures;
- d) It has associated certificate(s) for eUICC authentication;
- e) It has the Certificate Issuers (CI) root public key(s) for verifying SM-DP+ and SM-DS certificates;
- f) It has the certificate of the EUM.

1.5.2.3.3.2 eUICC Platform Layer

The Platform capabilities include:

- The Telecom Framework, which includes algorithms used by Network Access Applications (NAA) to access mobile networks. The NAAs are part of the Profiles, but the algorithms, as part of the Telecom Framework, are provisioned onto the eUICC during manufacturing.
- The Profile Package Interpreter, an eUICC Operating System service that translates the Profile Package data as defined in eUICC Profile Package Specification [\[Profile PKG\]](#) into an installed Profile using the specific internal format of the target eUICC.
- The Profile Rules Enforcer, which implements the enforcement of Enterprise Rules, and the Profile Policy Enabler (PPE). The PPE has two functions:
 - Verification that a Profile containing PPRs is authorized by the RAT;
 - Enforcement of the PPRs of a Profile.

1.5.2.4 eUICC Usage

The eUICC will contain several MNO Profiles, each of them being associated with a given International Mobile Subscriber Identity (IMSI).

The primary function of the Profile is to authenticate the validity of a Device when accessing the network. The Profile is MNO's property, and stores MNO specific information.

An eUICC with an enabled operational Profile provides the same functionality as a SIM or USIM card.

1.5.3 Life Cycle

The TOE life cycle is part of the product life cycle, considering both the smart card life cycle defined in [\[PP JCS\]](#) and the life cycle presented by [\[PP eUICC\]](#). (see [Figure 4](#)).

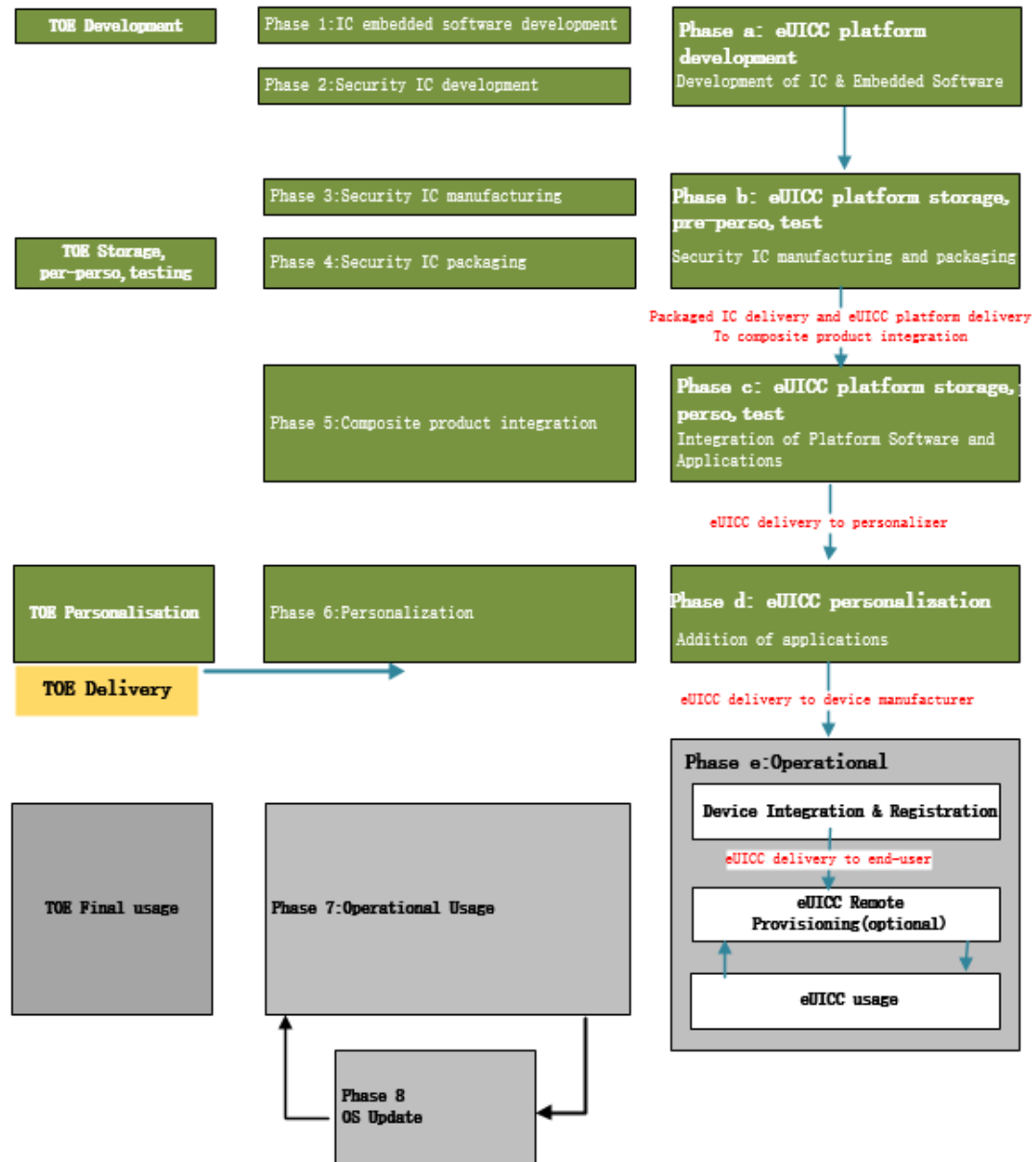


Figure 4: TOE Life Cycle within Product Life Cycle

1.5.3.1 Product life-cycle

Phase	Name	Description
1	Security IC Embedded Software Development	Phases 1 and 2 compose the product development: • Embedded Software development (COS/RE/Applications/other Platform Components) • IC Dedicated Software • IC development
2	Security IC Development	
3	Security IC manufacturing	The Security IC Manufacturer personalizes the smart card and conducts tests on behalf of the developer. The Security IC Manufacturing environment protects the integrity and confidentiality of the assets and of any related material, for instance test suites. The evaluation of this TOE includes the whole Security IC Manufacturing environment, in particular those locations where the assets is accessible for installation or testing.
4	Security IC Packaging	The IC Packaging Manufacturer is responsible for the IC packaging and testing.
5	Composite product integration	The Composite Product Manufacturer is responsible for the product finishing process. The TOE will be loaded into Security IC at this phase, to meet dynamic customization requested by customer.
6	Personalization	The Personalizer is responsible for the smartcard personalization prior final use.
7	Operational Usage	The smartcard operational phase.
8	OS Update	The IC Developer provide the user manual for customer to setting up the OS Update service.

Table 8:Product Life Cycle

After phase 6, the TOE is delivered to end customer as module form factor with its security functions fully enabled. The end customer shall follow the processes and procedures as defined in [\[AGD PRE\]](#) for secure acceptance and installation to conduct activities that are covered in phase 8.

The TOE final usage environment is that of the product where the TOE is embedded in. It covers a wide spectrum of situations that cannot be covered by evaluations. The TOE and the product provide the full set of security functionalities to avoid abuse of the product by untrusted entities.

1.5.3.2 eUICC specific life-cycle

The eUICC life-cycle is composed of the following stages as described by GSMA - Embedded UICC for Consumer Devices Protection Profile [\[PP eUICC\]](#):

Phase	Description
a	eUICC Platform Development: ● Corresponds to the first two stages of the IC development
b	Storage, pre-perso and test: ● Covers the stages related to manufacturing and packaging of the IC
c	eUICC Platform storage, pre-perso, test: ● covers the stage of the embedding of software products onto the eUICC;
d	eUICC Platform personalization: ● The TOE will be loaded into Security IC at this phase, to meet dynamic customization requested by customer. ● Covers the ECASD/ISD-R keys and optionally the addition of provisioning Profiles and Operational Profiles onto the eUICC; TOE Delivery: At this phase, the delivery of the TOE to the customer of the eUICC manufacturer happens at the latest.
e	Operational usage of the TOE, covers the following steps: ● eUICC integration onto the Device is performed by the Device Manufacturer. The Device Manufacturer and/or the eUICC Manufacturer also register the eUICC in a given SM-DS; ● The eUICC is then used to provide connectivity to the Device end-user. ● The eUICC may be provisioned again, at post-issuance, using the remote provisioning infrastructure.

Table 9:eUICC lifecycle stages and Delivery Options

1.5.4 TOE Delivery

The following table lists the items and methods for the TOE delivery:

Type	Name	Version	Delivery method
Composite Product	TMCOS 4.0 0.3 with eUICC extension on THN31 1.0.2	V4.00.3	Courier deliver
Document	[AGD_OPE]	[AGD_OPE]	Encrypted e-mail
	[AGD_PRE]	[AGD_PRE]	Encrypted e-mail

Table 10:The items and methods for the TOE delivery

2 Conformance Claims

2.1 CC Conformance Claim

Common Criteria Version:

This Security Target conforms to Common Criteria 2022 Release 1 [\[CC1\]](#) [\[CC2\]](#) [\[CC3\]](#)[\[CC5\]](#):

- Common Criteria for Information Technology Security Evaluation, Part 1 [\[CC1\]](#): Introduction and general model, CC:2022 Revision 1. November 2022.
- Common Criteria for Information Technology Security Evaluation, Part 2 [\[CC2\]](#): Security functional components, CC:2022 Revision 1. November 2022.
- Common Criteria for Information Technology Security Evaluation, Part 3 [\[CC3\]](#): Security assurance components, CC:2022 Revision 1. November 2022.
- Common Criteria for Information Technology Security Evaluation, Part 5 [\[CC5\]](#): Pre-defined packages of security requirements, CC:2022 Revision 1. November 2022.

The methodology used in the evaluation is [\[CEM\]](#):

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology, CC:2022 Revision 1. November 2022.

Errata for Common Criteria 2022 Release 1, parts 1 to 5, and CEM:2022 (Release 1) providing appropriate solutions as proposed corrections or interpretations, respectively.

- Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1) , CCMB-2024-07-002 Version 1.1, July 2024.

This Security Target is CC Part 2 [\[CC2\]](#) conformant and CC Part 3 [\[CC3\]](#) conformant of Common Criteria 2022 release 1.

CC Package Claim

This Security Target conforms to the assurance package **EAL4 augmented** with AVA_VAN.5 “Advanced methodical vulnerability analysis” and ALC_DVS.2 “Sufficiency of security measures”.

2.2 PP Claim

This Security Target claims conformance to the following Protection Profile.

2.2.1 GSMA SGP.25- Embedded UICC for Consumer Devices

Protection Profile

The Security Target claims demonstrable conformance to the eUICC for Consumer Devices Protection Profile [\[PP_eUICC\]](#).

2.3 Conformance Claim Rationale

2.3.1 Rationale for added elements

While the ST doesn't make any official claims of conformance to the Java Card protection profile, the ST does follow the protection profile since the TOE provides functionality coming from [\[PP_JCS\]](#).

The functionality from [\[PP_JCS\]](#) are compatible with those coming from [\[PP_eUICC\]](#) since the TOE type defined in [\[PP_eUICC\]](#) allow for an underlying certified Java Card platform based on [\[PP_JCS\]](#).

The ST also contains additional functionality coming from the OS Update feature for which the SPD and security objectives come from [\[PP_eUICC\]](#) module for OS Update (there is no configuration for this module and hence no conformance claims are made for this module, the reference to the module is only mentioned to indicate the source of the SPD and objectives). [\[PP_eUICC\]](#) allows replacement of some security objectives with those coming from [\[PP_JCS\]](#). Since the ST contains security objectives from [\[PP_JCS\]](#), the relevant security objectives from [\[PP_eUICC\]](#) have been removed and that is further discussed in [Section 2.3.2](#).

Based on the above declarations, an overview of the additional elements in this ST is provided below:

2.3.1.1 Additional SPD Statement

2.3.1.1.1 Threats

Section 4.1.2 presents the threats as presented in the [\[PP_JCS\]](#) as well as the additional threats. The additional threats are:

- T.CONFID-UPDATE-IMAGE.LOAD
- T.INTEG-UPDATE-IMAGE.LOAD
- T.UNAUTH-UPDATE-IMAGE.LOAD
- T.INTERRUPT_OSU

These threats are included for the OS Update which is additional functionality defined in [\[PP_eUICC\]](#). T.EXE-CODE-REMOTE is not included, since the TOE does not support Java Card RMI.

2.3.1.1.2 OSPs

Section 4.1.3 presents all the OSP defined in [\[PP_JCS\]](#), no item has been added, removed or modified.

2.3.1.1.3 Assumptions

[Section 4.1.4](#) presents most assumptions defined in [\[PP_JCS\]](#) as well as the additional ones.

The Card Manager is included in the TOE, it is ensuring that the deletion of applets through the Card Manager is secure, therefore A.DELETION is no longer relevant.

As the SCP is included in the TOE, OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are changed into the following Objectives on the TOE: O.SCP.RECOVERY, O.SCP.SUPPORT, and O.SCP.IC.

2.3.1.2 Additional Security Objectives Statement

[Section 5.1.1](#) presents most Security Objectives defined in [\[PP_JCS\]](#) as well as the additional ones.

The additional security objectives are:

- O.CONFID-UPDATE-IMAGE.LOAD
- O.AUTH-LOAD-UPDATE-IMAGE
- O.SECURE_LOAD_ACODE
- O.SECURE_AC_ACTIVATION
- O.TOE_IDENTIFICATION

These security objectives are included for the OS Update which is additional functionality defined in [\[PP eUICC\]](#).

[Section 5.2.1](#) presents most security Objectives for the Operational Environment defined in [\[PP_JCS\]](#) as well as the additional ones. The additional security Objectives for the Operational Environment are:

- OE.CONFID_UPDATE_IMAGE.CREATE is included for the OS Update which is additional functionality defined in [\[PP eUICC\]](#).

As the security Objectives for the Operational Environment is included in the TOE, OE.CARD-MANAGEMENT, OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are changed into the following Objectives on the TOE: O.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT, and O.SCP.IC.

2.3.1.3 Additional SFR Statement

The rationale for additional SFRs is provided in [Section 7.1](#).

2.3.2 Rationale for eUICC component

The TOE of this Security Target does not support the SGP.32 functionality.

2.3.2.1 SPD Statement for eUICC Component

The Security Problem Definition for the eUICC component of the TOE is compliant with the Security

Problem Definition described in the [PP_eUICC].

2.3.2.2 Security Objectives Statement for eUICC Component

This Security Target is compliant with the [PP_eUICC] and some objectives have been excluded because they overlap with the additional objectives defined in this ST that come from [PP_JCS], these security objectives are:

- OE.IC.SUPPORT
 - OE.IC.RECOVERY
 - OE.RE.PRE-PPI
 - OE.RE.SECURE-COMM
 - OE.RE.API
 - OE.RE.DATA-CONFIDENTIALITY
 - OE.RE.DATA-INTEGRITY
 - OE.RE.CODE-EXE.
-
- OE.IC.SUPPORT is included within the O.SCP.SUPPORT defined in [PP_JCS].
 - OE.IC.RECOVERY is included within the O.SCP.RECOVERY from defined in [PP_JCS].
 - OE.RE.PRE-PPI is a request on the Java Card System and is fully met by the security objectives of [PP_JCS] related to the following threats: T.DELETION, T.INSTALL. It is covered by the O.DELETION, O.INSTALL, O.CARD-MANAGEMENT and O.LOAD.
 - OE.RE.SECURE-COMM is a request on the Java Card System and is fully met by the security objectives of [PP_JCS] related to the following threats: T.CONFID-APPLI-DATA and T.INTEG-APPLI-DATA. It is covered by O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.RNG, O.KEY-MNGT and O.PIN-MNGT.
 - OE.RE.API is a request on the Java Card System and is fully met by the security objectives of [PP_JCS] related to the following threats: T.CONFID-JCS-CODE, T.INTEG-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-JCS-DATA. It covered by OE.VERIFICATION and O.NATIVE.
 - OE.RE.DATA-CONFIDENTIALITY is a request on the Java Card System and is fully met by the security objectives of [PP_JCS] related to the following threats: T.CONFID-APPLI-DATA. it is covered by OE.VERIFICATION, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.CIPHER, O.RNG, O.OPERATE_JCS, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.CARD-MANAGEMENT, O.SCP.RECOVERY and O.SCP.SUPPORT.
 - OE.RE.DATA-INTEGRITY is a request on the Java Card System and is fully met by the security objectives of [PP_JCS] related to the following threats: T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD. It covered by O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE, O.LOAD, O.SCP.RECOVERY, O.SCP.SUPPORT, O.SID, O.OPERATE_JCS, O.FIREWALL, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, O.ALARM, O.TRANSACTION, O.CIPHER, O.RNG, O.PIN-MNGT, O.KEY-MNGT and O.REALLOCATION.

- OE.RE.CODE-EXE is a request on the Java Card System and is fully met by the security objectives of [PP_JCS] related to the following threats: T.EXE-CODE1, T.EXE-CODE.2 and T.NATIVE. It covered by O.FIREWALL, OE.VERIFICATION, O.NATIVE, OE.CAP_FILE.

2.3.2.3 SFRs Statement for eUICC Component

[Section 7.2](#) presents most SFRs as defined in the [[PP_eUICC](#)].

3 Security Aspects

This chapter describes the main security issues of the Java Card System and its environment addressed in this Security Target, called “security aspects”, in a CC-independent way. In addition to this, they also give a semi-formal framework to express the CC security environment and objectives of the TOE. They can be instantiated as assumptions, threats, objectives (for the TOE and the environment) or organizational security policies. For instance, we will define hereafter the following aspect:

#.OPERATE (1) The TOE must ensure continued correct operation of its security functions. (2) The TOE must also return to a well-defined valid state before a service request in case of failure during its operation.

3.1 Confidentiality

- | | |
|---------------------|---|
| #.CONFID-APPLI-DATA | Application data must be protected against unauthorized disclosure. This concerns logical attacks at runtime in order to gain read access to other application’s data. |
| #.CONFID-JCS-CODE | Java Card System code must be protected against unauthorized disclosure. Knowledge of the Java Card System code may allow bypassing the TSF. This concerns logical attacks at runtime in order to gain a read access to executable code, typically by executing an application that tries to read the memory area where a piece of Java Card System code is stored. |
| #.CONFID-JCS-DATA | Java Card System data must be protected against unauthorized disclosure. This concerns logical attacks at runtime in order to gain a read access to Java Card System data. Java Card System data includes the data managed by the Java Card RE, the Java Card VM and the internal data of Java Card platform API classes as well. |

3.2 Integrity

- | | |
|--------------------|---|
| #.INTEG-APPLI-CODE | Application code must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to the memory zone where executable code is stored. In post-issuance application loading, this threat also concerns the modification of application code in transit to the card. |
| #.INTEG-APPLI-DATA | Application data must be protected against unauthorized modification. |

This concerns logical attacks at runtime in order to gain unauthorized write access to application data. In post-issuance application loading, this threat also concerns the modification of application data contained in a CAP file in transit to the card. For instance, a CAP file contains the values to be used for initializing the static fields of the CAP file.

#.INTEG-JCS-CODE Java Card System code must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to executable code.

#.INTEG-JCS-DATA Java Card System data must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to Java Card System data. Java Card System data includes the data managed by the Java Card RE, the Java Card VM and the internal data of Java Card API classes as well.

3.3 Unauthorized Executions

#.EXE-APPLI-CODE Application (byte)code must be protected against unauthorized execution. This Concerns:

1. invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([\[JAVASPEC\]](#), §6.6);
2. jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code;
3. unauthorized execution of a remote method from the CAD (if the TOE provides JCRMI functionality).

#.EXE-JCS-CODE Java Card System bytecode must be protected against unauthorized execution. Java Card System bytecode includes any code of the Java Card RE or API. These concerns:

1. invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([\[JAVASPEC\]](#), §6.6);
2. jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code. Note that execute access to native code of the Java Card System and applications is the concern of #.NATIVE.

#.FIREWALL The Firewall shall ensure controlled sharing of class instances⁷, and isolation of their data and code between CAP files (that is, controlled

execution contexts) as well as between CAP files and the JCRE context. An applet shall not read, write, compare a piece of data belonging to an applet that is not in the same context, or execute one of the methods of an applet in another context without its authorization.

#.NATIVE

Because the execution of native code is outside of the JCS TSF scope, it must be secured so as to not provide ways to bypass the TSFs of the JCS. Loading of native code, which is as well outside those TSFs, is submitted to the same requirements. Should native software be privileged in this respect, exceptions to the policies must include a rationale for the new security framework they introduce.

3.4 Bytecode Verification

#.VERIFICATION

Bytecode must be verified prior to being executed. Bytecode verification includes:

1. how well-formed CAP file is and the verification of the typing constraints on the bytecode,
2. binary compatibility with installed CAP files and the assurance that the export files used to check the CAP file correspond to those that will be present on the card when loading occurs.

3.5 Card Management

#.CARD-MANAGEMENT

(1) The card manager (CM) shall control the access to card management functions such as the installation, update or deletion of applets. (2) The card manager shall implement the card issuer's policy on the card.

#.INSTALL

(1) The TOE must be able to return to a safe and consistent state when the installation of a CAP file or an applet fails or be canceled (whatever the reasons). (2) Installing an applet must have no effect on the code and data of already installed applets. The installation procedure should not be used to bypass the TSFs. In short, it is an atomic operation, free of harmful effects on the state of the other applets. (3) The procedure of loading and installing a CAP file shall ensure its integrity and authenticity. In case of Extended CAP files,

installation of a CAP shall ensure installation of all the packages in the CAP file.

#.SID

Users and subjects of the TOE must be identified. (2) The identity of sensitive users and subjects associated with administrative and privileged roles must be particularly protected; this concerns the Java Card RE, the applets registered on the card, and especially the default applet and the currently selected applet (and all other active applets in Java Card System 2.2.x). A change of identity, especially standing for an administrative role (like an applet impersonating the Java Card RE), is a severe violation of the Security Functional Requirements (SFR). Selection controls the access to any data exchange between the TOE and the CAD and therefore, must be protected as well. The loading of a CAP file or any exchange of data through the APDU buffer (which can be accessed by any applet) can lead to disclosure of keys, application code or data, and so on.

#.OBJ-DELETION

(1) Deallocation of objects should not introduce security holes in the form of references pointing to memory zones that are no longer in use, or have been reused for other purposes. Deletion of collection of objects should not be maliciously used to circumvent the TSFs. (2) Erasure, if deemed successful, shall ensure that the deleted class instance is no longer accessible.

#.DELETION

(1) Deletion of installed applets (or CAP files) should not introduce security holes in the form of broken references to garbage collected code or data, nor should they alter integrity or confidentiality of remaining applets. The deletion procedure should not be maliciously used to bypass the TSFs. (2) Erasure, if deemed successful, shall ensure that any data owned by the deleted applet is no longer accessible (shared objects shall either prevent deletion or be made inaccessible). A deleted applet cannot be selected or receive APDU commands. CAP file deletion shall make the code of the CAP file is no longer available for execution. In case of Extended CAP files, deletion of a CAP shall ensure that code and data for all the packages in the CAP file is no longer available for execution. (3) Power failure or other failures during the process shall be taken into account in the implementation so as to preserve the SFRs. This does not mandate, however, the process to be atomic. For instance, an interrupted deletion may result in the loss of user data, as long as it does not violate the SFRs.

The deletion procedure and its characteristics (whether deletion is either physical or logical, what happens if the deleted application was the default applet, the order to be observed on the deletion steps) are implementation-dependent. The only commitment is that deletion shall not jeopardize the TOE (or its assets) in case of failure (such as power shortage).

Deletion of a single applet instance and deletion of a whole CAP file are functionally different operations and may obey different security rules. For instance, specific CAP files or packages can be declared to be undeletable (for instance, the Java Card API packages), or the dependency between installed CAP files may forbid the deletion (like a CAP file using super classes or super interfaces declared in another CAP file).

3.6 Services

#.ALARM	The TOE shall provide appropriate feedback upon detection of a potential violation. This particularly concerns the type errors detected by the bytecode verifier, the security exceptions thrown by the Java Card VM, or any other security-related event occurring during the execution of a TSF.
#.OPERATE	(1) The TOE must ensure continued correct operation of its security functions. (2) In case of failure during its operation, the TOE must also return to a well-defined valid state before the next service request.
#.RESOURCES	The TOE controls the availability of resources for the applications in order to prevent unauthorized denial of service or malfunction of the TSFs. This concerns both execution (dynamic memory allocation) and installation (static memory allocation) of applications and CAP files.
#.CIPHER	The TOE shall provide a means to the applications for ciphering sensitive data, for instance, through a programming interface to low-level, highly secure cryptographic services. In particular, those services must support cryptographic algorithms consistent with cryptographic usage policies and standards.
#.KEY-MNGT	The TOE shall provide a means to securely manage cryptographic keys. This includes: (1) Keys shall be generated in accordance with specified cryptographic key generation algorithms and specified cryptographic key sizes, (2) Keys must be distributed in accordance with specified cryptographic key distribution methods, (3) Keys must be initialized before being used, (4) Keys shall be destroyed in accordance with specified cryptographic key destruction methods.
#.PIN-MNGT	The TOE shall provide a means to securely manage PIN objects. This includes: (1) Atomic update of PIN value and try counter, (2) No rollback on the PIN- checking function, (3) Keeping the PIN value (once initialized) secret (for instance, no clear-PIN-reading function), (4) Protection of PIN's security attributes (state, try counter, try limit, ...) in integrity.
#.SCP	The smart card platform must be secure with respect to the SFRs. Then: (1) After a power loss, RF signal loss or sudden card removal prior to completion of some communication protocol, the SCP will allow the TOE on the next power up to either complete the interrupted operation or revert

to a secure state. (2) It does not allow the SFRs to be bypassed or altered and does not allow access to other low-level functions than those made available by packages of Java Card API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System. (3) It provides secure low-level cryptographic processing to the Java Card System. (4) It supports the needs for any update to a single persistent object or class field to be atomic, and possibly a low-level transaction mechanism. (5) It allows the Java Card System to store data in “persistent technology memory” or in volatile memory, depending on its needs (for instance, transient objects must not be stored in non-volatile memory). The memory model is structured and allows for low-level control accesses (segmentation fault detection). (6) It safely transmits low-level exceptions to the TOE (arithmetic exceptions, checksum errors), when applicable. Finally, it is required that (7) the IC is designed in accordance with a well-defined set of policies and standards (for instance, those specified in [\[PP0084b\]](#)), and will be tamper resistant to actually prevent an attacker from extracting or altering security data (like cryptographic keys) by using commonly employed techniques (physical probing and sophisticated analysis of the chip). This especially matters to the management (storage and operation) of confidential application data such as cryptographic keys.

#.TRANSACTION

The TOE must provide a means to execute a set of operations atomically. This mechanism must not jeopardize the execution of the user applications. The transaction status at the beginning of an applet session must be closed (no pending updates).

3.7 OS Update

#.CONFID-UPDATE-IMAGE

The update image must be kept confidential. This concerns the non-disclosure of the update image in the transit to the eUICC.

#.INTEG-UPDATE-IMAGE

The update image must be protected against unauthorized modification. This concerns the modification of the image in transit to the eUICC.

4 Security Problem Definition

4.1 SPD for Java Card System

4.1.1 Assets

Assets are security-relevant elements to be directly protected by the TOE. Confidentiality of assets is always intended with respect to un-trusted people or software, as various parties are involved during the first stages of the smart card product life-cycle; details are given in threats hereafter.

Assets may overlap, in the sense that distinct assets may refer (partially or wholly) to the same piece of information or data. For example, a piece of software may be either a piece of source code (one asset) or a piece of compiled code (another asset), and may exist in various formats at different stages of its development (digital supports, printed paper). This separation is motivated by the fact that a threat may concern one form at one stage, but be meaningless for another form at another stage.

The assets to be protected by the TOE are listed below. They are grouped according to whether it is data created by and for the user (User data) or data created by and for the TOE (TSF data). For each asset it is specified the kind of dangers that weigh on it.

4.1.1.1 User Data

Asset	Description
D.APP_CODE	The code of the applets and libraries loaded on the card. To be protected from unauthorized modification.
D.APP_C_DATA	Confidentiality – sensitive data of the applications, like the data contained in an object, an array view, a static field, a local variable of the currently executed method, or a position of the operand stack. To be protected from unauthorized disclosure.
D.APP_I_DATA	Integrity sensitive data of the applications, like the data contained in an object, an array view and the PIN security attributes (PIN Try limit, PIN Try counter and State). To be protected from unauthorized modification.
D.APP_KEYs	Cryptographic keys owned by the applets. To be protected from unauthorized disclosure and modification.
D.PIN	Any end-user's PIN. To be protected from unauthorized disclosure and modification.
D.CARD_MNGT_DATA	The data of the card management environment, like for instance, the identifiers, the privileges, life cycle states, the memory resource quotas

	of applets and security domains. To be protected from unauthorized modification.
--	--

Table 11:User Data Assets

4.1.1.2 TSF Data

Asset	Description
D.API_DATA	Private data of the API, like the contents of its private fields. To be protected from unauthorized disclosure and modification.
D.CRYPTO	Cryptographic data used in runtime cryptographic computations, like a seed used to generate a key. To be protected from unauthorized disclosure and modification.
D.JCS_CODE	The code of the Java Card System. To be protected from unauthorized disclosure and modification.
D.JCS_DATA	The internal runtime data areas necessary for the execution of the Java Card VM, such as, for instance, the frame stack, the program counter, the class of an object, the length allocated for an array, any pointer used to chain data-structures. To be protected from unauthorized disclosure or modification.
D.SEC_DATA	The runtime security data of the Java Card RE, like, for instance, the AIDs used to identify the installed applets, the currently selected applet, the current context of execution and the owner of each object. To be protected from unauthorized disclosure and modification.
D.TOE_IDENTIFIER	Identification Data to identify the TOE.
D.UPDATE_IMAGE	Can be an update for the OS, as a patch or a complete OS replacement, or separate bootloader It is sent to the TOE. It possibly includes executable code, configuration data and/or image type information. It has to be protected from unauthorized disclosure and modification. Is also referred to as Additional Code
D.OS-UPDATE_KEY(S)	Key(s) used for OS Update. To be protected from unauthorized disclosure and modification.

Table 12:TSF Data Assets

4.1.2 Threats

This section introduces the threats to the assets against which specific protection within the TOE or its environment is required. The definition of the grouping is taken from Section 5.2 of [\[PP JCS\]](#).

4.1.2.1 Confidentiality

T.CONFID-APPLI-DATA

The attacker executes an application to disclose data belonging to another application. See #.CONFID-APPLI-DATA for details.

Application Note:

Refining the ADV_ARC “non-bypassability” requirements to explicit the coverage of side channel attacks by the security architecture of the ST TOE.

Directly threatened asset(s): D.APP_C_DATA, D.PIN and D.APP_KEYS.

T.CONFID-JCS-CODE

The attacker executes an application to disclose the Java Card System code. See #.CONFID-JCS-CODE for details.

Directly threatened asset(s): D.JCS_CODE.

T.CONFID-JCS-DATA

The attacker executes an application to disclose data belonging to the Java Card System. See #.CONFID-JCS-DATA for details.

Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.

4.1.2.2 Integrity

T.INTEG-APPLI-CODE

The attacker executes an application to alter (part of) its own code or another application's code. See #.INTEG-APPLI-CODE for details.

Directly threatened asset(s): D.APP_CODE.

T.INTEG-APPLI-CODE.LOAD

The attacker modifies (part of) its own or another application code when an application CAP file is transmitted to the card for installation. See #.INTEG-APPLI-CODE for details.

Directly threatened asset(s): D.APP_CODE.

T.INTEG-APPLI-DATA

The attacker executes an application to alter (part of) another application's data. See #.INTEG-APPLI-DATA for details.

Directly threatened asset(s): D.APP_I_DATA, D.PIN, and D.APP_KEYS.

T.INTEG-APPLI-DATA.LOAD

The attacker modifies (part of) the initialization data contained in an application CAP file when the CAP file is transmitted to the card for installation. See #.INTEG-APPLI-DATA for details.

Directly threatened asset(s): D.APP_I_DATA and D_APP_KEY.

T.INTEG-JCS-CODE

The attacker executes an application to alter (part of) the Java Card System code. See #.INTEG-JCS-CODE for details.

Directly threatened asset(s): D.JCS_CODE.

T.INTEG-JCS-DATA

The attacker executes an application to alter (part of) Java Card System or API data. See #.INTEG-JCS-DATA for details.

Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.

4.1.2.3 Identity Usurpation

T.SID.1

An applet impersonates another application, or even the Java Card RE, in order to gain illegal access to some resources of the card or with respect to the end user or the terminal. See #.SID for details.

Directly threatened asset(s): D.SEC_DATA (other assets may be jeopardized should this attack succeed, for instance, if the identity of the JCRE is usurped), D.PIN and D.APP_KEYS.

T.SID.2

The attacker modifies the TOE's attribution of a privileged role (e.g. default applet and currently selected applet), which allows illegal impersonation of this role. See #.SID for further details.

Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on whose identity was forged).

4.1.2.4 Unauthorized Execution

T.EXE-CODE.1

An applet performs an unauthorized execution of a method. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE for details.

Directly threatened asset(s): D.APP_CODE.

T.EXE-CODE.2

An applet performs an execution of a method fragment or arbitrary data. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE for details.

Directly threatened asset(s): D.APP_CODE.

T.NATIVE

An applet executes a native method to bypass a TOE Security Function such as the firewall. See #.NATIVE for details.

Directly threatened asset(s): D.JCS_DATA.

4.1.2.5 Denial of Service

T.RESOURCES

An attacker prevents correct operation of the Java Card System through consumption of some resources of the card: RAM or NVRAM. See #.RESOURCES for details.

Directly threatened asset(s): D.JCS_DATA.

4.1.2.6 Card Management

T.DELETION

The attacker deletes an applet or a CAP file already in use on the card, or uses the deletion functions to pave the way for further attacks (putting the TOE in an insecure state). See #.DELETION for details).

Directly threatened asset(s): D.SEC_DATA and D.APP_CODE.

T.INSTALL

The attacker fraudulently installs post-issuance of an applet on the card. This concerns either the installation of an unverified applet or an attempt to induce a malfunction in the TOE through the installation process. See #.INSTALL for details.

Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on the virulence of the installed application).

4.1.2.7 Services

T.OBJ-DELETION

The attacker keeps a reference to a garbage collected object in order to force the TOE to execute an unavailable method, to make it to crash, or to gain access to a memory containing data that is now being used by another application. See #.OBJ-DELETION for further details.

Directly threatened asset(s): D.APP_C_DATA, D.APP_I_DATA and D.APP_KEYS

4.1.2.8 Miscellaneous

T.PHYSICAL

The attacker discloses or modifies the design of the TOE, its sensitive data or application code by physical (opposed to logical) tampering means. This threat includes IC failure analysis, electrical probing, unexpected tearing, and DPA. That also includes the modification of the runtime execution of Java Card System or SCP software through alteration of the intended execution order of (set of) instructions through physical tampering techniques.

This threatens all the identified assets.

This threat refers to the point (7) of the security aspect #.SCP, and all aspects related to confidentiality and integrity of code and data.

4.1.2.9 OS Update

T.CONFID-UPDATE-IMAGE.LOAD

The attacker discloses (part of) the image used to update the TOE in the field while the image is transmitted to the card for installation. See #.CONFID-UPDATE-IMAGE for details. Directly threatened asset(s): D.UPDATE_IMAGE, D.JCS_CODE, and D.JCS_DATA.

T.UNAUTH-UPDATE-IMAGE.LOAD

The attacker tries to upload an unauthorized Update Image. Directly threatened asset(s): D.JCS_CODE, D.JCS_DATA, D.UPDATE_IMAGE, and D.OS-UPDATE_KEY(S).

T.INTEG-UPDATE-IMAGE.LOAD

The attacker modifies (part of) the image used to update the TOE in the field while the image is transmitted to the card for installation. See #.INTEG-UPDATE-IMAGE for details. Directly threatened asset(s): D.UPDATE_IMAGE, D.JCS_CODE, and D.JCS_DATA.

T.INTERRUPT-OSU

The attacker tries to interrupt the OS Update procedure (Load Phase through activation of additional code) leaving the TOE in a partially functional state. Directly threatened asset(s): D.JCS_CODE, D.JCS_DATA, D.UPDATE_IMAGE, D.TOE_IDENTIFIER.

4.1.3 Organizational Security Policies

This section describes the organizational security policies to be enforced with respect to the TOE environment. The definition of the grouping is taken from Section 5.3 of [\[PP JCS\]](#).

OSP.VERIFICATION

This policy shall ensure the consistency between the export files used in the verification and those used for installing the verified file. The policy must also ensure that no modification of the file is performed in between its verification and the signing by the verification authority. See #.VERIFICATION for details.

If the application development guidance provided by the platform developer contains recommendations related to the isolation property of the platform, this policy shall also ensure that the verification authority checks that these recommendations are applied in the application code.

4.1.4 Assumptions

This section introduces the assumptions made on the environment of the Java Card System.

A.CAP_FILE

CAP Files loaded post-issuance do not contain native methods. The Java Card specification explicitly "does not include support for native methods" ([\[JCV M22\]](#), §3.3) outside the API.

A.VERIFICATION

All the bytecodes are verified at least once, before the loading, before the installation or before the execution, depending on the card capabilities, in order to ensure that each bytecode is valid at execution time.

4.2 SPD for eUICC

4.2.1 Assets for eUICC

The following assets of eUICC which presented in section 3.1 of [\[PP eUICC\]](#) are protected by the <TMCOS 4.0 0.3>.

Assets	Descriptions
D.MNO_KEYS	Keys used by MNO OTA Platform to request management operations from the ISD-P. The keys are loaded during provisioning and stored under the control of the MNO SD.
D.PROFILE_NAA_PARAMS	Parameters used for network authentication, including keys. Such parameters may include for example elliptic curve parameters. Parameters are loaded during provisioning and stored under the control of the ISD-P. They may be transmitted to the Telecom Framework, which contains the authentication algorithms.
D.PROFILE_IDENTITY	The International Mobile Subscriber Identity is the user credential when authenticating on a MNO's network via an Authentication algorithm. The IMSI is a representation of the subscriber's identity and will be used by the MNO as an index for the subscriber in its HLR. Each IMSI is stored under the control of the ISD-P during provisioning.
D.PROFILE_RULES	Data describing the profile policy rules (PPRs) of a profile , and the Enterprise Rules (optional). These rules are loaded during provisioning and stored under the control of the ISD-P. They are managed by the MNO OTA Platform.
D.PROFILE_USER_CODES	This asset consists of: - the optional Activation Code that End User may use to initiate a Profile Download and Installation via the Local User Interface (LUId); - the hash of the optional Confirmation Code (Hashed Confirmation Code) that End User may use to confirm a Profile Download and Installation via the Local User Interface (LUId).
D.PROFILE_CODE	The profile applications include first and second level applications ([ETSI_TS_102221]), in particular: - The MNO-SD and the Security Domains under the control of the MNO-SD (CASD, SSD); - The other applications that may be provisioned within the MNO-SD (network access applications, and so on). This asset also includes, by convention, the file system of the Profile. All these applications are under the control of the MNO SD.
D.TSF_CODE	The TSF Code distinguishes between - the ISD-R, ISD-Ps and ECASD; - the Platform Code.
D.PLATFORM_DATA	- The data of the platform environment, like for instance, - the identifiers and privileges including SM-DS OID, MNO OID, SM-DP+ OID; - the eUICC life-cycle state of the ISD-P security domain.

D.DEVICE_INFO	This asset includes the security-sensitive elements of Device Information data, such as the device type allocation code (TAC) or the device capabilities (ex. Support for updating of certificate revocation lists (CRLs)), that is provided to the eUICC by the LPAd/IPAd.
D.PLATFORM_RAT	Data describing the Rules Authorisation Table (RAT) of the eUICC. These rules are initialised at eUICC manufacturing time or during the initial device setup provided that there is no installed operational profile. The OEM or EUM is responsible for setting the content of the RAT. RAT is stored in the eUICC.
D.SK.EUICC.ECDSA	The eUICC private key(s), stored in ECASD, used by the eUICC to prove its identity, generate shared secrets with remote actors, and generate signatures.
D.CERT.EUICC.ECDSA	Certificate(s) issued by the EUM for a specific, individual, eUICC. Certificates contain public keys PK.EUICC.ECDSA and are stored in ECASD. This certificate(s) can be verified using the EUM Certificate.
D.PK.CI.ECDSA	The eSIM CA public key (D.PK.CI.ECDSA) used to verify the certification chain of eUICC and remote actors. It is stored in ECASD.
D.EID	The EID (eUICC-ID) uniquely identifies the eUICC. This identifier is set by the eUICC manufacturer and does not change during operational life of the eUICC. It is stored in ECASD. The EID is used as a key by SM-DP+ and SM-DS to identify eUICCs in their databases.
D.SECRETS	This asset includes: - the one-time keys of the eUICC and the SM-DP+: - otSK.EUICC.ECKA, - otPK.EUICC.ECKA, - otSK.EUICC.ECKAeac (optional), - otPK.EUICC.ECKAeac (optional) - otPK.DP.ECKA; - the shared secret (ShS) used to protect the Profile download; and - session keys (S-ENC and S-MAC) and the initial MAC chaining value.
D.CERT.EUM.ECDSA	The Certificate(s) of the EUM (CERT.EUM.ECDSA).
D.CRLs	The optional certificate revocation lists (extract) stored in the eUICC.

Table 13: Assets for eUICC

4.2.2 Users / Subjects for eUICC

This section describes the User / Subjects to be enforced with respect to the TOE environment. The definition of the grouping is taken from Section 3.2 of [[PP_eUICC](#)].

This section distinguishes between:

- Users, which are entities external to the TOE that may access its services or interfaces;
- Subjects, which are specific parts of the TOE performing specific operations. The subjects are subparts of the asset D.TSF_CODE.

Users	Descriptions
U.SM-DP+	Role that prepares the Profiles and manages the secure download and installation of these Profiles onto the eUICC.
U.SM-DS	Role that securely performs functions of discovery.
U.MNO-OTA	An MNO platform for remote management of UICCs and the content of Enabled MNO Profiles on eUICCs.
U.MNO-SD	A MNO-SD is a Security Domain part of the Profile, owned by the MNO, providing the Secured Channel to the MNO's OTA Platform (U.MNO-OTA). It is used to manage the content of a Profile once the Profile is enabled. An eUICC can contain more than one MNO-SD.
U.End-User	The person using the Device.

Table 14:Users for eUICC

Subjects	Descriptions
S.ISD-R	The ISD-R is responsible for the creation of new ISD-Ps and life-cycle management of all ISD-Ps. The ISD-R includes LPA/IPA Services that provides the necessary access to the services and data required by LPA/IPA functions. LPA/IPA Services are mandatory, regardless of the fact whether it is LP Ae/IP Ae or LP Ad/IP Ad which is active. The ISD-R acts also as an object. In this case, it is named SO.ISD-R.
S.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 a Profile. The ISD-P acts also as an object. In this case, it is named SO.ISD-P.
S.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.

S.PPI	Profile Package Interpreter, an eUICC Operating System service that translates the Profile Package data as defined in eUICC Profile Package Specification [Profile_PKG] into an installed Profile using the specific internal format of the target eUICC.
S.PRE	Profile Rules Enforcer (PRE), which enforces the reference Enterprise Rules and contains the Profile Policy Enabler (PPE). The PPE has two functions: • Verification that a Profile containing PPRs is authorised by the RAT; • Enforcement of the PPRs of a Profile.
S.TELECOM	The Telecom Framework is an Operating System service that provides standardised network authentication algorithms to the NAAs hosted in the ISD-Ps.

Table 15:Subjects for eUICC

4.2.3 Threats for eUICC

This section introduces the threats to the assets against which specific protection within the TOE or its environment is required. The definition of the grouping is taken from Section 3.3 of [[PP_eUICC](#)].

Threats	Descriptions
T.UNAUTHORIZED-PROFILE-MNG	A malicious on-card application: ○ modifies or discloses profile data belonging to ISD-P or MNO-SD; ○ executes or modifies operations from profile applications (ISD-P, MNO-SD and applications controlled by MNO-SD); ○ modifies or discloses the ISD-P or MNO-SD application Such threat typically includes for example: ○ direct access to fields or methods of the Java objects; ○ exploitation of the APDU buffer and global byte array. Directly threatens the assets: D.MNO_KEYS, D.TSF_CODE (ISD-P), D.PROFILE_*;

T.UNAUTHORIZED-PLATFORM-MNG	A malicious on-card application: ○ modifies or discloses data of the ISD-R or PRE; ○ executes or modifies operations from ISD-R or PRE; ○ modifies the rules authorisation table (RAT) stored in the PRE Such threat typically includes for example: ○ direct access to fields or methods of the Java objects; ○ exploitation of the APDU buffer and global byte array. Directly threatened assets are D.TSF_CODE, D.PLATFORM_DATA and D.PLATFORM_RAT. By altering the behaviour of ISD-R or PRE, the attacker indirectly threatens the provisioning status of the eUICC, thus also threatens the same assets as T.UNAUTHORIZED-PROFILE-MNG.
T.PROFILE-MNG-INTERCEPTION	An off-card or on-card actor alters or eavesdrops the transmission between eUICC and SM-DP+ (ES8+), or eUICC and MNO OTA Platform (ES6), Device and eUICC in case of RPM (UpdateMetadataRequest) and eUICC in case of eUICC Package in order to: ○ disclose, replace or modify the content of a profile during its download to the eUICC; ○ download a profile on the eUICC without authorization; ○ replace or modify the content of a command from SM-DP+ or MNO OTA platform; ○ replace or modify the content of Profile Metadata (ex. The Profile Policy Rules (PPR), Enterprise Rules, ...) data when updated by the MNO OTA platform or by RPM request; NOTE: the attacker may be an on-card application intercepting transmissions to the security domains, or an off-card actor intercepting OTA transmissions or interface between the eUICC and the Device. Directly threatens the assets: D.MNO_KEYS, D.TSF_CODE (ISD-P and ISD-R), D.PROFILE_.*.

T.PROFILE-MNG-ELIGIBILITY	An off-card or on-card actor alters or eavesdrops the transmission between eUICC and SM-DP+ (ES8+), or alters the Device Information when provided from the LPA/PA to the eUICC, in order to compromise the eligibility of the eUICC, for example: ○ downgrade the security of the profile sent to the eUICC by claiming compliance to a previous version of the specification, or lack of cryptographic support; ○ obtain an unauthorized profile by modifying the Device Info or eUICC identifier. NOTE: the attacker may be an on-card application intercepting transmissions to the security domains, or an off-card actor intercepting OTA transmissions or interface between the eUICC and the Device. Directly threatens the assets: D.TSF_CODE, D.DEVICE_INFO, D.EID.
T.UNAUTHORIZED-IDENTITY-MNG	A malicious on-card application: ○ discloses or modifies data belonging to the “Identity management data” or the “TSF Code” asset category: ▪ discloses or modifies D.SK.EUICC.ECDSA, D.SECRETS, ▪ modifies D.CERT.EUICC.ECDSA, D.PK.CI.ECDSA, D.EID, D.CERT.EUM.ECDSA, D.CRLs, ▪ modifies the generation method (part of D.TSF_CODE) for shared secrets, one- time keys or session keys (i.e. methods used to generate D.SECRETS); ○ discloses or modifies functionalities of the ECASD (part of D.TSF_CODE). Such threat typically includes for example: ○ direct access to fields or methods of the Java objects; ○ exploitation of the APDU buffer and global byte array ○ impersonation of an application, of the Runtime Environment, or modification of privileges of an application Directly threatens the assets: D.TSF_CODE, D.SK.EUICC.ECDSA, D.SECRETS, D.CERT.EUICC.ECDSA, D.PK.CI.ECDSA, D.EID, D.CERT.EUM.ECDSA,D.CRLs

T.IDENTITY-INTERCEPTION	An off-card or on-card application may try to intercept credentials, either on-card or off-card, in order to use them on another eUICC or on a simulator ○ modify them / replace them with other credentials. This includes on-card interception of: ○ the shared secrets used in profile download (D.SECRETS) ○ the eUICC-ID (D.EID) This does not include: ○ off-card or on-card interception of SM-DP+ credentials during profile download (taken into account by T.PROFILE-MNG-INTERCEPTION) Directly threatens the assets: D.SECRETS, D.EID.
T.UNAUTHORIZED-eUICC	An off-card actor achieves the installation of a legitimate profile on an unauthorized eUICC, or on any other unauthorized platform (for example a simulator or soft SIM). This involves targeting assets used to provide eUICC identity: the objects controlled by the ECASD (EUICC private key and EID), the ECASD code (D.TSF_CODE (ECASD)), and the generation of SCP03t-related key material (D.SECRETS). Directly threatens the assets: D.TSF_CODE (ECASD), D.SK.EUICC.ECDSA, D.EID, D.SECRETS.

T.LPAd-INTERFACE-EXPLOIT	An off-card actor exploits the interfaces to LPAd/IPAd (interfaces ES10a, ES10b and ES10c (SGP.22)) to: ○ either impersonate the LPAd/IPAd (Man-in-the-middle, masquerade), or ○ exploit a flaw in the interface to modify or disclose sensitive assets, or execute code (extension of T.LOGICAL-ATTACK and T.PHYSICAL-ATTACK targeting specifically the interfaces to LPAd/IPAd). The attacker could thus perform unauthorized profile and platform management, for instance by circumventing the End User confirmation (SGP.22) needed for such actions, execute eUICCMemoryReset. The attacker could also compromise the eligibility check process by compromising the Device Information that is normally passed on from the LPA/IPAd to the eUICC before profile download and installation. The difference to the threats T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED- PLATFORM-MNG, and T.PROFILE-MNG-ELIGIBILITY, is on the interfaces used to perform the attack (ES10a,b,c). Directly threatened asset: D.DEVICE_INFO, D.PLATFORM_DATA.
T.UNAUTHORIZED-MOBILE-ACCESS	An off-card actor or on-card application tries to leverage upon flaws of the network authentication algorithms (e.g., Milenage and Tuak) to gain access to network authentication keys, in order to later authenticate on the mobile network of a MNO in place of a legitimate Profile. Directly threatens the assets: D.PROFILE_NAA_PARAMS.

T.LOGICAL-ATTACK	An on-card application bypasses the Platform security measures by logical means, in order to disclose or modify sensitive data when they are processed by the Platform: ○ IC and OS software ○ Runtime Environment (for example provided by JCS) ○ the Profile Rules Enforcer ○ the Profile Package Interpreter ○ the Telecom Framework (accessing Network Authentication Parameters). An example of such a threat would consist of using buffer overflows to access confidential data manipulated by native libraries. This threat also includes cases of unauthorized code execution by applications. Directly threatens the assets: D.TSF_CODE, D.PROFILE_NAA_PARAMS, D.PROFILE_RULES, D.PLATFORM_DATA, D.PLATFORM_RAT.
T.PHYSICAL-ATTACK	The off-card actor discloses or modifies the design of the TOE, its sensitive data or application code by physical (as opposed to logical) tampering means. This threat includes environmental stress, IC failure analysis, electrical probing, unexpected tearing, and side channels. That also includes the modification of the TOE runtime execution through alteration of the intended execution order of (set of) instructions through physical tampering techniques. Directly threatens: all assets.

Table 16: Threats for eUICC

4.2.4 Organizational Security Policies for eUICC

This section describes the organizational security policies to be enforced with respect to the TOE environment. The definition of the grouping is taken from Section 3.4 of [\[PP_eUICC\]](#).

OSP	Description
OSP.LIFE-CYCLE	The TOE must enforce the eUICC life-cycle defined in [SGP22]. In particular: ○ There is a limit on the number of ISD-Ps enabled at a time: • the eUICC supports MEP, the limit is 2 ISD-Ps ○ The eUICC must enforce the profile policy rules (PPR) in case a profile state change is attempted (installation, disabling or

	deletion of a profile), except during the memory reset or test memory reset functions: in this case, the eUICC may disable and delete the currently enabled profile, even if a PPR states that the profile cannot be disabled or deleted; o The eUICC must enforce the rules authorisation table (RAT) before a profile containing PPRs is authorised to be installed on the eUICC.
--	--

Table 17:OSP for eUICC

4.2.5 Assumptions for eUICC

This section introduces the assumptions made on the environment of the eUICC. The definition of the grouping is taken from Section 3.5 of [PP_eUICC].

Assumptions	Description
A.TRUSTED-PATHS-LPAd-IPAd	It is assumed that the interfaces ES10a, ES10b and ES10c (SGP.22) are trusted paths between the eUICC and LPAd/IPAd, when LPAd/IPAd is present and active. It is also assumed that the LPAd/IPAd is a trusted component. It is assumed that LPAd has a means to authenticate the End User (SGP.22).
A.ACTORS	Actors of the infrastructure (eSIM CA, EUM, SM-DP+, SM-DS, and MNO) securely manage their own credentials and otherwise sensitive data. In particular for the overall mobile authentication mechanism defined in 3GPP TS 33.102 [22] to be secure, certain properties need to hold that are outside the scope of the eUICC. In particular, subscriber keys need to be strongly generated and securely managed. The following assumptions are therefore stated: - o The key K is randomly generated during profile preparation and is securely transported to the Authentication Centre belonging to the MNO; - o The random challenge RAND is generated with sufficient entropy in the Authentication Centre belonging to the MNO; - o The Authentication Centre belonging to the MNO generates unique sequence numbers SQN, so that each quintuplet can only be used once; - o Triplets / Quintets are communicated securely between MNOs for roaming
A.APPLICATIONS	The applications shall comply with the security guidelines document for the used platform (operating system). These

	guidelines must substantially describe the application writing style and the platform security mechanisms (e.g. security domains, application firewall) that shall be used to ensure that the applications do not harm the TOE.
--	---

Table 18: Assumptions for eUICC

5 Security Objectives

5.1 Security Objectives for the TOE

This section defines the security objectives to be achieved by the TOE.

5.1.1 Security Objectives for Java Card System

5.1.1.1 Identification

O.SID

The TOE shall uniquely identify every subject (applet, package, or CAP file) before granting it access to any service.

5.1.1.2 Execution

O.FIREWALL

The TOE shall ensure controlled sharing of data containers owned by applets of different CAP files or the JCRE and between applets and the TSFs. See #.FIREWALL for details.

O.GLOBAL_ARRAYS_CONFID

The TOE shall ensure that the APDU buffer that is shared by all applications is always cleared upon applet selection.

The TOE shall ensure that the global byte array used for the invocation of the install method of the selected applet is always cleared after the return from the install method.

O.GLOBAL_ARRAYS_INTEG

The TOE shall ensure that no application can store a reference to the APDU buffer, a global byte array created by the user through makeGlobalArray() method and the byte array used for invocation of the install method of the selected applet.

O.ARRAY_VIEWS_CONFID

The TOE shall ensure that no application can read elements of an array view not having array view security attribute ATTR_READABLE_VIEW.

The TOE shall ensure that an application can only read the elements of the array view within the bounds of the array view.

O.ARRAY_VIEWS_INTEG

The TOE shall ensure that no application can write to an array view not having array view security attribute ATTR_WRITABLE_VIEW.

The TOE shall ensure that an application can only write within the bounds of the array view.

O.NATIVE

The only means that the Java Card VM shall provide for an application to execute native code is the invocation of a method of the Java Card API, or any additional API. See #.NATIVE for details.

O.OPERATE_JCS

The TOE must ensure continued correct operation of its security functions. See #.OPERATE for details.

Note: This objective corresponds to O.OPERATE in [PP JCS], rename in ST to avoid duplication of objective name with the eUICC objective.

O.REALLOCATION

The TOE shall ensure that the re-allocation of a memory block for the runtime areas of the Java Card VM does not disclose any information that was previously stored in that block.

O.RESOURCES

The TOE shall control the availability of resources for the applications. See #.RESOURCES for details.

5.1.1.3 Services

O.ALARM

The TOE shall provide appropriate feedback information upon detection of a potential security violation. See #.ALARM for details.

O.CIPHER

The TOE shall provide a means to cipher sensitive data for applications in a secure way. In particular, the TOE must support cryptographic algorithms consistent with cryptographic usage policies and standards. See #.CIPHER for details.

O.RNG

The TOE shall ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have sufficient entropy.

The TOE shall ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

O.KEY-MNGT

The TOE shall provide a means to securely manage cryptographic keys. This concerns the correct

generation, distribution, access and destruction of cryptographic keys. See #.KEY-MNGT.

O.PIN-MNGT

The TOE shall provide a means to securely manage PIN objects (including the PIN try limit, PIN try counter and states). If the PIN try limit is reached, no further PIN authentication must be allowed. See #.PIN-MNGT for details.

Application Note:

PIN objects may play key roles in the security architecture of client applications. The way they are stored and managed in the memory of the smart card must be carefully considered, and this applies to the whole object rather than the sole value of the PIN. For instance, the try limit and the try counter's value are as sensitive as that of the PIN and the TOE must restrict their modification only to authorized applications such as the card manager.

O.TRANSACTION

The TOE must provide a means to execute a set of operations atomically. See #.TRANSACTION for details.

O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.RNG and O.CIPHER are actually provided to applets in the form of Java Card APIs. Vendor-specific libraries can also be present on the card and made available to applets; those may be built on top of the Java Card API or independently. These proprietary libraries will be evaluated together with the TOE.

5.1.1.4 Object Deletion

O.OBJ-DELETION

The TOE shall ensure the object deletion shall not break references to objects. See #.OBJ-DELETION for further details.

5.1.1.5 Applet Management

O.DELETION

The TOE shall ensure that both applet and CAP file deletion perform as expected. See #.DELETION for details.

O.LOAD

The TOE shall ensure that the loading of a CAP file into the card is safe.

Besides, for code loaded post-issuance, the TOE shall verify the integrity and authenticity evidences generated during the verification of the application CAP file by the verification authority. This

verification by the TOE shall occur during the loading or later during the install process.

Application Note:

Usurpation of identity resulting from a malicious installation of an applet on the card may also be the result of perturbing the communication channel linking the CAD and the card. Even if the CAD is placed in a secure environment, the attacker may try to capture, duplicate, permute or modify the CAP files sent to the card. He may also try to send one of its own applications as if it came from the card issuer. Thus, this objective is intended to ensure the integrity and authenticity of loaded CAP files.

O.INSTALL

The TOE shall ensure that the installation of an applet performs as expected (See #.INSTALL for details).

Besides, for code loaded post-issuance, the TOE shall verify the integrity and authenticity evidences generated during the verification of the application CAP file by the verification authority. If not performed during the loading process, this verification by the TOE shall occur during the install process.

5.1.1.6 Card Management

O.CARD-MANAGEMENT

The card manager shall control the access to card management functions such as the installation, update or deletion of applets. It shall also implement the card issuer's policy on the card.

The card manager is an application with specific rights, which is responsible for the administration of the smart card. This component will in practice be tightly connected with the TOE, which in turn shall very likely rely on the card manager for the effective enforcing of some of its security functions. Typically, the card manager shall be in charge of the life cycle of the whole card, as well as that of the installed applications (applets). The card manager should prevent that card content management (loading, installation, deletion) is carried out, for instance, at invalid states of the card or by non-authorized actors. It shall also enforce security policies established by the card issuer.

5.1.1.7 Smart Card Platform

O.SCP.IC

The SCP shall provide all IC security features against physical attacks.

This security objective for the TOE refers to the point (7) of the security aspect #.SCP:

- It is required that the IC is designed in accordance with a well-defined set of policies and Standards (likely specified in another protection profile), and will be tamper resistant to actually prevent an attacker from extracting or altering security data (like cryptographic keys) by using commonly employed techniques (physical probing and sophisticated analysis of the chip). This especially

matters to the management (storage and operation) of cryptographic keys.

O.SCP.RECOVERY

If there is a loss of power, or if the smart card is withdrawn from the CAD while an operation is in progress, the SCP must allow the TOE to eventually complete the interrupted operation successfully, or recover to a consistent and secure state.

This security objective for the TOE refers to the security aspect #.SCP(1): The smart card platform must be secure with respect to the SFRs. Then after a power loss or sudden card removal prior to completion of some communication protocol, the SCP will allow the TOE on the next power up to either complete the interrupted operation or revert to a secure state.

O.SCP.SUPPORT

The SCP shall support the TSFs of the TOE.

This security objective for the TOE refers to the security aspects 2, 3, 4 and 5 of #.SCP:

(2) It does not allow the TSFs to be bypassed or altered and does not allow access to other low-level functions than those made available by packages of the API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System.

(3) It provides secure low-level cryptographic processing to the Java Card System.

(4) It supports the needs for any update to a single persistent object or class field to be atomic, and possibly a low-level transaction mechanism.

(5) It allows the Java Card System to store data in "persistent technology memory" or in volatile memory, depending on its needs (for instance, transient objects must not be stored in non-volatile memory). The memory model is structured and allows for low-level control accesses (segmentation fault detection).

5.1.1.8 OS Update

O.CONFID-UPDATE-IMAGE.LOAD

The TOE shall ensure that the encrypted image transferred to the device is not disclosed during the installation. The keys used for the image decryption shall be kept confidential.

O.AUTH-LOAD-UPDATE-IMAGE

The TOE shall ensure that it is only possible to load an authorized image.

O.SECURE_LOAD_ACODE

The Loader of the Initial TOE shall check an evidence of authenticity and integrity of the loaded Additional Code. The Loader enforces that only the allowed version of the Additional Code can be

loaded on the Initial TOE. The Loader shall forbid the loading of an Additional Code not intended to be assembled with the Initial TOE. During the Load Phase of an Additional Code, the TOE shall remain secure.

O.SECURE_AC_ACTIVATION

Activation of the Additional Code and update of the Identification Data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the Final TOE shall be completed before activation. If the Atomic Activation is successful, then the resulting product is the Final TOE, otherwise (in case of interruption or incident which prevents the forming of the Final TOE), the Initial TOE shall remain in its initial state or fail secure.

O.TOE_IDENTIFICATION

The Identification Data identifies the Initial TOE and Additional Code. The TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data. After Atomic Activation of the Additional Code, the Identification Data of the Final TOE allows identifications of Initial TOE and Additional Code. The user shall be able to uniquely identify Initial TOE and Additional Code(s) which are embedded in the Final TOE.

5.1.2 Security Objectives for eUICC

The Security Objectives for the eUICC component of the TOE is strictly the same in [\[PP eUICC\]](#).

5.2 Security Objectives for the Operational Environment

This section introduces the security objectives to be achieved by the environment.

5.2.1 Operational Environment of Java Card System

OE.CAP_FILE

No CAP file loaded post-issuance shall contain native methods.

OE.VERIFICATION

All the bytecodes shall be verified at least once, before the loading, before the installation or before the execution, depending on the card capabilities, in order to ensure that each bytecode is valid at execution time. See #. VERIFICATION for details.

Additionally, the applet shall follow all the recommendations, if any, mandated in the platform guidance for maintaining the isolation property of the platform.

Application Note:

Constraints to maintain the isolation property of the platform are provided by the platform developer in application development guidance. The constraints apply to all application code loaded in the platform.

OE.CODE-EVIDENCE

For application code loaded pre-issuance, evaluated technical measures implemented by the TOE or audited organizational measures must ensure that loaded application has not been changed since the code verification required in OE. VERIFICATION.

For application code loaded post-issuance and verified off-card according to the requirements of OE. VERIFICATION, the verification authority shall provide digital evidence to the TOE that the application code has not been modified after the code verification and that he is the actor who performed code verification.

For application code loaded post-issuance and partially or entirely verified on-card, technical measures must ensure that the verification required in OE. VERIFICATION are performed. On-card bytecode verifier is out of the scope of this Security Target.

Application Note:

For application code loaded post-issuance and verified off-card, the integrity and authenticity evidence can be achieved by electronic signature of the application code, after code verification, by the actor

who performed verification.

OE.CONFID_UPDATE_IMAGE.CREATE

Confidentiality of Update Image – CREATE

The off-card Update Image Creator ensures that the confidentiality and integrity requirements are met.

5.2.2 Operational Environment of eUICC

5.2.2.1 Actors

OE.CI

The eSIM CA is a trusted third-party for the purpose of authentication of the entities of the system. The eSIM CA provides certificates for the EUM, SM-DS and SM-DP+. The eSIM CA must ensure the security of its own private keys.

OE.SM-DP+

The SM-DP+ shall be a trusted actor responsible for the data preparation and the associated OTA servers. The SM-DP+ site must be accredited following GSMA SAS.

It must ensure the security of the profiles it manages and loads into the eUICC, including but not limited to:

- MNO keys including OTA keys (telecom keys either generated by the SM-DP+ or by the MNO),
- Application Provider Security Domain keys (APSD keys),
- Controlling Authority Security Domain keys (CASD keys).

The SM-DP+ must ensure that any key used in ISD-P are securely generated before they are transmitted to the eUICC. The SM-DP+ must ensure that any key used in ISD-P are not compromised before they are transmitted to the eUICC.

The security of the ISD-P token verification keys must be ensured by a well defined security policy that covers generation, storage, distribution, destruction and recovery. This policy is enforced by the SM-DP+ in collaboration with the personalizer.

Application Note:

The SM-DP+ replaces the OE.PERSONALIZER as defined in [\[PP USIM\]](#).

OE.SM-DS

The SM-DS shall be a trusted actor responsible for the Discovery Service. The SM-DS site must be accredited following GSMA SAS. The SM-DS has secure communication channels with SM-DP+ or another SM-DS.

The SM-DS must ensure the security of credentials received from the SM-DP+ or another SM-DS.

OE.MNO

The MNOs must ensure that any key used in the profile (ISD-P, MNO SD, and any other SSD) are securely generated before they are transmitted on the eUICC via the MNO OTA Platform. The MNOs must ensure that any key used in the profile (ISD-P, MNO SD, and any other SSD) are not compromised before they are transmitted on the eUICC via the MNO OTA Platform.

Administrators of the mobile operator OTA servers shall be trusted people. They shall be trained to use and administer those servers. They have the means and the equipment to perform their tasks. They must be aware of the sensitivity of the assets they manage and the responsibilities associated with the administration of OTA servers. OTA Platform communication on ES6 makes use of at least a minimum security settings defined for ES5 in [\[SGP 02\]](#) §2.4.

5.2.2.2 Platform**OE.IC.PROOF_OF_IDENTITY**

The underlying IC used by the TOE is uniquely identified.

OE.RE.IDENTITY

The Runtime Environment shall ensure the secure identification of the applications it executes.

OE.TRUSTED-PATHS-LPAd-IPAd

The interfaces ES10a, ES10b and ES10c (SGP.22) are trusted paths between the eUICC and LPAd/IPAd, when LPAd/IPAd is present and active.

5.2.2.3 Profile**OE.APPLICATIONS**

The applications shall comply with the security guidelines document for the platform (operating system) used. These guidelines must substantially describe the application writing style and the platform security mechanisms (e.g. security domains, application firewall) that shall be used to ensure that the applications do not harm the TOE.

Application Note:

The use of these guidelines aims to provide a reasonable assurance that an application will not pose a security risk to another application loaded on this product, even before considering the security features provided by the platform.

This objective implies the objective OE.VERIFICATION from the JCS Protection Profile ([\[PP JCS\]](#)).

Application Note:

In the case when GlobalPlatform is the used platform, the guidelines of [\[GP Applications\]](#) shall be applied.

OE.MNO-SD

The Security Domain U.MNO-SD must use the secure channel SCP80/81 provided by the TOE according to [\[SGP 02\]](#).

5.3 Security Objectives Rationale

5.3.1 Java Card System

5.3.1.1 Rationale for Threats

5.3.1.1.1 Confidentiality

T.CONFID-APPLI-DATA

This threat is countered by the security objective for the operational environment regarding bytecode verification (OE.VERIFICATION). It is also covered by the isolation commitments stated in the (O.FIREWALL) objective. It relies in its turn on the correct identification of applets stated in (O.SID). Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (O.OPERATE_JCS) objective.

As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.

The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE_JCS and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

As applets may need to share some data or communicate with the CAD, cryptographic functions are required to actually protect the exchanged information (O.CIPHER, O.RNG). Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them. Keys, PIN's are particular cases of an application's sensitive data (the Java Card System may possess keys as well) that ask for appropriate management (O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION). If the PIN class of the Java Card API is used, the objective (O.FIREWALL) shall contribute in covering this threat by controlling the sharing of the global PIN between the applets.

Other application data that is sent to the applet as clear text arrives to the APDU buffer, which is a resource shared by all applications. The disclosure of such data is prevented by the security objective

O.GLOBAL_ARRAYS_CONFID.

An applet might share data buffer with another applet using array views without the array view security attribute `ATTR_READABLE_VIEW`. The disclosure of data of the applet creating the array view is prevented by the security object `O.ARRAY_VIEWS_CONFID`.

Finally, any attempt to read a piece of information that was previously used by an application but has been logically deleted is countered by the `O.REALLOCATION` objective. That objective states that any information that was formerly stored in a memory block shall be cleared before the block is reused.

T.CONFID-JCS-CODE

This threat is countered by the list of properties described in the (`#.VERIFICATION`) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of those instructions enables reading a piece of code, no Java Card applet can therefore be executed to disclose a piece of code. Native applications are also harmless because of the objective `O.NATIVE`, so no application can be run to disclose a piece of code.

The (`#.VERIFICATION`) security aspect is addressed in this Security Target by the objective for the environment `OE.VERIFICATION`.

The objectives `O.CARD-MANAGEMENT` and `OE.VERIFICATION` contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

T.CONFID-JCS-DATA

This threat is covered by bytecode verification (`OE.VERIFICATION`) and the isolation commitments stated in the (`O.FIREWALL`) security objective. This latter objective also relies in its turn on the correct identification of applets stated in (`O.SID`).

Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (`O.OPERATE_JCS`) objective.

As the firewall is a software tool automating critical controls, the objective `O.ALARM` asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.

The objectives `O.CARD-MANAGEMENT` and `OE.VERIFICATION` contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objectives `O.SCP.RECOVERY` and `O.SCP.SUPPORT` are intended to support the `O.OPERATE_JCS` and `O.ALARM` objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

5.3.1.1.2 Integrity

T.INTEG-APPLI-CODE

This threat is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code. Native applications are also harmless because of the objective O.NATIVE, so no application can run to modify a piece of code.

The (#.VERIFICATION) security aspect is addressed in this configuration by the objective for the environment OE.VERIFICATION.

The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that integrity and authenticity evidences exist for the application code loaded into the platform.

T.INTEG-APPLI-CODE.LOAD

This threat is countered by the security objective O.LOAD which ensures that the loading of CAP files is done securely and thus preserves the integrity of CAP files' code.

The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. By controlling the access to card management functions such as the installation, update or deletion of applets the objective O.CARD-MANAGEMENT contributes to cover this threat.

T.INTEG-APPLI-DATA

This threat is countered by bytecode verification (OE.VERIFICATION) and the isolation commitments stated in the (O.FIREWALL) objective. This latter objective also relies in its turn on the correct identification of applets stated in (O.SID). Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (O.OPERATE_JCS) objective.

As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.

The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application

code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE_JCS and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

Concerning the confidentiality and integrity of application sensitive data, as applets may need to share some data or communicate with the CAD, cryptographic functions are required to actually protect the exchanged information (O.CIPHER, O.RNG). Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them. Keys and PIN's are particular cases of an application's sensitive data (the Java Card System may possess keys as well) that ask for appropriate management (O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION). If the PIN class of the Java Card API is used, the objective (O.FIREWALL) is also concerned.

Other application data that is sent to the applet as clear text arrives to the APDU buffer, which is a resource shared by all applications. The integrity of the information stored in that buffer is ensured by the objective O.GLOBAL_ARRAYS_INTEG.

An applet might share data buffer with another applet using array views without the array view security attribute ATTR_WRITABLE_VIEW. The integrity of data of the applet creating the array view is ensured by the security objective O.ARRAY_VIEWS_INTEG.

Finally, any attempt to read a piece of information that was previously used by an application but has been logically deleted is countered by the O.REALLOCATION objective. That objective states that any information that was formerly stored in a memory block shall be cleared before the block is reused.

T.INTEG-APPLI-DATA.LOAD

This threat is countered by the security objective O.LOAD which ensures that the loading of CAP files is done securely and thus preserves the integrity of applications data.

The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. By controlling the access to card management functions such as the installation, update or deletion of applets the objective O.CARD-MANAGEMENT contributes to cover this threat.

T.INTEG-JCS-CODE

This threat is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code. Native applications are also harmless because of the objective O.NATIVE, so no application can be run to modify a piece of code.

The (#.VERIFICATION) security aspect is addressed in this configuration by the objective for the

environment OE.VERIFICATION.

The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.

T.INTEG-JCS-DATA

This threat is countered by bytecode verification (OE.VERIFICATION) and the isolation commitments stated in the (O.FIREWALL) objective. This latter objective also relies in its turn on the correct identification of applets stated in (O.SID). Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (O.OPERATE_JCS) objective.

As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken.

The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE_JCS and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

5.3.1.1.3 Identity Usurpation

T.SID.1

As impersonation is usually the result of successfully disclosing and modifying some assets, this threat is mainly countered by the objectives concerning the isolation of application data (like PINs), ensured by the (O.FIREWALL). Uniqueness of subject-identity (O.SID) also participates to face this threat. It should be noticed that the AIDs, which are used for applet identification, are TSF data.

In this configuration, usurpation of identity resulting from a malicious installation of an applet on the card is covered by the objective O.INSTALL.

The installation parameters of an applet (like its name) are loaded into a global array that is also shared by all the applications. The disclosure of those parameters (which could be used to impersonate the applet) is countered by the objectives O.GLOBAL_ARRAYS_CONFID and O.GLOBAL_ARRAYS_INTEG.

The objective O.CARD-MANAGEMENT contributes, by preventing usurpation of identity resulting from a malicious installation of an applet on the card, to counter this threat.

T.SID.2

This is covered by integrity of TSF data, subject-identification (O.SID), the firewall (O.FIREWALL) and its good working order (O.OPERATE_JCS).

The objective O.INSTALL contributes to counter this threat by ensuring that installing an applet has no effect on the state of other applets and thus can't change the TOE's attribution of privileged roles.

The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE_JCS objective of the TOE, so they are indirectly related to the threats that this latter objective contributes to counter.

5.3.1.1.4 Unauthorized Execution

T.EXE-CODE.1

Unauthorized execution of a method is prevented by the objective OE.VERIFICATION. This threat particularly concerns the point (8) of the security aspect #.VERIFICATION (access modifiers and scope of accessibility for classes, fields and methods). The O.FIREWALL objective is also concerned, because it prevents the execution of non-shareable methods of a class instance by any subject apart from the class instance owner.

T.EXE-CODE.2

Unauthorized execution of a method fragment or arbitrary data is prevented by the objective OE.VERIFICATION. This threat particularly concerns those points of the security aspect related to control flow confinement and the validity of the method references used in the bytecodes.

T.NATIVE

This threat is countered by O.NATIVE which ensures that a Java Card applet can only access native methods indirectly that is, through an API. OE.CAP_FILE also covers this threat by ensuring that no CAP files containing native code shall be loaded in post-issuance. In addition to this, the bytecode verifier also prevents the program counter of an applet to jump into a piece of native code by confining the control flow to the currently executed method (OE.VERIFICATION).

5.3.1.1.5 Denial of Service

T.RESOURCES

This threat is directly countered by objectives on resource-management (O.RESOURCES) for runtime purposes and good working order (O.OPERATE_JCS) in a general manner.

Consumption of resources during installation and other card management operations are covered, in case of failure, by O.INSTALL.

It should be noticed that, for what relates to CPU usage, the Java Card platform is single-threaded and it is possible for an ill-formed application (either native or not) to monopolize the CPU. However, a smart card can be physically interrupted (card removal or hardware reset) and most CADs implement a timeout policy that prevent them from being blocked should a card fails to answer. That point is out of scope of this Security Target, though.

Finally, the objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE_JCS and O.RESOURCES objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

5.3.1.1.6 Card Management

T.DELETION

This threat is covered by the O.DELETION security objective which ensures that both applet and CAP file deletion perform as expected.

The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

T.INSTALL

This threat is covered by the security objective O.INSTALL which ensures that the installation of an applet performs as expected and the security objectives O.LOAD which ensures that the loading of a CAP file into the card is safe.

The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

5.3.1.1.7 Services

T.OBJ-DELETION

This threat is covered by the O.OBJ-DELETION security objective which ensures that object deletion shall not break references to objects.

5.3.1.1.8 Miscellaneous

T.PHYSICAL

Covered by O.SCP.IC. Physical protections rely on the underlying platform and are therefore an environmental issue.

5.3.1.1.9 OS Update

T.CONFID-UPDATE-IMAGE.LOAD

O.CONFID-UPDATE-IMAGE.LOAD	Counters the threat by ensuring the confidentiality of D.UPDATE_IMAGE during installing it on the TOE.
OE.CONFID_UPDATE_IMAGE.CREATE	Counters the threat by ensuring that the D.UPDATE_IMAGE is not transferred in plain and that the keys are kept secret.

T.INTEG-UPDATE-IMAGE.LOAD

O.SECURE_LOAD_ACODE	Counters the threat directly by ensuring the authenticity and integrity of D.UPDATE_IMAGE.
---------------------	--

T.UNAUTH-UPDATE-IMAGE.LOAD

O.SECURE_LOAD_ACODE	Counters the threat directly by ensuring that only authorized (allowed version) images can be installed.
O.AUTH-LOAD-UPDATE-IMAGE	Counters the threat by ensuring the confidentiality of D.OS-UPDATE_KEY(S) during installing it on the TOE.

T.INTERRUPT_OSU

O.SECURE_LOAD_ACODE	Counters the threat directly by ensuring that the TOE remains in a secure state after interruption of the OS Update procedure (Load Phase).
O.TOE_IDENTIFICATION	Counters the threat directly by ensuring that D.TOE_IDENTIFICATION is only updated after successful OS Update procedure.
O.SECURE_AC_ACTIVATION	Counters the threat directly by ensuring that the update OS is only activated after successful (atomic) OS Update procedure.

5.3.1.2 Rationale for OSPs

OSP.VERIFICATION

This policy is upheld by the security objective of the environment OE.VERIFICATION which guarantees that all the bytecodes shall be verified at least once, before the loading, before the installation or before the execution in order to ensure that each bytecode is valid at execution time.

This policy is also upheld by the security objective of the environment OE.CODE-EVIDENCE which ensures that evidences exist that the application code has been verified and not changed after verification, and by the security objective for the TOE O.LOAD which shall ensure that the loading of a CAP file into the card is safe.

5.3.1.3 Rationale for Assumptions

A.CAP_FILE

This assumption is upheld by the security objective for the operational environment OE.CAP_FILE which ensures that no CAP file loaded post-issuance shall contain native methods.

A.VERIFICATION

This assumption is upheld by the security objective on the operational environment OE.VERIFICATION which guarantees that all the bytecodes shall be verified at least once, before the loading, before the installation or before the execution in order to ensure that each bytecode is valid at execution time.

This assumption is also upheld by the security objective of the environment OE.CODE-EVIDENCE which ensures that evidences exist that the application code has been verified and not changed after verification.

5.3.1.4 SPD and Security Objectives

Threats	Security Objectives	Rationale
T.CONFID-APPLI-DATA	O.SCP.RECOVERY,O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION,O.SID, O.OPERATE_JCS, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.ARRAY_VIEWS_CONFID, O.ALARM, O.TRANSACTION, O.CIPHER,O.RNG,O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION	Section 5.3.1.1
T.CONFID-JCS-CODE	OE.VERIFICATION, O.CARD-MANAGEMENT, O.NATIVE	Section 5.3.1.1
T.CONFID-JCS-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE_JCS, O.FIREWALL, O.ALARM	Section 5.3.1.1
T.INTEG-APPLI-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE	Section 5.3.1.1

T.INTEG-APPLI-CODE.LOAD	O.LOAD, O.CARD-MANAGEMENT, OE.CODE-EVIDENCE	Section 5.3.1.1
T.INTEG-APPLI-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION,O.SID, O.OPERATE_JCS, O.FIREWALL,O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG,O.ALARM, O.TRANSACTION,O.CIPHER,O.RNG, O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION, OE.CODE-EVIDENCE,	Section 5.3.1.1
T.INTEG-APPLI-DATA.LOAD	O.LOAD, O.CARD-MANAGEMENT, OE.CODE-EVIDENCE	Section 5.3.1.1
T.INTEG-JCS-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE	Section 5.3.1.1
T.INTEG-JCS-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE_JCS, O.FIREWALL, O.ALARM, OE.CODE-EVIDENCE	Section 5.3.1.1
T.SID.1	O.CARD-MANAGEMENT, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.INSTALL, O.SID	Section 5.3.1.1
T.SID.2	O.SCP.RECOVERY, O.SCP.SUPPORT, O.SID, O.OPERATE_JCS, O.FIREWALL, O.INSTALL	Section 5.3.1.1
T.EXE-CODE.1	OE.VERIFICATION, O.FIREWALL	Section 5.3.1.1
T.EXE-CODE.2	OE.VERIFICATION	Section 5.3.1.1
T.NATIVE	OE.VERIFICATION, OE.CAP_FILE , O.NATIVE	Section 5.3.1.1
T.RESOURCES	O.INSTALL, O.OPERATE_JCS, O.RESOURCES, O.SCP.RECOVERY, O.SCP.SUPPORT	Section 5.3.1.1
T.DELETION	O.DELETION, O.CARD-MANAGEMENT	Section 5.3.1.1

T.INSTALL	O.INSTALL, O.LOAD, O.CARD-MANAGEMENT	Section 5.3.1.1
T.OBJ-DELETION	O.OBJ-DELETION	Section 5.3.1.1
T.PHYSICAL	O.SCP.IC	Section 5.3.1.1
T.CONFID-UPDATE-IMAGE. LOAD	O.CONFID-UPDATE-IMAGE.LOAD OE.CONFID_UPDATE_IMAGE.CREATE	Section 5.3.1.1
T.INTEG-UPDATE-IMAGE. LOAD	O.SECURE_LOAD_ACODE	Section 5.3.1.1
T.UNAUTH-UPDATE-IMAGE. LOAD	O.SECURE_LOAD_ACODE O.AUTH-LOAD-UPDATE-IMAGE	Section 5.3.1.1
T.INTERRUPT_OSU	O.SECURE_LOAD_ACODE O.TOE_IDENTIFICATION O.SECURE_AC_ACTIVATION	Section 5.3.1.1

Table 19: Threats and Security Objectives - Coverage for JavaCard

Note: Due to the conformance statement in section 2.3.2.2, some of the eUICC security objectives are replaced by Java Card security objectives, so these Java Card security objectives map the eUICC threats and are supplemented in Table 20 to be consistent with Table 25.

Security Objectives	Threats
O.SID	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.1, T.SID.2, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
O.FIREWALL	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.1, T.SID.2, T.EXE-CODE.1, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-INTERCEPTION, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.IDENTITY-INTERCEPTION, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK

O.GLOBAL_ARRAYS_CONFID	T.CONFID-APPLI-DATA, T.SID.1, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK, T.PHYSICAL-ATTACK
O.GLOBAL_ARRAYS_INTEG	T.INTEG-APPLI-DATA, T.SID.1, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK
O.ARRAY_VIEWS_CONFID	T.CONFID-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK, T.PHYSICAL-ATTACK
O.ARRAY_VIEWS_INTEG	T.INTEG-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK
O.NATIVE	T.CONFID-JCS-CODE, T.INTEG-APPLI-CODE, T.INTEG- JCS-CODE, T.NATIVE, T.UNAUTHORIZED- PROFILE-MNG, T.UNAUTHORIZED- PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL-ATTACK
O.OPERATE_JCS	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK, T.PHYSICAL-ATTACK
O.REALLOCATION	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK, T.PHYSICAL-ATTACK

O.RESOURCES	T.RESOURCES
O.ALARM	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK, T.PHYSICAL-ATTACK
O.CIPHER	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-INTERCEPTION, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.IDENTITY-INTERCEPTION, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
O.RNG	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-INTERCEPTION, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.IDENTITY-INTERCEPTION, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
O.KEY-MNGT	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-INTERCEPTION, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.IDENTITY-INTERCEPTION, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
O.PIN-MNGT	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-INTERCEPTION, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.IDENTITY-INTERCEPTION, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
O.TRANSACTION	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG,

	T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK, T.PHYSICAL-ATTACK
O.OBJ-DELETION	T.OBJ-DELETION
O.DELETION	T.DELETION
O.LOAD	T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI- DATA.LOAD, T.INSTALL, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK
O.INSTALL	T.SID.1, T.SID.2, T.RESOURCES, T.INSTALL
OE.CAP_FILE	T.NATIVE, T.LOGICAL-ATTACK
O.CARD-MANAGEMENT	T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-APPLI-CODE, T.INTEG- APPLI-CODE.LOAD, T.INTEG-APPLI-DATA, T.INTEG- APPLI-DATA.LOAD, T.INTEG-JCS-CODE, T.INTEG-JCS- DATA, T.SID.1, T.DELETION, T.INSTALL, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK, T.PHYSICAL-ATTACK
O.SCP.IC	T.PHYSICAL
O.SCP.RECOVERY	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-INTERCEPTION, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.IDENTITY-INTERCEPTION, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
O.SCP.SUPPORT	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-INTERCEPTION, T.PROFILE-MNG-ELIGIBILITY,

	T.UNAUTHORIZED-IDENTITY-MNG, T.IDENTITY-INTERCEPTION, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
OE.VERIFICATION	T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-APPLI-CODE, T.INTEG- APPLI-DATA, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, T.EXE-CODE.1, T.EXE-CODE.2, T.NATIVE, T.UNAUTHORIZED- PROFILE-MNG, T.UNAUTHORIZED- PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL-ATTACK, T.PHYSICAL-ATTACK
OE.CODE-EVIDENCE	T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, T.UNAUTHORIZED-PROFILE-MNG, T.UNAUTHORIZED-PLATFORM-MNG, T.PROFILE-MNG-ELIGIBILITY, T.UNAUTHORIZED-IDENTITY-MNG, T.LOGICAL- ATTACK
O.CONFID-UPDATE-IMAGE.LOAD	T.CONFID-UPDATE-IMAGE.LOAD
OE.CONFID_UPDATE_IMAGE.C REATE	T.CONFID-UPDATE-IMAGE.LOAD
O.SECURE_LOAD_ACODE	T.INTEG-UPDATE-IMAGE.LOAD T.UNAUTH-UPDATE-IMAGE.LOAD T.INTERRUPT_OSU
O.AUTH-LOAD-UPDATE-IMAGE	T.UNAUTH-UPDATE-IMAGE.LOAD
O.TOE_IDENTIFICATION	T.INTERRUPT_OSU
O.SECURE_AC_ACTIVATION	T.INTERRUPT_OSU

Table 20:Security Objectives and Threats - Coverage for JavaCard

Organisational Security Policies	Security Objectives	Rationale
OSP.VERIFICATION	OE.VERIFICATION, O.LOAD, OE.CODE-EVIDENCE	<u>Section 5.3.1.2</u>

Table 21:OSPs and Security Objectives - Coverage for JavaCard

Security Objectives	Organisational Security Policies
O.SID	
O.FIREWALL	
O.GLOBAL_ARRAYS_CONFID	
O.GLOBAL_ARRAYS_INTEG	
O.ARRAY_VIEWS_CONFID	

O.ARRAY_VIEWS_INTEG	
O.NATIVE	
O.OPERATE_JCS	
O.REALLOCATION	
O.RESOURCES	
O.ALARM	
O.CIPHER	
O.RNG	
O.KEY-MNGT	
O.PIN-MNGT	
O.TRANSACTION	
O.OBJ-DELETION	
O.DELETION	
O.LOAD	OSP.VERIFICATION
O.INSTALL	
OE.CAP_FILE	
O.CARD-MANAGEMENT	
O.SCP.IC	
O.SCP.RECOVERY	
O.SCP.SUPPORT	
OE.VERIFICATION	OSP.VERIFICATION
OE.CODE-EVIDENCE	OSP.VERIFICATION

Table 22:Security Objectives and OSPs - Coverage for JavaCard

Assumptions	Security Objectives for the Operational Environment	Rationale
A.CAP_FILE	OE.CAP_FILE	Section 5.3.1.3
A.VERIFICATION	OE.VERIFICATION,OE.CODE-EVIDENCE	Section 5.3.1.3

Table 23:Assumptions and Security Objectives for the Operational Environment - Coverage for JavaCard

Security Objectives for the Operational Environment	Assumptions
OE.CAP_FILE	A.CAP_FILE
OE.VERIFICATION	A.VERIFICATION
OE.CODE-EVIDENCE	A.VERIFICATION

Table 24:Security Objectives for the Operational Environment and Assumptions -Coverage for JavaCard

5.3.2 Security Objective Rational related to eUICC

5.3.2.1 Threats

5.3.2.1.1 Unauthorized profile and platform management

T.UNAUTHORIZED-PROFILE-MNG This threat is covered by requiring authentication and authorization from the legitimate actors:

- O.PRE-PPI and O.eUICC-DOMAIN-RIGHTS ensure that only authorized and authenticated actors (SM-DP+ and MNO OTA Platform) will access the Security Domains functions and content;
 - OE.SM-DP+ and OE.MNO protect the corresponding credentials when used off- card.
- The on-card access control policy relies upon the underlying Runtime Environment, which ensures confidentiality and integrity of application data (OE.VERIFICATION, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.CIPHER, O.OPERATE_JCS, O.NATIVE, O.RNG, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE-EVIDENCE, O.LOAD).

The authentication is supported by corresponding secure channels:

- O.SECURE-CHANNELS and O.INTERNAL-SECURE-CHANNELS provide a secure channel for communication with SM-DP+ and a secure channel for communication with MNO OTA Platform. These secure channels rely upon the underlying Runtime Environment, which protects the applications communications (O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.RNG, O.KEY-MNGT and O.PIN-MNGT).

Since the MNO-SD Security Domain is not part of the TOE, the operational environment has to guarantee that it will use securely the SCP80/81 secure channel provided by the TOE (OE.MNO-SD).

In order to ensure the secure operation of the Application Firewall, the following objectives for the operational environment are also required:

- compliance to security guidelines for applications (OE.APPLICATIONS).

T.UNAUTHORIZED-PLATFORM-MNG This threat is covered by requiring authentication and authorization from the legitimate actors:

- O.PRE-PPI and O.eUICC-DOMAIN-RIGHTS ensure that only authorized and authenticated actors will access the Security Domains functions and content.
- OE.SM-DP+ protect the corresponding credentials when used off- card.

The on-card access control policy relies upon the underlying Runtime Environment, which ensures confidentiality and integrity of application data (OE.VERIFICATION, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.CIPHER, O.OPERATE_JCS, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT, O.NATIVE, O.RNG,

O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE-EVIDENCE, O.LOAD).

In order to ensure the secure operation of the Application Firewall, the following objectives for the operational environment are also required:

- compliance to security guidelines for applications (OE.APPLICATIONS).

T.PROFILE-MNG-INTERCEPTION Commands and profiles are transmitted by the SM-DP+ to its on-card representative (ISD-P), while profile data (including meta-data such as PPRs) is also transmitted by the MNO OTA Platform to its on-card representative (MNO-SD) by means of RPM requests from Profile owner to ISD-R (UpdateMetadataRequest).

Consequently, the TSF ensures:

- Security of the transmission to the Security Domain (O.SECURE-CHANNELS and O.INTERNAL-SECURE-CHANNELS) by requiring authentication from SM-DP+ and MNO OTA Platforms, and protecting the transmission from unauthorized disclosure, modification and replay. These secure channels rely upon the underlying Runtime Environment, which protects the applications communications (O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.RNG, O.KEY-MNGT, O.PIN-MNGT).

Since the MNO-SD Security Domain is not part of the TOE, the operational environment has to guarantee that it will securely use the SCP80/81 secure channel provided by the TOE (OE.MNO-SD).

OE.SM-DP+ and OE.MNO ensure that the credentials related to the secure channels will not be disclosed when used by off-card actors.

T.PROFILE-MNG-ELIGIBILITY Device Info and eUICCInfo2, transmitted by the eUICC to the SM-DP+, are used by the SM-DP+ to perform the Eligibility Check prior to allowing profile download onto the eUICC.

Consequently, the TSF ensures:

- Security of the transmission to the Security Domain (O.SECURE-CHANNELS and O.INTERNAL-SECURE-CHANNELS) by requiring authentication from SM-DP+, and protecting the transmission from unauthorized disclosure, modification and replay. These secure channels rely upon the underlying Runtime Environment, which protects the applications communications (O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.RNG, O.KEY-MNGT, O.PIN-MNGT).

OE.SM-DP+ ensures that the credentials related to the secure channels will not be disclosed when used by off-card actors.

O.DATA-INTEGRITY, O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE, O.LOAD, O.SCP.RECOVERY, O.SCP.SUPPORT, O.SID, O.OPERATE_JCS, O.FIREWALL, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG,

O.ALARM, O.TRANSACTION, O.CIPHER, O.RNG, O.PIN-MNGT, O.KEY-MNGT and O.REALLOCATION ensure that the integrity of Device Info and eUICCInfo2 is protected at the eUICC level.

5.3.2.1.2 Identity tampering

T.UNAUTHORIZED-IDENTITY-MNG O.PRE-PPI and O.eUICC-DOMAIN-RIGHTS covers this threat by providing an access control policy for ECASD content and functionality. The on- card access control policy relies upon the underlying Runtime Environment, which ensures confidentiality and integrity of application data (OE.VERIFICATION, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.CIPHER, O.OPERATE_JCS, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT, O.NATIVE, O.RNG, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE-EVIDENCE, O.LOAD).

OE.RE.IDENTITY ensures that at the Java Card level, the applications cannot impersonate other actors or modify their privileges.

T.IDENTITY-INTERCEPTION O.INTERNAL-SECURE-CHANNELS ensures the secure transmission of the shared secrets from the ECASD to ISD-R and ISD-P. These secure channels rely upon the underlying Runtime Environment, which protects the applications communications (O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.RNG, O.KEY-MNGT, O.PIN-MNGT).

OE.CI ensures that the eSIM CA will manage securely its credentials off-card.

5.3.2.1.3 eUICC cloning

T.UNAUTHORIZED-eUICC O.PROOF_OF_IDENTITY guarantees that the off-card actor can be provided with a cryptographic proof of identity based on an EID.

O.PROOF_OF_IDENTITY guarantees this EID uniqueness by basing it on the eUICC hardware identification (which is unique due to OE.IC.PROOF_OF_IDENTITY).

5.3.2.1.4 LPAd impersonation

T.LPAd-INTERFACE-EXPLOIT OE.TRUSTED-PATHS-LPAd-IPAd ensures that the interfaces ES10a, ES10b and ES10c (SGP.22) are trusted paths to the LPAd/IPA.

5.3.2.1.5 Unauthorized access to the mobile network

T.UNAUTHORIZED-MOBILE-ACCESS The objective O.ALGORITHMS ensures that a profile may only access the mobile network using a secure authentication method, which prevents impersonation by an attacker.

5.3.2.1.6 Second level threats

T.LOGICAL-ATTACK This threat is covered by controlling the information flow between Security Domains and the PRE, PPI, the Telecom Framework or any native/OS part of the TOE. As such it is covered:

- by the APIs provided by the Runtime Environment (OE.VERIFICATION, O.NATIVE);
- by the APIs of the TSF (O.API); the APIs of Telecom Framework, PRE and PPI shall ensure atomic transactions (O.SCP.SUPPORT).

Whenever sensitive data of the TOE are processed by applications, confidentiality and integrity must be protected at all times by the Runtime Environment (OE.VERIFICATION, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.CIPHER, O.OPERATE_JCS, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT, O.NATIVE, O.RNG, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE-EVIDENCE, O.LOAD). However these sensitive data are also processed by the PPE, PPI and the Telecom Framework, which are not protected by these mechanisms. Consequently,

- the TOE itself must ensure the correct operation of PRE, PPI and Telecom Framework (O.OPERATE), and
- PRE, PPI and Telecom Framework must protect the confidentiality and integrity of the sensitive data they process (O.DATA-CONFIDENTIALITY, O.DATA-INTEGRITY).

The following objectives are also required:

- prevention of unauthorized code execution by applications (O.FIREWALL, OE.VERIFICATION, O.NATIVE, OE.CAP_FILE),
- compliance to security guidelines for applications (OE.APPLICATIONS).

T.PHYSICAL-ATTACK This threat is countered mainly by physical protections which rely on the underlying Platform.

The security objectives O.SCP.SUPPORT and O.SCP.RECOVERY protect sensitive assets of the Platform against loss of integrity and confidentiality and especially ensure the TSFs cannot be

bypassed or altered.

In particular, the security objective O.SCP.SUPPORT provides functionality to ensure atomicity of sensitive operations, secure low level access control and protection against bypassing of the security features of the TOE. In particular, it explicitly ensures the independent protection in integrity of the Platform data.

Since the TOE cannot only rely on the IC protection measures, the TOE shall enforce any necessary mechanism to ensure resistance against side channels (O.DATA- CONFIDENTIALITY). For the same reason, the Runtime Environment (to which Java Card System can be an implementation) security architecture must cover side channels (OE.VERIFICATION, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.CIPHER, O.OPERATE_JCS, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.CARD-MANAGEMENT, O.SCP.RECOVERY, O.RNG and O.SCP.SUPPORT,.).

5.3.2.2 Organisational Security Policies

5.3.2.2.1 Life-cycle

OSP.LIFE-CYCLE O.PRE-PPI ensures that there is a single ISD-P enabled at a time if the eUICC supports only SEP (Single Enabled Profile).

The profile deletion capability relies on the secure application deletion mechanisms provided by O.DELETION, O.INSTALL, O.CARD-MANAGEMENT and O.LOAD.

O.OPERATE contributes to this OSP by ensuring that the Platform security functions are always enforced.

5.3.2.3 Assumptions

5.3.2.3.1 Device assumptions

A.TRUSTED-PATHS-LPAd-IPAd This assumption is upheld by OE.TRUSTED-PATHS-LPAd-IPAd.

5.3.2.3.2 Miscellaneous

A.ACTORS This assumption is upheld by objectives OE.CI, OE.SM-DP+, OE.SM-DS and OE.MNO which ensure that credentials and otherwise sensitive data will be managed correctly by each actor of the infrastructure.

A.APPLICATIONS This assumption is directly upheld by objective OE.APPLICATIONS.

5.3.2.4 SPD and Security Objectives

Threats	Security Objectives	Rationale
T.UNAUTHORIZED-PROFILE-MNG	O.eUICC-DOMAIN-RIGHTS, OE.SM-DP+, OE.MNO, O.PRE-PPI, O.SECURE-CHANNELS, OE.APPLICATIONS, O.INTERNAL-SECURE-CHANNELS, O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.KEY-MNGT, O.PIN-MNGT, OE.VERIFICATION, O.NATIVE, O.RNG, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.OPERATE_JCS, O.TRANSACTION, O.CARD-MANAGEMENT, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE-EVIDENCE, O.LOAD, OE.MNO-SD	Section 5.3.2.1
T.UNAUTHORIZED-PLATFORM-MNG	O.eUICC-DOMAIN-RIGHTS, O.PRE-PPI, OE.APPLICATIONS, O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.KEY-MNGT, O.PIN-MNGT, OE.VERIFICATION, O.NATIVE, O.RNG, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.OPERATE_JCS, O.TRANSACTION, O.CARD-MANAGEMENT, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE-EVIDENCE, O.LOAD, OE.SM-DP+	Section 5.3.2.1
T.PROFILE-MNG-	OE.SM-DP+, OE.MNO, O.SECURE-CHANNELS,	Section 5.3.2.1

INTERCEPTION	O.INTERNAL-SECURE-CHANNELS,O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.RNG, O.KEY-MNGT, O.PIN-MNGT, OE.MNO-SD	
T.PROFILE-MNG- ELIGIBILITY	OE.SM-DP+, O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.KEY-MNGT, O.PIN-MNGT, O.SECURE-CHANNELS, O.INTERNAL-SECURE-CHANNELS, O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE, O.LOAD, O.SID, O.OPERATE_JCS, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, O.ALARM, O.TRANSACTION,O.RNG, O.REALLOCATION,O.DATA-INTEGRITY	Section 5.3.2.1
T.UNAUTHORIZED- IDENTITY-MNG	O.eUICC-DOMAIN-RIGHTS, O.PRE-PPI O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.KEY-MNGT, O.PIN-MNGT, OE.VERIFICATION, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.OPERATE_JCS,O.NATIVE,O.RNG, O.TRANSACTION, O.CARD-MANAGEMENT, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE- EVIDENCE,O.LOAD, OE.RE.IDENTITY	Section 5.3.2.1
T.IDENTITY- INTERCEPTION	OE.CI, O.INTERNAL-SECURE-CHANNELS, O.FIREWALL, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.RNG, O.KEY-MNGT, O.PIN-MNGT	Section 5.3.2.1
T.UNAUTHORIZED- eUICC	O.PROOF_OF_IDENTITY, OE.IC.PROOF_OF_IDENTITY	Section 5.3.2.1
T.LPAd-INTERFACE- EXPLOIT	OE.TRUSTED-PATHS-LPAd-IPAd	Section 5.3.2.1
T.UNAUTHORIZED- MOBILE-ACCESS	O.ALGORITHMS	Section 5.3.2.1
T.LOGICAL-ATTACK	O.DATA-CONFIDENTIALITY, O.DATA-INTEGRITY, O.API, OE.APPLICATIONS, O.OPERATE, OE.VERIFICATION, O.NATIVE, O.FIREWALL, O.SCP.RECOVERY,	Section 5.3.2.1

	O.SCP.SUPPORT, O.CIPHER, O.KEY-MNGT, O.PIN-MNGT, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.OPERATE_JCS, O.RNG, O.TRANSACTION, O.CARD-MANAGEMENT, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, OE.CODE- EVIDENCE, O.LOAD, OE.CAP_FILE	
T.PHYSICAL- ATTACK	O.SCP.SUPPORT, O.SCP.RECOVERY, O.DATA-CONFIDENTIALITY, OE. VERIFICATION, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.SID, O.ARRAY_VIEWS_CONFID, O.REALLOCATION, O.ALARM, O.CIPHER, O.RNG, O.OPERATE_JCS, O.KEY-MNGT, O.PIN- MNGT, O.TRANSACTION, O.CARD- MANAGEMENT	Section 5.3.2.1

Table 25: Threats and Security Objectives – Coverage for eUICC

Organisational Security Policies	Security Objectives	Rationale
OSP.LIFE-CYCLE	O.PRE-PPI, O.DELETION, O.INSTALL, O.CARD- MANAGEMENT , O.LOAD, O.OPERATE	Section 5.3.2.2

Table 26: OSPs and Security Objectives – Coverage for eUICC

Assumptions	Security Objectives for the Operational Environment	Rationale
A.TRUSTED-PATHS-LPAd-IPAd	OE.TRUSTED-PATHS-LPAd-IPAd	Section 5.3.2.3
A.ACTORS	OE.CI, OE.SM-DP+, OE.MNO, OE.SM-DS	Section 5.3.2.3
A.APPLICATIONS	OE.APPLICATIONS	Section 5.3.2.3

Table 27: Assumptions and Security Objectives for the Operational Environment – Coverage for eUICC

6 Extended Components Definition

No extended components have been applied in this TOE.

7 Security Requirements

7.1 Security Requirements for Java Card System

7.1.1 Security Functional Requirements

This section states the security functional requirements for the [\[PP JCS\]](#).

Group	Description
Core with Logical Channels (CoreG_LC)	The CoreG_LC contains the requirements concerning the runtime environment of the Java Card System implementing logical channels. This includes the firewall policy and the requirements related to the Java Card API. Logical channels are a Java Card specification version 2.2 feature.
Installation (InstG)	The InstG contains the security requirements concerning the installation of post-issuance applications. It does not address card management issues in the broad sense, but only those security aspects of the installation procedure that are related to applet execution.
Applet deletion (ADELG)	The ADELG contains the security requirements for erasing installed applets from the card, a feature introduced in Java Card specification version 2.2.
Object deletion (ODELG)	The ODELG contains the security requirements for the object deletion capability. This provides a safe memory recovering mechanism. This is a Java Card specification version 2.2 feature.
Secure carrier (CarG)	The CarG group contains minimal requirements for secure downloading of applications on the card. This group contains the security requirements for preventing, the installation of a CAP file that has not been bytecode verified, or that has been modified after bytecode verification.
Smart Card Platform (SCPG)	The SCPG group contains the security requirements for the smart card platform, that is, operating system and chip that the Java Card System is implemented upon.
Card Manager (CMGRG)	The CMGRG group contains the security requirements for the card manager.
OS Update	The OS Update group contains the security requirements for the OS update.

Table 28:Requirement Groups

The SFRs refer to all potentially applicable subjects, objects, information, operations and security attributes.

Subjects are active components of the TOE that (essentially) act on the behalf of users. The users of the TOE include people or institutions (like the applet developer, the card issuer, the verification authority), hardware (like the CAD where the card is inserted or the PCD) and software components (like the

application packages installed on the card). Some of the users may just be aliases for other users. For instance, the verification authority in charge of the bytecode verification of the applications may be just an alias for the card issuer.

Subjects (prefixed with an "S") are described in the following table:

Subject	Description
S.ADEL	The applet deletion manager which also acts on behalf of the card issuer. It may be an applet ([JCRE] , §11), but its role asks anyway for a specific treatment from the security viewpoint.
S.APPLET	Any applet instances.
S.BCV	The bytecode verifier (BCV), which acts on behalf of the verification authority who is in charge of the bytecode verification of the CAP files.
S.CAD	The CAD represents off-card entity that communicates with the S.INSTALLER. If the TOE provides JCRMI functionality, CAD can request RMI services by issuing commands to the card.
S.INSTALLER	The installer is the on-card entity which acts on behalf of the card issuer. This subject is involved in the loading of CAP files and installation of applets.
S.JCRE	The runtime environment under which Java programs in a smart card are executed.
S.JCVM	The bytecode interpreter that enforces the firewall at runtime.
S.LOCAL	Operand stack of a JCVM frame, or local variable of a JCVM frame containing an object or an array of references.
S.MEMBER	Any object's field, static field or array position.
S.CAP_FILE	A CAP file may contain multiple Java language packages. A package is a namespace within the Java programming language that may contain classes and interfaces. A CAP file may contain packages that define either user library, or one or several applets. A CAP file compliant with Java Card Specifications version 3.1 may contain multiple Java language packages. An EXTENDED CAP file as specified in Java Card Specifications version 3.1 may contain only applet packages, only library packages or a combination of library packages. A COMPACT CAP file as specified in Java Card Specifications version 3.1 or CAP files compliant to previous versions of Java Card Specification, MUST contain only a single package representing a library or one or more applets.
S.CM	Card Manager which is a generic term for the card management entities of a GlobalPlatform card as defined in [GP] .
S.OSU	OS Update provides secure functionality to update the TOE operating system with an image created by a trusted off-card entity (S.UpdateImageCreator)
S.UpdateImageCreator	The off-card Update Image Creator ensures that the confidentiality and

	integrity requirements are met.
--	---------------------------------

Table 29:Java Card Subject Descriptions

Objects (prefixed with an "O") are described in the following table:

Object	Description
O.APPLLET	Any installed applet, its code and data.
O.CODE_CAP_FILE	The code of a CAP file, including all linking information. On the Java Card platform, a CAP file is the installation unit.
O.JAVAOBJECT	Java class instance or array. It should be noticed that KEYS, PIN, arrays and applet instances are specific objects in the Java programming language.
O.GP_KEY	Keys are used for token and DAP verification and generation as well as secure channel protocol as defined in [GP] .

Table 30:Object Groups

Information (prefixed with an "I") is described in the following table:

Information	Description
I.APDU	Any APDU sent to or from the card through the communication channel.
I.DATA	JCVM Reference Data: objectref addresses of APDU buffer, JCRE-owned instances of APDU class and byte array for install method.

Table 31:Information Groups

Security attributes linked to these subjects, objects and information are described in the following table with their values:

Security attribute	Description/Value
Active Applets	The set of the active applets' AIDs. An active applet is an applet that is selected on at least one of the logical channels.
Applet Selection Status	"Selected" or "Deselected".
Applet's version number	The version number of an applet indicated in the export file.
Context	Package AID or "Java Card RE".
Currently Active Context	Package AID or "Java Card RE".
Dependent package AID	Allows the retrieval of the package AID and Applet's version number ([JCVM] , §4.5.2).
LC Selection Status	Multiselectable, Non-multiselectable or "None".
LifeTime	CLEAR_ON_DESELECT or PERSISTENT (*).
Owner	The Owner of an object is either the applet instance that created the object or the CAP file (library) where it has been defined (these latter objects can only

	be arrays that initialize static fields of the CAP file). The owner of a remote object is the applet instance that created the object.
Package AID	The AID of each package indicated in the export file.
Registered Applets	The set of AID of the applet instances registered on the card.
Resident packages	The set of AIDs of the packages already loaded on the card.
Selected Applet Context	Package AID or "None".
Sharing	Standard, SIO, Array View, Java Card RE entry point or global array.
Static References	Static fields of a CAP file may contain references to objects. The Static References attribute record those references.
Security Level	Security level for secure messaging as defined in [GP] .
Privilege	The privilege assigned to on card entity as defined in [GP] .
Life Cycle Status	Life cycle status associated with on card entity as defined in [GP] .
DAP	Data authentication pattern for cap file as defined in [GP] .
MAC	Message authentication code as defined in [GP] .
GP key	Keys as defined in [GP] .
DM token	Delegated Management token as defined in [GP] .
Key type	Key type identifier as defined in [GP] .
Key checksum	Checksum used to verify the integrity of key values as defined in [GP] .
Secure Channel Protocol	Secure Channel Protocol version used.
Session Key	Secure Channel's session key.
Sequence Counter	Secure Channel Session's Sequence Counter.
ICV	Secure Channel Session's ICV.

Table 32:Security attribute description

(*) Transient objects of type CLEAR_ON_RESET behave like persistent objects in that they can be accessed only when the Currently Active Context is the object's context.

Operations (prefixed with "OP") are described in the following table. Each operation has parameters given between brackets, among which there is the "accessed object", the first one, when applicable. Parameters may be seen as security attributes that are under the control of the subject performing the operation.

Operation	Description
OP.ARRAY_ACCESS(O.JAVAOBJECT, field)	Read/Write an array component.
OP.ARRAY_LENGTH (O.JAVAOBJECT, field)	Get length of an array component.
OP.ARRAY_T_ALOAD(O.JAVAOBJECT, field)	Read from an array component
OP.ARRAY_T_ASTORE(O.JAVAOBJECT, field)	Write to an array component
OP.ARRAY_AASTORE(O.JAVAOBJECT, field)	Store into reference array component

OP.CREATE(Sharing, LifeTime) (*)	Creation of an object (new, makeTransient or createArrayView call).
OP.DELETE_APPLET(O.APPLET,...)	Delete an installed applet and its objects, either logically or physically.
OP.DELETE_CAP_FILE(O.CODE_CAP_FILE,...)	Delete a CAP file, either logically or physically.
OP.DELETE_CAP_FILE_APPLET(O.CODE_CAP_FILE,...)	Delete a CAP file and its installed applets, either logically or physically.
OP.INSTANCE_FIELD(O.JAVAOBJECT, field)	Read/Write a field of an instance of a class in the Java programming language.
OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1,...)	Invoke a virtual method (either on a class instance or an array object).
OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1,...)	Invoke an interface method.
OP.JAVA(...)	Any access in the sense of [JCRE], §6.2.8. It stands for one of the operations OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW, OP.TYPE_ACCESS. OP.ARRAY_LENGTH
OP.PUT(S1,S2,I)	Transfer a piece of information I from S1 to S2.
OP.THROW(O.JAVAOBJECT)	Throwing of an object (athrow, see [JCRE], §6.2.8.7).
OP.TYPE_ACCESS(O.JAVAOBJECT, class)	Invoke checkcast or instanceof on an object in order to access to classes (standard or shareable interfaces objects).
OP.CCM(...)	The Card Content Management as defined in [GP]. It stands for one of the operations OP.LOAD, OP.INSTALL, OP.DELETE, OP.PUT_KEY.
OP.TRIGGER_OSU	APDU Command that

	initializes the OS Update procedure
--	-------------------------------------

Table 33:Operation Description

(*) For this operation, there is no accessed object. This rule enforces that shareable transient objects are not allowed. For instance, during the creation of an object, the JavaCardClass attribute's value is chosen by the creator.

The following table lists all the security functional requirements of the TOE from [\[PP JCS\]](#). They apply to this Security Target with some modification (i.e. operation performed or replaced).

The notation used is detailed below:

- Equivalent (E): The element in the ST is the same as in [\[PP JCS\]](#).
- Addition (A): The element is newly defined in the ST; it is not present in [\[PP JCS\]](#) and does not affect it.
- Refinement(R): The element in the ST refines the corresponding [\[PP JCS\]](#) element.
- X The element is present in [\[PP JCS\]](#).

Security Functional requirements	[PP JCS]	Security Target
FDP_ACC.2/FIREWALL	X	(E)
FDP_ACF.1/FIREWALL	X	(E)
FDP_IFC.1/JCVM	X	(E)
FDP_IFF.1/JCVM	X	(E)
FDP_RIP.1/OBJECTS	X	(E)
FMT_MSA.1/JCRE	X	(E)
FMT_MSA.1/JCVM	X	(E)
FMT_MSA.2/FIREWALL_JCVM	X	(E)
FMT_MSA.3/FIREWALL	X	(E)
FMT_MSA.3/JCVM	X	(E)
FMT_SMF.1/JCS	X	(R) Refined with iteration.
FMT_SMR.1/JCS	X	(R) Refined with iteration.
FCS_CKM.1	X	(E)
FCS_CKM.6	- <i>Note: FCS_CKM.4 has been deprecated in the latest version of CC:2022, use FSC_CKM.6 instead of FSC_CKM.4</i>	(E)
FCS_COP.1	X	(E)
FCS_RNG.1	X	(E)
FDP_RIP.1/ABORT	X	(E)
FDP_RIP.1/APDU	X	(E)

FDP_RIP.1/bArray	X	(E)
FDP_RIP.1/GlobalArray	X	(E)
FDP_RIP.1/KEYS	X	(E)
FDP_RIP.1/TRANSIENT	X	(E)
FDP_ROL.1/FIREWALL	X	(E)
FAU_ARP.1	X	(E)
FDP_SDI.2/DATA	X	(E)
FPR_UNO.1	X	(E)
FPT_FLS.1/JCS	X	(R) Refined with iteration.
FPT_TDC.1	X	(E)
FIA_ATD.1/AID	X	(E)
FIA_UID.2/AID	X	(E)
FIA_USB.1/AID	X	(E)
FMT_MTD.1/JCRE	X	(E)
FMT_MTD.3/JCRE	X	(E)
FDP_ITC.2/Installer	X	(E)
FMT_SMR.1/Installer	X	(E)
FPT_FLS.1/Installer	X	(E)
FPT_RCV.3/Installer	X	(E)
FDP_ACC.2/ADEL	X	(E)
FDP_ACF.1/ADEL	X	(E)
FDP_RIP.1/ADEL	X	(E)
FMT_MSA.1/ADEL	X	(E)
FMT_MSA.3/ADEL	X	(E)
FMT_SMF.1/ADEL	X	(E)
FMT_SMR.1/ADEL	X	(E)
FPT_FLS.1/ADEL	X	(E)
FDP_RIP.1/ODEL	X	(E)
FPT_FLS.1/ODEL	X	(E)
FCO_NRO.2/CM	X	(E)
FDP_IFC.2/CM	X	(E)
FDP_IFF.1/CM	X	(E)
FDP_UIT.1/CM	X	(E)
FIA_UID.1/CM	X	(E)
FMT_MSA.1/CM	X	(E)
FMT_MSA.3/CM	X	(E)
FMT_SMF.1/CM	X	(E)
FMT_SMR.1/CM	X	(E)
FTP_ITC.1/CM	X	(E)
FDP_ACC.1/CMGR	-	(A)
FDP_ACF.1/CMGR	-	(A)

FMT_MSA.1/CMGR	-	(A)
FMT_MSA.3/CMGR	-	(A)
FPT_RCV.4/SCP	-	(A)
FPT_TST.1/SCP	-	(A)
FPT_PHP.3/SCP	-	(A)
FDP_IFC.2/OSU	-	(A)
FDP_IFF.1/OSU	-	(A)
FMT_MSA.3 /OSU	-	(A)
FMT_MSA.1/OSU	-	(A)
FMT_SMR.1/OSU	-	(A)
FMT_SMF.1/OSU	-	(A)
FIA_UID.1/OSU	-	(A)
FIA_UAU.1 /OSU	-	(A)
FIA_UAU.4 /OSU	-	(A)
FPT_FLS.1/OSU	-	(A)
FCS_CKM.3/TDES	-	(A)
FCS_CKM.3/AES	-	(A)
FCS_CKM.3/ECC	-	(A)
FCS_CKM.2/TDES	-	(A)
FCS_CKM.2/AES	-	(A)

Table 34:SFRs nation map

7.1.1.1 COREG_LC SFRs

This group is focused on the main security policy of the Java Card System, known as the firewall.

7.1.1.1.1 Firewall Policy

FDP_ACC.2/FIREWALL Complete access control

FDP_ACC.2.1/FIREWALL The TSF shall enforce the [assignment :FIREWALL access control SFP] on [assignment : S.CAP_FILE, S.JCRE, S.JCVM, O.JAVAOBJECT] and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- OP.CREATE,
- OP.INVK_INTERFACE,
- OP.INVK_VIRTUAL,
- OP.JAVA,
- OP.THROW,

- OP.TYPE_ACCESS.
- OP.ARRAY_LENGTH
- OP.ARRAY_T_ALOAD
- OP.ARRAY_T_ASTORE
- OP.ARRAY_AASTORE

FDP_ACC.2.2/FIREWALL The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

Application Note:

It should be noticed that accessing array's components of a static array, and more generally fields and methods of static objects, is an access to the corresponding O.JAVAOBJECT.

FDP_ACE1/FIREWALL Security attribute-based access control
--

FDP_ACE1.1/FIREWALL The TSF shall enforce the [assignment: FIREWALL access control SFP] to objects based on the following [assignment :

Subject/Object	Security attributes
S.CAP_FILE	LC Selection Status
S.JCVM	Active Applets, Currently Active Context
S.JCRE	Selected Applet Context
O.JAVAOBJECT	Sharing, Context, LifeTime

Table 35:The FIREWALL access control SFP to objects

].

FDP_ACE1.2/FIREWALL The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:[assignment :

- R.JAVA.1 ([JCRE], §6.2.8): S.CAP_FILE may freely perform, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW or OP.TYPE_ACCESS upon any O.JAVAOBJECT whose Sharing attribute has value "JCRE entry point" or "global array".
- R.JAVA.2 ([JCRE], §6.2.8): S.CAP_FILE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE or OP.THROW upon any O.JAVAOBJECT whose Sharing attribute has value "Standard" and whose Lifetime attribute has value "PERSISTENT" only if O.JAVAOBJECT's Context attribute has the same value as the active context.
- R.JAVA.3 ([JCRE], §6.2.8.10): S. CAP_FILE may perform OP.TYPE_ACCESS upon an O.JAVAOBJECT with Context attribute different from the currently active context, whose Sharing attribute has value "SIO" only if O.JAVAOBJECT is being cast into (checkcast) or is being verified as being an instance of (*instanceof*) an interface that extends the Shareable interface.

- R.JAVA.4 ([JCRE], §6.2.8.6): S.CAP_FILE may perform OP.INVK_INTERFACE upon an O.JAVAOBJECT with Context attribute different from the currently active context, whose Sharing attribute has the value "SIO", and whose Context attribute has the value "Package AID", only if the invoked interface method extends the Shareable interface and one of the following conditions applies:
 - a) The value of the attribute Selection Status of the package whose AID is "Package AID" is "Multiselectable",
 - b) The value of the attribute Selection Status of the package whose AID is "Package AID" is "Non-multiselectable", and either "Package AID" is the value of the currently selected applet or otherwise "Package AID" does not occur in the attribute Active Applets.
 - R.JAVA.5: S.CAP_FILE may perform OP.CREATE upon O.JAVAOBJECT only if the value of the Sharing parameter is "Standard" or "SIO".
 - R.JAVA.6 ([JCRE], §6.2.8): S.CAP_FILE may freely perform OP.ARRAY_ACCESS or OP.ARRAY_LENGTH upon any O.JAVAOBJECT whose Sharing attribute has value "global array".
-].

FDP_ACF.1.3/FIREWALL The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[assignment:**

- 1) The subject S.JCRE can freely perform OP.JAVA("") and OP.CREATE, with the exception given in FDP_ACF.1.4/FIREWALL, provided it is the Currently Active Context.
- 2) The only means that the subject S.JCVM shall provide for an application to execute native code is the invocation of a Java Card API method (through OP.INVK_INTERFACE or OP.INVK_VIRTUAL).

].

FDP_ACF.1.4/FIREWALL The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment:**

- 1) Any subject with OP.JAVA upon an O.JAVAOBJECT whose LifeTime attribute has value "CLEAR_ON_DESELECT" if O.JAVAOBJECT's Context attribute is not the same as the Selected Applet Context.
- 2) Any subject attempting to create an object by the means of OP.CREATE and a "CLEAR_ON_DESELECT" LifeTime parameter if the active context is not the same as the Selected Applet Context.
- 3) S.CAP_FILE performing OP.ARRAY_AASTORE of the reference of an O.JAVAOBJECT whose sharing attribute has value "global array" or "Temporary".
- 4) S.CAP_FILE performing OP.PUTFIELD or OP.PUTSTATIC of the reference of an

O.JAVAOBJECT whose sharing attribute has value “global array” or “Temporary”

- 5) R.JAVA.7 ([JCRE], §6.2.8.2): S.CAP_FILE performing OP.ARRAY_T_ASTORE into an array view without ATTR_WRITABLE_VIEW access attribute.
 - 6) R.JAVA.8 ([JCRE], §6.2.8.2): S.CAP_FILE performing OP.ARRAY_T_ALOAD into an array view without ATTR_READABLE_VIEW access attribute.
- l.

Application Note: FDP_ACF.1.4/FIREWALL:

- The deletion of applets may render some O.JAVAOBJECT inaccessible, and the Java Card RE may be in charge of this aspect. This can be done, for instance, by ensuring that references to objects belonging to a deleted application are considered as a null reference. Such a mechanism is implementation-dependent.

In the case of an array type, fields are components of the array ([JVM], §2.14, §2.7.7), as well as the length; the only methods of an array object are those inherited from the Object class.

The Sharing attribute defines five categories of objects:

- Standard ones, whose both fields and methods are under the firewall policy,
- Shareable interface Objects (SIO), which provide a secure mechanism for inter-applet communication,
- JCRE entry points (Temporary or Permanent), who have freely accessible methods but protected fields,
- Global arrays, having both unprotected fields (including components; refer to JavaCardClass discussion above) and methods.
- Array Views, having fields/elements access controlled by access control attributes, ATTR_READABLE_VIEW and ATTR_WRITABLE_VIEW and methods.

When a new object is created, it is associated with the Currently Active Context. But the object is owned by the applet instance within the Currently Active Context when the object is instantiated ([JCRE], §6.1.3). An object is owned by an applet instance, by the JCRE or by the library where it has been defined (these latter objects can only be arrays that initialize static fields of CAP files).

([JCRE], Glossary) Selected Applet Context. The Java Card RE keeps track of the currently selected Java Card applet. Upon receiving a SELECT command with this applet's AID, the Java Card RE makes this applet the Selected Applet Context. The Java Card RE sends all APDU commands to the Selected Applet Context.

While the expression "Selected Applet Context" refers to a specific installed applet, the relevant aspect to the policy is the context (Package AID) of the selected applet. In this policy, the "Selected Applet Context" is the AID of the selected package.

([\[JCRE\]](#), §6.1.2.1) At any point in time, there is only one active context within the Java Card VM (this is called the Currently Active Context).

It should be noticed that the invocation of static methods (or access to a static field) is not considered by this policy, as there are no firewall rules. They have no effect on the active context as well and the "acting CAP File" is not the one to which the static method belongs to in this case.

It should be noticed that the Java Card platform, version 2.2.x and version 3.x.x Classic Edition, introduces the possibility for an applet instance to be selected on multiple logical channels at the same time, or accepting other applets belonging to the same package being selected simultaneously. These applets are referred to as multiselectable applets. Applets that belong to a same package are either all multiselectable or not ([\[JCVM\]](#), §2.2.5). Therefore, the selection mode can be regarded as an attribute of packages. No selection mode is defined for a library package.

An applet instance will be considered an active applet instance if it is currently selected in at least one logical channel. An applet instance is the currently selected applet instance only if it is processing the current command. There can only be one currently selected applet instance at a given time ([\[JCRE\]](#), §4).

FDP_IFC.1/JCVM Subset information flow control

FDP_IFC.1.1/JCVM The TSF shall enforce the [assignment: JCVM information flow control SFP] on S.JCVM, S.LOCAL, S.MEMBER, I.DATA and OP.PUT(S1, S2, I)].

Application Note:

It should be noticed that references of temporary Java Card RE entry points, which cannot be stored in class variables, instance variables or array components, are transferred from the internal memory of the Java Card RE (TSF data) to some stack through specific APIs (Java Card RE owned exceptions) or Java Card RE invoked methods (such as the process(APDU apdu)); these are causes of OP.PUT(S1,S2,I) operations as well.

FDP_IFE.1/JCVM Simple security attributes

FDP_IFE.1.1/JCVM The TSF shall enforce the [assignment: JCVM information flow control SFP] based on the following types of subject and information security attributes [assignment:

Subjects	Security attributes
S.JCVM	Currently Active Context

].

FDP_IFE.1.2/JCVM The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment:

- An operation OP.PUT(S1, S.MEMBER, I.DATA) is allowed if and only if the Currently Active Context is "Java Card RE";

- other OP.PUT operations are allowed regardless of the Currently Active Context's value.
-].

FDP_IFF.1.3/JCVM The TSF shall enforce [assignment: no additional information flow control SFP rules].

FDP_IFF.1.4/JCVM The TSF shall explicitly authorize an information flow based on the following rules:[assignment: none].

FDP_IFF.1.5/JCVM The TSF shall explicitly deny an information flow based on the following rules: [assignment: none].

Application Note:

The storage of temporary Java Card RE-owned objects references is runtime-enforced ([JCRE], §6.2.8.1-3).

It should be noticed that this policy essentially applies to the execution of bytecode. Native methods, the Java Card RE itself and possibly some API methods can be granted specific rights or limitations through the FDP_IFF.1.3/JCVM to FDP_IFF.1.5/JCVM elements. The way the Java Card virtual machine manages the transfer of values on the stack and local variables (returned values, uncaught exceptions) from and to internal registers is implementation-dependent. For instance, a returned reference, depending on the implementation of the stack frame, may transit through an internal register prior to being pushed on the stack of the invoker. The returned bytecode would cause more than one OP.PUT operation under this scheme.

FDP_RIP.1/OBJECTS Subset residual information protection

FDP_RIP.1.1/OBJECTS The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: allocation of the resource to] the following objects: [assignment: class instances and arrays].

Application Note:

The semantics of the Java programming language requires for any object field and array position to be initialized with default values when the resource is allocated [JVM], §2.5.1.

FMT_MSA.1/JCRE Management of security attributes

FMT_MSA.1.1/JCRE The TSF shall enforce the [assignment: FIREWALL access control SFP] to restrict the ability to [selection: modify] the security attributes [assignment: Selected Applet Context] to [assignment: S.JCRE].

Application Note:

The modification of the Selected Applet Context should be performed in accordance with the rules given in [JCRE], §4 and [JCVM], §3.4.

FMT_MSA.1/JCVM Management of security attributes

FMT_MSA.1.1/JCVM The TSF shall enforce the [assignment: FIREWALL access control SFP and the JCVM information flow control SFP] to restrict the ability to [selection: modify] the security attributes [assignment: Currently Active Context and Active Applets] to [assignment: S.JCVM].

Application Note:

The modification of the Currently Active Context should be performed in accordance with the rules given in [JCRE], §4 and [JCVM], §3.4.

FMT_MSA.2/FIREWALL_JCVM Secure security attributes

FMT_MSA.2.1/FIREWALL_JCVM The TSF shall ensure that only secure values are accepted for [assignment: all the security attributes of subjects and objects defined in the FIREWALL access control SFP and the JCVM information flow control SFP].

Application Note:

The following rules are given as examples only. For instance, the last two rules are motivated by the fact that the Java Card API defines only transient arrays factory methods. Future versions may allow the creation of transient objects belonging to arbitrary classes; such evolution will naturally change the range of "secure values" for this component.

- The Context attribute of an O.JAVAOBJECT must correspond to that of an installed applet or be "Java Card RE".
- An O.JAVAOBJECT whose Sharing attribute is a Java Card RE entry point or a global array necessarily has "Java Card RE" as the value for its Context security attribute.
- Any O.JAVAOBJECT whose Sharing attribute value is not "Standard" has a PERSISTENT-LifeTime attribute's value.
- Any O.JAVAOBJECT whose LifeTime attribute value is not PERSISTENT has an array type as JavaCardClass attribute's value.

FMT_MSA.3/FIREWALL Static attribute initialization

FMT_MSA.3.1/FIREWALL The TSF shall enforce the [assignment: FIREWALL access control SFP] to provide [selection: restrictive] default values for security attributes that are used to enforce

the SFP.

FMT_MSA.3.2/FIREWALL[Editorially Refined] The TSF shall not allow **[assignment: any role]** to specify alternative initial values to override the default values when an object or information is created.

Application Note:

FMT_MSA.3.1/FIREWALL

- Objects' security attributes of the access control policy are created and initialized at the creation of the object or the subject. Afterwards, these attributes are no longer mutable (FMT_MSA.1/JCRE). At the creation of an object (OP.CREATE), the newly created object, assuming that the FIREWALL access control SFP permits the operation, gets its Lifetime and Sharing attributes from the parameters of the operation; on the contrary, its Context attribute has a default value, which is its creator's Context attribute and AID respectively ([JCRE], §6.1.3). There is one default value for the Selected Applet Context that is the default applet identifier's Context, and one default value for the Currently Active Context that is "Java Card RE".
- The knowledge of which reference corresponds to a temporary entry point object or a global array and which does not is solely available to the Java Card RE (and the Java Card virtual machine).

FMT_MSA.3.2/FIREWALL

- The intent is that none of the identified roles has privileges with regard to the default values of the security attributes. It should be noticed that creation of objects is an operation controlled by the FIREWALL access control SFP. The operation shall fail anyway if the created object would have had security attributes whose value violates FMT_MSA.2.1/FIREWALL_JCVM.

FMT_MSA.3/JCVM Static attribute initialization

FMT_MSA.3.1/JCVM The TSF shall enforce the **[assignment: JCVM information flow control SFP]** to provide **[selection: restrictive]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/JCVM [Editorially Refined] The TSF shall not allow **[assignment: any role]** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/JCS Specification of Management Functions

FMT_SMF.1.1/JCS The TSF shall be capable of performing the following management functions:
[assignment:

- **modify the Currently Active Context, the Selected Applet Context and the Active Applets.**

].

FMT_SMR.1/JCS Security roles

FMT_SMR.1.1/JCS The TSF shall maintain the roles [assignment:

- Java Card RE (JCRE),
- Java Card VM (JCVM).

].

FMT_SMR.1.2/JCS The TSF shall be able to associate users with roles.

7.1.1.1.2 Application Programming Interface

The following SFRs are related to the Java Card API.

It should be noticed that the execution of the additional native code is not within the TSF. Nevertheless, access to API native methods from the Java Card System is controlled by TSF because there is no difference between native and interpreted methods in their interface or invocation mechanism.

FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

Iteration	Algorithm	Key size	Standards
ECC	ECC	ansix9p224r1, ansix9p256r1, ansix9p384r1, ansix9p521r1, brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1 and brainpoolP512r1 curves	ANSI X9.62-2005, RFC 5639
TDES	GP SCP02 session key	112, 168 bit(for SCP02)	[GP] (for SCP02)
AES	GP SCP03 session key	128,192, 256 bit(for SCP03)	[GP] (for SCP03)

Application Note:

The keys can be generated and diversified in accordance with [JCAPI] specification in classes KeyBuilder, KeyPair (at least Session key generation) and RandomData.

FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: “Distribution method” column of the table below] that meets the following: [assignment: [JCAPI](#)]

Iteration	Distribution method	Standard
TDES	JC API setKey()	JCAPI
AES	JC API setKey()	JCAPI

FCS_CKM.3 Cryptographic key access

FCS_CKM.3.1 The TSF shall perform [key access] in accordance with a specified cryptographic key access method [assignment: “Key access method” column of the table below] that meets the following: [assignment: [JCAPI](#)]

Iteration	Key access method	Standard
TDES	JC API getKey()	JCAPI
AES	JC API getKey()	JCAPI
ECC	JC API: getA(), getB(), getG(), getR(), getK(),getS(),getW()	JCAPI

FCS_CKM.6 Cryptographic key destruction

FCS_CKM.6.1 The TSF shall destroy [assignment:

- D.APP_KEYs
- D.OS-UPDATE_KEY(S)

] when [selection: no longer needed]

FCS_CKM.6.2 The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [assignment: clean the buffer with random bytes] that meets the following: [assignment: none].

Application Note:

- The keys are reset as specified in [JCAPI] Key class, with the method clearKey(). Any access to a cleared key for ciphering or signing shall throw an exception.

FCS_COP.1 Cryptographic operation

FCS_COP.1.1 The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

Iteration	Operation	Algorithm	Key Size	Standards
TDES	Encryption and Decryption	ALG_DES_CBC_ISO9797_M1 ALG_DES_CBC_ISO9797_M2 ALG_DES_CBC_NOPAD ALG_DES_CBC_PKCS5 ALG_DES_ECB_ISO9797_M1 ALG_DES_ECB_ISO9797_M2 ALG_DES_ECB_NOPAD ALG_DES_ECB_PKCS5	LENGTH_DES3_2KEY,	NIST-SP800-67 ISO9797-1 DES NOPAD DES PKCS#5 DES 9797 M1 M2
TDES_MA C	Generation and Verification	ALG_DES_MAC4_ISO9797_1_M1_ALG3 ALG_DES_MAC4_ISO9797_1_M2_ALG3 ALG_DES_MAC4_ISO9797_M1 ALG_DES_MAC4_ISO9797_M2 ALG_DES_MAC8_ISO9797_1_M1_ALG3 ALG_DES_MAC8_ISO9797_1_M2_ALG3 ALG_DES_MAC8_ISO9797_M1 ALG_DES_MAC8_ISO9797_M2 ALG_DES_MAC4_NOPAD ALG_DES_MAC8_NOPAD ALG_DES_MAC4_PKCS5 ALG_DES_MAC8_PKCS5	LENGTH_DES3_2KEY,	[JCAPI]
ECDSA	Signature and Verification	ALG_ECDSA_SHA_256	256 bits	[JCAPI]
AES_CIPHER	Encryption and Decryption	ALG_AES_BLOCK_128_CBC_NOPAD ALG_AES_BLOCK_128_ECB_NOPAD ALG_AES_CBC_ISO9797_M1 ALG_AES_CBC_ISO9797_M2 ALG_AES_CBC_PKCS5 ALG_AES_BLOCK_128_ECB_NOPAD ALG_AES_ECB_ISO9797_M1 ALG_AES_ECB_ISO9797_M2 ALG_AES_ECB_PKCS5	128, 192, 256 bits	[JCAPI]
AES_MAC	Signature and Verification	ALG_AES_CMAC_128 ALG_AES_MAC_128_NOPAD ALG_AES_MAC_192_NOPAD ALG_AES_MAC_256_NOPAD	128, 192, 256 bits	[JCAPI]

ECDH	Key Agreement	ALG_EC_SVDP_DH ALG_EC_SVDP_DHC ALG_EC_SVDP_DH_PLAIN ALG_EC_SVDP_DHC_PLAIN ALG_EC_SVDP_DH_PLAIN_XY	-	[JCAPI]
SHA	Secure Hash Computation	SHA-256	256 bits	[JCAPI]
HAMC	Signature	HAMC	256 bits	FIPS 198 The Keyed-Hash Message Authentication Code(HMAC)

FDP_RIP.1/ABORT Subset residual information protection

FDP_RIP.1.1/ABORT The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: deallocation of the resource from]** the following objects:
[assignment: any reference to an object instance created during an aborted transaction].

Application Note:

The events that provoke the de-allocation of a transient object are described in [JCRE], §5.1.

FDP_RIP.1/APDU Subset residual information protection

FDP_RIP.1.1/APDU The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[assignment: allocation of the resource to]** the following objects:
[assignment: the APDU buffer].

Application Note:

The allocation of a resource to the APDU buffer is typically performed as the result of a call to the process() method of an applet.

FDP_RIP.1/GlobalArray Subset residual information protection

FDP_RIP.1.1/GlobalArray [Refined] The TSF shall ensure that any previous information content of a resource is made unavailable upon **[selection: deallocation of the resource from]** the applet as a result of returning from the process method to the following objects: **[assignment: a user Global Array].**

Application Note:

An array resource is allocated when a call to the API method JCSysm.makeGlobalArray is performed. The Global Array is created as a transient JCRE Entry Point Object ensuring that reference to it cannot be retained by any application. On return from the method which called

JCSystem.makeGlobalArray, the array is no longer available to any applet and is deleted and the memory in use by the array is cleared and reclaimed in the next object deletion cycle.

FDP_RIP.1/bArray Subset residual information protection

FDP_RIP.1.1/bArray The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: deallocation of the resource from]** the following objects: **[assignment: the bArray object]**.

Application Note:

A resource is allocated to the bArray object when a call to an applet's install() method is performed. There is no conflict with FDP_ROL.1 here because of the bounds on the rollback mechanism (FDP_ROL.1.2/FIREWALL): the scope of the rollback does not extend outside the execution of the install() method, and the de-allocation occurs precisely right after the return of it.

FDP_RIP.1/KEYS Subset residual information protection

FDP_RIP.1.1/KEYS The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: deallocation of the resource from]** the following objects: **[assignment: the cryptographic buffer (D.CRYPTO)]**.

Application Note:

- The javacard.security & javacardx.crypto packages do provide secure interfaces to the cryptographic buffer in a transparent way. See javacard.security.KeyBuilder and Key interface of [JCAP1](#).

FDP_RIP.1/TRANSIENT Subset residual information protection

FDP_RIP.1.1/TRANSIENT The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[assignment: deallocation of the resource from]** the following objects: **[assignment: any transient object]**.

Application Note:

- The events that provoke the de-allocation of any transient object are described in [JCRE], §5.1.
- The clearing of CLEAR_ON_DESELECT objects is not necessarily performed when the owner of the objects is deselected. In the presence of multiselectable applet instances, CLEAR_ON_DESELECT memory segments may be attached to applets that are active in different logical channels. Multiselectable applet instances within a same package must share the transient memory segment if they are concurrently active ([JCRE], §4.3).

FDP_ROL.1/FIREWALL Basic rollback

FDP_ROL.1.1/FIREWALL The TSF shall enforce **[assignment: the FIREWALL access control SFP and the JCVM information flow control SFP]** to permit the rollback of the **[assignment: operations OP.JAVA and OP.CREATE]** on the **[assignment: object O.JAVAOBJECT]**.

FDP_ROL.1.2/FIREWALL The TSF shall permit operations to be rolled back within the [assignment: scope of a select(), deselect(), process(), install() or uninstall() call, notwithstanding the restrictions given in [JCRE], §7.7, within the bounds of the Commit Capacity ([JCRE], §7.8), and those described in [JCAPI]].

Application Note:

Transactions are a service offered by the APIs to applets. It is also used by some APIs to guarantee the atomicity of some operation. This mechanism is either implemented in Java Card platform or relies on the transaction mechanism offered by the underlying platform. Some operations of the API are not conditionally updated, as documented in [JCAPI] (see for instance, PIN-blocking, PIN-checking, update of Transient objects).

7.1.1.1.3 Card Security Management

FAU_ARP.1 Security alarms

FAU_ARP.1.1 The TSF shall take [assignment: one of the following actions:

- throw an exception,
- lock the card session,
- reinitialize the Java Card System and its data.

]upon detection of a potential security violation.

Refinement:

The "potential security violation" stands for one of the following events:

- CAP file inconsistency,
- typing error in the operands of a bytecode,
- applet life cycle inconsistency,
- card manager life cycle inconsistency,
- card tearing (unexpected removal of the Card out of the CAD) and power failure,
- abort of a transaction in an unexpected context, (see abortTransaction(), [JCAPI] and ([JCRE], §7.6.2)
- violation of the Firewall or JCVM SFPs,
- unavailability of resources,
- array overflow.

Application Note:

- The developer shall provide the exhaustive list of actual potential security violations the TOE reacts to. For instance, other runtime errors related to applet's failure like uncaught exceptions.

- The bytecode verification defines a large set of rules used to detect a "potential security violation". The actual monitoring of these "events" within the TOE only makes sense when the bytecode verification is performed on-card.
- Depending on the context of use and the required security level, there are cases where the card manager and the TOE must work in cooperation to detect and appropriately react in case of potential security violation. This behavior must be described in this component. It shall detail the nature of the feedback information provided to the card manager (like the identity of the offending application) and the conditions under which the feedback will occur (any occurrence of the `java.lang.SecurityException` exception).
- The "locking of the card session" may not appear in the policy of the card manager. Such measure should only be taken in case of severe violation detection; the same holds for the re-initialization of the Java Card System. Moreover, the locking should occur when "clean" re-initialization seems to be impossible.
- The locking may be implemented at the level of the Java Card System as a denial of service (through some systematic "fatal error" message or return value) that lasts up to the next "RESET" event, without affecting other components of the card (such as the card manager). Finally, because the installation of applets is a sensitive process, security alerts in this case should also be carefully considered herein.

FDP_SDL2/DATA Stored data integrity monitoring and action
--

FDP_SDL2.1/DATA The TSF shall monitor user data stored in containers controlled by the TSF for **[assignment: integrity errors]** on all objects, based on the following attributes: **[assignment: integrity-sensitive data]**.

FDP_SDL2.2/DATA Upon detection of a data integrity error, the TSF shall **[assignment: throw an exception and reset card session]**.

Application Note:

- Although no such requirement is mandatory in the Java Card specification, at least an exception shall be raised upon integrity errors detection on cryptographic keys, PIN values and their associated security attributes. Even if all the objects cannot be monitored, cryptographic keys and PIN objects shall be considered with particular attention by ST authors as they play a key role in the overall security.
- It is also recommended to monitor integrity errors in the code of the native applications and Java Card applets.

For integrity sensitive application, their data shall be monitored (D.APP_I_DATA): applications may need to protect information against unexpected modifications, and explicitly control whether a piece of information has been changed between two accesses. For example, maintaining the integrity of an electronic purse's balance is extremely important because this value represents real money. Its

modification must be controlled, for illegal ones would denote an important failure of the payment system.

- A dedicated library could be implemented and made available to developers to achieve better security for specific objects, following the same pattern that already exists in cryptographic APIs, for instance.

FPR_UNO.1 Unobservability

FPR_UNO.1.1 The TSF shall ensure that **[assignment: all users]** are unable to observe the operation **[assignment: cryptographic operations / comparisons operations on Key values / PIN values]** by **[assignment: S.JCRE, S.Applet]**.

Application Note:

The non-observability of operations on sensitive information such as keys appears as impossible to circumvent in the smart card world. The precise list of operations and objects is left unspecified, but should at least concern secret keys and PIN values when they exist on the card, as well as the cryptographic operations and comparisons performed on them.

FPT_FLS.1/JCS Failure with preservation of secure state

FPT_FLS.1.1/JCS The TSF shall preserve a secure state when the following types of failures occur: **those associated to the potential security violations described in FAU_ARP.1.**

Application Note:

The Java Card RE Context is the Current context when the Java Card VM begins running after a card reset ([JCRE], §6.2.3) or after a proximity card (PICC) activation sequence ([JCRE]). Behavior of the TOE on power loss and reset is described in [JCRE], §3.6 and §7.1. Behavior of the TOE on RF signal loss is described in [JCRE], §3.6.1.

FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret **[assignment: the CAP files, the bytecode and its data arguments]** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **[assignment:**

- the rules defined in [JCVM] specification,
- the API tokens defined in the export files of reference implementation,
- The rules defined in [GP] specification

].

when interpreting the TSF data from another trusted IT product.

Application Note:

Concerning the interpretation of data between the TOE and the underlying Java Card platform, it is assumed that the TOE is developed consistently with the SCP functions, including memory management, I/O functions and cryptographic functions.

7.1.1.1.4 AID Management

FIA_ATD.1/AID User attribute definition

FIA_ATD.1.1/AID The TSF shall maintain the following list of security attributes belonging to individual users: **[assignment:**

- Package AID,
- Applet's version number,
- Registered applet AID,
- Applet Selection Status.

].

Refinement:

"Individual users" stand for applets.

FIA_UID.2/AID User identification before any action

FIA_UID.2.1/AID The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note:

- By users here it must be understood the ones associated to the packages (or applets) that act as subjects of policies. In the Java Card System, every action is always performed by an identified user interpreted here as the currently selected applet or the package that is the subject's owner. Means of identification are provided during the loading procedure of the CAP file and the registration of applet instances.
- The role Java Card RE defined in FMT_SMR.1/JCS is attached to an IT security function rather than to a "user" of the CC terminology. The Java Card RE does not "identify" itself to the TOE, but it is part of it.

FIA_USB.1/AID User-subject binding

FIA_USB.1.1/AID The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **[assignment: Package AID]**.

FIA_USB.1.2/AID The TSF shall enforce the following rules on the initial association of user security

attributes with subjects acting on the behalf of users: **[assignment: rules for the initial selection of applet defined in [JCRE], §4]**.

FIA_USB.1.3/AID The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[assignment: rules for the applet selection defined in [JCRE], §4]**.

Application Note:

The user is the applet and the subject is the S.CAP_FILE. The subject security attribute "Context" shall hold the user security attribute "Package AID".

FMT_MTD.1/JCRE Management of TSF data

FMT_MTD.1.1/JCRE The TSF shall restrict the ability to **[selection: modify]** the **[assignment: list of registered applets' AIDs]** to **[assignment: the JCRE]**.

Application Note:

- The installer and the Java Card RE manage other TSF data such as the applet life cycle or packages, but this management is implementation specific. Objects in the Java programming language may also try to query AIDs of installed applets through the lookupAID(...) API method.
- The installer, applet deletion manager or even the card manager may be granted the right to modify the list of registered applets' AIDs in specific implementations (possibly needed for installation and deletion; see #.DELETION and #.INSTALL).

FMT_MTD.3/JCRE Secure TSF data

FMT_MTD.3.1/JCRE The TSF shall ensure that only secure values are accepted for **[assignment: the registered applets' AIDs]**.

7.1.1.2 INSTG SFRs

This group consists of the SFRs related to the installation of the applets, which addresses security aspects outside the runtime. The installation of applets is a critical phase, which lies partially out of the boundaries of the firewall, and therefore requires specific treatment. In this Security Target, loading a CAP file or installing an applet modeled as importation of user data (that is, user application's data) with its security attributes (such as the parameters of the applet used in the firewall rules).

FDP_ITC.2/Installer Import of user data with security attributes

FDP_ITC.2.1/Installer The TSF shall enforce **[assignment: the CAP FILE LOADING information flow control SFP]** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/Installer The TSF shall use the security attributes associated with the imported user

data.

FDP_ITC.2.3/Installer The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/Installer The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/Installer The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:**[assignment:**

CAP file loading is allowed only if, for each dependent package, its AID attribute is equal to a resident package AID attribute, the major version attribute associated to the dependent package file is equal to the major version attribute of the resident package and the minor version attribute is equal to or less than the minor version attribute associated to the resident package ([JCVM], §4.5.2)

].

Application Note:

FDP_ITC.2.1/Installer:

- The most common importation of user data is CAP file loading and applet installation on the behalf of the installer. Security attributes consist of the shareable flag of the class component, AID and version numbers of the package or packages contained within the CAP file, maximal operand stack size and number of local variables for each method, and export and import components (accessibility).

FDP_ITC.2.3/Installer:

- The format of the CAP file is precisely defined in [JCVM] specifications; it contains the user data (like applet's code and data) and the security attributes altogether. Therefore, there is no association to be carried out elsewhere.

FDP_ITC.2.4/Installer:

- Each CAP file and all the packages contained within a CAP file contain a Version attribute, which is a pair of major and minor version numbers ([JCVM], §4.5). With the AID, it describes the package defined in the CAP file. When an export file is used during preparation of a CAP file, the version numbers and AIDs of imported packages indicated in the export file are recorded in the CAP files ([JCVM], §4.5.2): the dependent packages' Version and AID attributes allow the retrieval of these identifications. Implementation-dependent checks may occur on a case-by-case basis to check that packages are binary compatible. Packages have "package Version Numbers" ([JCVM]) that indicate binary compatibility or incompatibility between successive implementations of a package, which directly concern this requirement.

FDP_ITC.2.5/Installer:

- A package may depend on (import or use data from) other packages already installed. This dependency is explicitly stated in the loaded package in the form of a list of package AIDs.
- The intent of this rule is to ensure the binary compatibility of the package with those already on the card ([JCVM], §4.4).
- The installation (the invocation of an applet's install method by the installer) is implementation dependent ([JCRE], §11.2).
- Other rules governing the installation of an applet, that is, its registration to make it SELECTABLE by giving it a unique AID, are also implementation dependent (see, for example, [JCRE], §11).

FMT_SMR.1/Installer Security roles

FMT_SMR.1.1/Installer The TSF shall maintain the roles: **[assignment: Installer]**.

FMT_SMR.1.2/Installer The TSF shall be able to associate users with roles.

FPT_FLS.1/Installer Failure with preservation of secure state

FPT_FLS.1.1/Installer The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the installer fails to load/install a CAP file/applet as described in [JCRE] §11.1.5]**.

Application Note:

The TOE may provide additional feedback information to the card manager in case of potential security violations (see FAU_ARP.1).

FPT_RCV.3/Installer Automated recovery without undue loss

FPT_RCV.3.1/Installer When automated recovery from **[assignment: none]** is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2/Installer For **[assignment: failures during load/installation of a package/applet and deletion of a package/applet/object]**, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3/Installer The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding **[assignment: 0%]** for loss of TSF data or objects under the control of the TSF.

FPT_RCV.3.4/Installer The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

Application Note:

FPT_RCV.3.1/Installer:

- This element is not within the scope of the Java Card specification, which only mandates the behavior of the Java Card System in good working order. Further details on the "maintenance mode" shall be provided in specific implementations. The following is an excerpt from [\[CC2\]](#), p298: In this maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only authorized users should be allowed access to this mode but the real details of who can access this mode is a function of FMT: Security management. If FMT: Security management does not put any controls on who can access this mode, then it may be acceptable to allow any user to restore the system if the TOE enters such a state. However, in practice, this is probably not desirable as the user restoring the system has an opportunity to configure the TOE in such a way as to violate the SFRs.

FPT_RCV.3.2/Installer:

- Should the installer fail during loading/installation of a CAP file/applet, it has to revert to a "consistent and secure state". The Java Card RE has some clean up duties as well; see [\[JCRE\]](#), §11.1.5 for possible scenarios. This component shall include among the listed failures the deletion of a CAP file/applet. See ([\[JCRE\]](#), §11.3.4) for possible scenarios..
- Other events such as the unexpected tearing of the card, power loss, and so on, are partially handled by the underlying hardware platform (see [\[PP0084b1\]](#)) and, from the TOE's side, by events "that clear transient objects" and transactional features. See FPT_FLS.1.1/JCS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ABORT and FDP_ROL.1/FIREWALL.

FPT_RCV.3.3/Installer:

- The quantification is implementation dependent, but some facts can be recalled here. First, the SCP ensures the atomicity of updates for fields and objects, and a power-failure during a transaction or the normal runtime does not create the loss of otherwise-permanent data, in the sense that memory on a smart card is essentially persistent with this respect (EEPROM). Data stored on the RAM and subject to such failure is intended to have a limited lifetime anyway (runtime data on the stack, transient objects' contents). According to this, the loss of data within the TSF scope should be limited to the same restrictions of the transaction mechanism.

7.1.1.3 ADELG SFRs

This group consists of the SFRs related to the deletion of applets and/or CAP files, enforcing the applet deletion manager (ADEL) policy on security aspects outside the runtime. Deletion is a critical operation and therefore requires specific treatment.

FDP_ACC.2/ADEL Complete access control

FDP_ACC.2.1/ADEL The TSF shall enforce the **[assignment: ADEL access control SFP]** on **[S.ADEL, S.JCRE, S.JCVM, O.JAVAOBJECT, O.APPLET and O.CODE_CAP_FILE]** and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- OP.DELETE_APPLET,
- OP.DELETE_CAP_FILE,
- OP.DELETE_CAP_FILE_APPLET.

FDP_ACC.2.2/ADEL The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

FDP_ACF.1/ADEL Security attribute-based access control

FDP_ACF.1.1/ADEL The TSF shall enforce the [assignment: ADEL access control SFP] to objects based on the following [assignment :

Subject/Object	Attributes
S.JCVM	Active Applets
S.JCRE	Selected Applet Context, Registered Applets, Resident CAP files
O.CODE_CAP_FILE	AIDs of packages within a CAP file, Dependent package AID, Static References
O.APPLET	Applet Selection Status
O.JAVAOBJECT	Owner, Remote

].

FDP_ACF.1.2/ADEL The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:[assignment:

In the context of this policy, an object O is reachable if and only one of the following conditions hold:

- ✓ the owner of O is a registered applet instance A (O is reachable from A),
- ✓ a static field of a resident package P contains a reference to O (O is reachable from P),
- ✓ there exists a valid remote reference to O (O is remote reachable),
- ✓ there exists an object O' that is reachable according to either (1) or (2) or (3) above and O' contains a reference to O (the reachability status of O is that of O').

].

The following access control rules determine when an operation among controlled subjects and objects is allowed by the policy:

- R.JAVA.14 ([JCRE], §11.3.4.29, Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon an O.APPLET only if,
 - ✓ S.ADEL is currently selected,
 - ✓ there is no instance in the context of O.APPLET that is active in any logical channel and

- ✓ there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance distinct from O.APPLET, or O.JAVAOBJECT is reachable from a package P, or ([JCRE], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.15 ([JCRE], §11.3.4.2.110, Multiple Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon several O.APPLET only if,
 - ✓ S.ADEL is currently selected,
 - ✓ there is no instance of any of the O.APPLET being deleted that is active in any logical channel and
 - ✓ there is no O.JAVAOBJECT owned by any of the O.APPLET being deleted such that either O.JAVAOBJECT is reachable from an applet instance distinct from any of those O.APPLET, or O.JAVAOBJECT is reachable from a package, or ([JCRE], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.16 ([JCRE], §11.3.4.311, Applet/Library CAP file Deletion): S.ADEL may perform OP.DELETE_CAP_FILE upon an O.CODE_CAP_FILE only if,
 - ✓ S.ADEL is currently selected,
 - ✓ no reachable O.JAVAOBJECT, from a CAP file distinct from O.CODE_CAP_FILE that is an instance of a class that belongs to O.CODE_CAP_FILE, exists on the card and
 - ✓ there is no resident package on the card that depends on O.CODE_CAP_FILE.
- R.JAVA.17 ([JCRE], §11.3.4.412, Applet CAP file and Contained Instances Deletion): S.ADEL may perform OP.DELETE_CAP_FILE_APPLET upon an O.CODE_CAP_FILE only if,
 - ✓ S.ADEL is currently selected,
 - ✓ no reachable O.JAVAOBJECT, from a CAP file distinct from O.CODE_CAP_FILE, which is an instance of a class that belongs to O.CODE_CAP_FILE exists on the card,
 - ✓ there is no CAP file loaded on the card that depends on O.CODE_CAP_FILE, and
 - ✓ for every O.APPLET of those being deleted it holds that: (i) there is no instance in the context of O.APPLET that is active in any logical channel and (ii) there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance not being deleted, or O.JAVAOBJECT is reachable from a CAP file not being deleted, or ([JCRE], §8.5) O.JAVAOBJECT is remote reachable.

FDP_ACF.1.3/ADEL The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[assignment: none]**.

FDP_ACF.1.4/ADEL The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment: any subject but S.ADEL to O.CODE_CAP_FILE or O.APPLET for the purpose of deleting them from the card]**.

Application Note:

FDP_ACF.1.2/ADEL:

- This policy introduces the notion of reachability, which provides a general means to describe objects that are referenced from a certain applet instance or CAP file.
- S.ADEL calls the "uninstall" method of the applet instance to be deleted, if implemented by the applet, to inform it of the deletion request. The order in which these calls and the dependencies checks are performed are out of the scope of this Security Target.

FDP_RIP.1/ADEL Subset residual information protection

FDP_RIP.1.1/ADEL The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: deallocation of the resource from]** the following objects: **[assignment: applet instances and/or CAP files when one of the deletion operations in FDP_ACC.2.1/ADEL is performed on them]**.

Application Note:

Deleted freed resources (both code and data) may be reused, depending on the way they were deleted (logically or physically). Requirements on de-allocation during applet/CAP file deletion are described in [\[JCRE\]](#), §11.3.4.1, §11.3.4.2 and §11.3.4.3.

FMT_MSA.1/ADEL Management of security attributes

FMT_MSA.1.1/ADEL The TSF shall enforce the **[assignment: ADEL access control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment :Registered Applets and Resident CAP files]** to **[assignment: the Java Card RE]**.

FMT_MSA.3/ADEL Static attribute initialization

FMT_MSA.3.1/ADEL The TSF shall enforce the **[assignment: ADEL access control SFP]** to provide **[restrictive]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ADEL The TSF shall allow the **[assignment: none]**, to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/ADEL Specification of Management Functions

FMT_SMF.1.1/ADEL The TSF shall be capable of performing the following management functions: **[assignment: modify the list of registered applets' AIDs and the Resident CAP files]**.

FMT_SMR.1/ADEL Security roles

FMT_SMR.1.1/ADEL The TSF shall maintain the roles: **[assignment: applet deletion manager]**.

FMT_SMR.1.2/ADEL The TSF shall be able to associate users with roles.

FPT_FLS.1/ADEL Failure with preservation of secure state

FPT_FLS.1.1/ADEL The TSF shall preserve a secure state when the following types of failures occur:
[assignment: the applet deletion manager fails to delete a CAP file/applet as described in [JCRE], §11.3.4].

Application Note:

- The TOE may provide additional feedback information to the card manager in case of a potential security violation (see FAU_ARP.1).
- The CAP file/applet instance deletion must be atomic. The "secure state" referred to in the requirement must comply with Java Card specification ([\[JCRE\]](#), §11.3.4.)

7.1.1.4 ODELG SFRs

The following requirements concern the object deletion mechanism. This mechanism is triggered by the applet that owns the deleted objects by invoking a specific API method.

FDP_RIP.1/ODEL Subset residual information protection

FDP_RIP.1.1/ODEL The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: deallocation of the resource from]** the following objects:
[assignment: the objects owned by the context of an applet instance which triggered the execution of the method `javacard.framework.JCSystem.requestObjectDeletion()`].

Application Note:

- Freed data resources resulting from the invocation of the method `javacard.framework.JCSystem.requestObjectDeletion()` may be reused. Requirements on de-allocation after the invocation of the method are described in [\[JCAPI\]](#).
- There is no conflict with FDP_ROL.1 here because of the bounds on the rollback mechanism: the execution of `requestObjectDeletion()` is not in the scope of the rollback because it must be performed in between APDU command processing, and therefore no transaction can be in progress.

FPT_FLS.1/ODEL Failure with preservation of secure state

FPT_FLS.1.1/ODEL The TSF shall preserve a secure state when the following types of failures occur:
[assignment: the object deletion functions fail to delete all the unreferenced objects owned by the applet that requested the execution of the method].

Application Note:

The TOE may provide additional feedback information to the card manager in case of potential security violation (see FAU_ARP.1).

7.1.1.5 CARG SFRs

This group includes requirements for preventing the installation of CAP files that has not been bytecode verified, or that has been modified after bytecode verification.

FCO_NRO.2/CM Enforced proof of origin

FCO_NRO.2.1/CM The TSF shall enforce the generation of evidence of origin for transmitted **[assignment: application CAP files]** at all times.

FCO_NRO.2.2/CM The TSF shall be able to relate the **[assignment: identity]** of the originator of the information, and the **[assignment: application CAP file]**, of the information to which the evidence applies.

FCO_NRO.2.3/CM The TSF shall provide a capability to verify the evidence of origin of information to **[selection: recipient]** given **[assignment: limitations that no evidence is retained on the card for future verification once the evidence is verified]**.

Application Note:

FCO_NRO.2.1/CM:

- Upon reception of a new application CAP file for installation, the card manager shall first check that it actually comes from the verification authority and represented by the subject S.BCV. The verification authority is indeed the entity responsible for bytecode verification.

FCO_NRO.2.3/CM:

- The exact limitations on the evidence of origin are implementation dependent. In most of the implementations, the card manager performs an immediate verification of the origin of the CAP file using an electronic signature mechanism, and no evidence is kept on the card for future verification.

FDP_IFC.2/CM Complete information flow control

FDP_IFC.2.1/CM The TSF shall enforce the **[assignment: CAP FILE LOADING information flow control SFP]** on **[assignment: S.INSTALLER, S.BCV, S.CAD and LAPDU]** and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2/CM The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

Application Note:

- The subjects covered by this policy are those involved in the loading of an application CAP file by the card through a potentially unsafe communication channel.
- The operations that make information to flow between the subjects are those enabling to send a message through and to receive a message from the communication channel linking the card to the outside world. It is assumed that any message sent through the channel as clear text can be read by an attacker. Moreover, an attacker may capture any message sent through the communication channel and send its own messages to the other subjects.

- The information controlled by the policy is the APDUs exchanged by the subjects through the communication channel linking the card and the CAD. Each of those messages contain part of an application CAP file that is required to be loaded on the card, as well as any control information used by the subjects in the communication protocol.

FDP_IFF.1/CM Simple security attributes

FDP_IFF.1.1/CM The TSF shall enforce the **[assignment: CAP FILE LOADING information flow control SFP]** based on the following types of subject and information security attributes:

[assignment:

Subject / Information	Security Attribute
S.CAD	Secure channel keys, DAP key.
S.INSTALLER	Secure channel keys, DAP key, security level.
I.APDU	MAC, DAP block, sequence number.

].

FDP_IFF.1.2/CM The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

[assignment:

- the subject S.INSTALLER shall accept a message conveyed in I.APDU only if the subject S.CAD is authenticated during secure channel establishment,
- the subject S.INSTALLER shall accept an application CAP file only if the MAC and DAP block associated with I.APDU is verified and the sequence number in all the I.APDUs sent by the subject S.CAD is checked in order.

].

FDP_IFF.1.3/CM The TSF shall enforce the **[assignment: none]**.

FDP_IFF.1.4/CM The TSF shall explicitly authorize an information flow based on the following rules:
[assignment: none].

FDP_IFF.1.5/CM The TSF shall explicitly deny an information flow based on the following rules:

[assignment:

- **The TOE fails to verify the integrity and authenticity evidences of the application CAP file.**
- **CAP file which is not verified by off-card S.BCV.**

].

Application Note:

FDP_IFF.1.1/CM:

- The security attributes used to enforce the CAP FILE LOADING SFP are implementation dependent. More precisely, they depend on the communication protocol enforced between the CAD and the card. For instance, some of the attributes that can be used are: (1) the keys used by the subjects to encrypt/decrypt their messages; (2) the number of pieces the application CAP file has been split into in order to be sent to the card; (3) the ordinal of each piece in the decomposition of the CAP file, etc. See [\[GP Amendment D\]](#).

FDP_IFF.1.2/CM:

- The precise set of rules to be enforced by the function is implementation dependent. The whole exchange of messages shall verify at least the following two rules: (1) the subject S.INSTALLER shall accept a message only if it comes from the subject S.CAD; (2) the subject S.INSTALLER shall accept an application CAP file only if it has received without modification and in the right order all the APDUs sent by the subject S.CAD.

FDP_IFF.1.5/CM:

- The verification of the integrity and authenticity evidences can be performed either during loading or during the first installation of an application of the CAP file.

FDP_UIT.1/CM Data exchange integrity

FDP_UIT.1.1/CM The TSF shall enforce the **[assignment: CAP FILE LOADING information flow control SFP]** to **[selection: receive]** user data in a manner protected from **[selection: modification, deletion, insertion, replay]** errors.

FDP_UIT.1.2/CM [Refined] The TSF shall be able to determine on receipt of user data, whether **[selection: modification, deletion, insertion, replay of some of the pieces of the application sent by the CAD]** has occurred.

Application Note:

Modification errors should be understood as modification, substitution, unrecoverable ordering change of data and any other integrity error that may cause the application CAP file to be installed on the card to be different from the one sent by the CAD.

FIA_UID.1/CM Timing of identification

FIA_UID.1.1/CM The TSF shall allow **[assignment: applet selection, APDU dispatch and initiation of secure channel]** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2/CM The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note:

The list of TSF-mediated actions is implementation-dependent, but CAP file installation requires the user to be identified. Here by user is meant the one(s) that in the Security Target shall be associated to

the role(s) defined in the component FMT_SMR.1/CM.

FMT_MSA.1/CM Management of security attributes

FMT_MSA.1.1/CM The TSF shall enforce the [assignment: CAP FILE LOADING information flow control SFP] to restrict the ability to [selection: change, modify and delete] the security attributes [assignment: Package AID] to [assignment: none].

FMT_MSA.3/CM Static attribute initialization

FMT_MSA.3.1/CM The TSF shall enforce the [assignment: CAP FILE LOADING information flow control SFP] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/CM The TSF shall allow the [assignment: none] to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/CM Specification of Management Functions

FMT_SMF.1.1/CM The TSF shall be capable of performing the following management functions: [assignment: Card Content Management consisting of loading, installing, extradition and deletion by Card Manager].

FMT_SMR.1/CM Security roles

FMT_SMR.1.1/CM The TSF shall maintain the roles [assignment: Card Manager].

FMT_SMR.1.2/CM The TSF shall be able to associate users with roles.

FTP_ITC.1/CM Inter-TSF trusted channel

FTP_ITC.1.1/CM The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/CM [Refined] The TSF shall permit [selection: the CAD placed in the card issuer secured environment] to initiate communication via the trusted channel.

FTP_ITC.1.3/CM The TSF shall initiate communication via the trusted channel for [assignment: loading/installing a new application CAP file on the card].

Application Note:

There is no dynamic CAP file loading on the Java Card platform. New CAP files can be installed on the card only on demand of the card issuer.

7.1.1.6 SCPG SFRs

This group contains the security requirements for the smart card platform, that is, operating system and chip that the Java Card System is implemented upon. The requirements are expressed in terms of security functional requirements from [\[CC2\]](#).

FPT_TST.1/SCP TSF Testing

FPT_TST.1.1/SCP The TSF shall run a suite of self-tests **[assignment: during initial start-up]** to demonstrate the correct operation of **[assignment: security mechanisms of the IC]**.

FPT_TST.1.2/SCP The TSF shall provide authorized users with the capability to verify the integrity of **[assignment: user PIN, user Keys]**.

FPT_TST.1.3/SCP The TSF shall provide authorized users with the capability to verify the integrity of **[assignment: Keys]**.

FPT_PHP.3/SCP Resistance to physical attacks

FPT_PHP.3.1/SCP The TSF shall resist **[assignment: physical manipulation and physical probing]** to the **[assignment: all TOE components implementing the TSF]** by responding automatically such that the SFRs are always enforced.

FPT_RCV.4/SCP Function recovery

FPT_RCV.4.1/SCP The TSF shall ensure that **[assignment: reading from and writing to static and objects' fields interrupted by power loss]** have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

FCS_RNG.1 Random number generation

FCS_RNG.1.1 The TSF shall provide a **[selection: physical]** random number generator that implements: **[assignment: the following defined security capabilities of the PTG.2 RNG classes]**

- A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.
- If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.
- The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started. And (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.

- The online test procedure shall be effective to detect non-tolerable weakness of the random numbers soon.
- The online test procedure checks the quality of the raw random number sequence. It is triggered applied upon specified internal events. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.

].

FCS_RNG.1.2 The TSF shall provide [selection: 32 bit random number words] that meet : [assignment: *the following defined security capabilities of the PTG.2 RNG classes*

- Test procedure A and no other test suites does not distinguish the internal random numbers from output sequences of an ideal RNG.
- The average Shannon entropy per internal random bit exceeds 0.997.

].

7.1.1.7 CMGRG SFRs

This group includes requirements for Card Manager.

FDP_ACC.1/CMGR Subset access control

FDP_ACC.1.1/CMGR The TSF shall enforce the [assignment: **CARD CONTENT MANAGEMENT access control SFP**] on [assignment: **S.CM, S.CAD, O.APPLET, O.CODE_CAP_FILE, O.GP_KEY** and operations among subjects and objects covered by the SFP].

Refinement:

The operations involved in the policy are:

- OP.LOAD,
- OP.INSTALL,
- OP.DELETE,
- OP.PUT_KEY.

FDP_ACF.1/CMGR Security attribute-based access control

FDP_ACF.1.1/CMGR The TSF shall enforce the [assignment: **CARD CONTENT MANAGEMENT access control SFP**] to objects based on the following:[assignment:

Subject/Object	Security attributes
S.CM	Card Life Cycle, Life Cycle Status, Privileges, Registered Applets, Active Applets, Applet Selection Status, Security Level, GP key.

S.CAD	Security Level, DM token, GP key.
O.CODE_CAP_FILE	DAP, MAC.
O.GP_KEY	Key types and key checksum.

].

FDP_ACF.1.2/CMGR The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: the runtime behaviors defined for the following operations as defined in [\[GP\]](#):

- The S.CM load O.CODE_CAP_FILE into the card on behalf of S.CAD for OP.LOAD operation based on the attributes defined in [\[GP\]](#);
- install O.CODE_CAP_FILE on the card on behalf of S.CAD for OP.INSTALL operation based on the attributes defined in [\[GP\]](#);
- delete O.CODE_CAP_FILE from the card on behalf of S.CAD for OP.DELETE operation based on the attributes defined in [\[GP\]](#);
- load O.GP_KEY into the card on behalf of S.CAD for OP.PUT_KEY operation based on the attributes defined in [\[GP\]](#).

FDP_ACF.1.3/CMGR The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[assignment: none]**.

FDP_ACF.1.4/CMGR The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment: Only O.CODE_CAP_FILE can be loaded or deleted by S.CM]**.

FMT_MSA.1/CMGR Management of security attributes

FMT_MSA.1.1/CMGR The TSF shall enforce the **[assignment: CARD CONTENT MANAGEMENT access control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: Privileges, Life-cycle Status, Security Level, GP key]** to **[assignment: S.CM]**.

FMT_MSA.3/CMGR Static attribute initialization

FMT_MSA.3.1/CMGR The TSF shall enforce the **[assignment: CARD CONTENT MANAGEMENT access control SFP]** to provide **[selection: restrictive]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/CMGR The TSF shall allow the **[assignment: none]** to specify alternative initial values to override the default values when an object or information is created.

7.1.1.8 OSU SFRs

FDP_IFC.2/OSU Complete information flow control

FDP_IFC.2.1/OSU The TSF shall enforce the [assignment: OS Update information flow control SFP] on [assignment: S.OSU and D.UPDATE_IMAGE].

FDP_IFC.2.2/OSU The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

FDP_IFF.1/OSU Simple security attributes

FDP_IFF.1.1/OSU The TSF shall enforce the [assignment: OS Update information flow control SFP] based on the following types of subject and information security attributes:[assignment:

Subjects	Security attributes
S.OSU	Current Version Number, Verification Key, Package Decryption Key, Received Version Number

].

FDP_IFF.1.2/OSU The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:[assignment:

- The TOE shall reject the D.UPDATE_IMAGE which signature verification failed.
- The TOE shall reject the D.UPDATE_IMAGE that cannot be decrypted.

].

FDP_IFF.1.3/OSU The TSF shall enforce the additional information flow control SFP rules:[assignment:

- The TSF shall only authorize D.UPDATE_IMAGE if the following rules apply:
 - The Received Version shall be higher than or equal to the Current Version.

].

FDP_IFF.1.4/OSU The TSF shall explicitly authorise an information flow based on the following rules:[assignment: none].

FDP_IFF.1.5/OSU The TSF shall explicitly deny an information flow based on the following rules:[assignment:none].

Application note:

The on-card S.OSU role interacts with the off-card S.UpdateImageCreator via OSU commands. The D.UPDATE_IMAGE is split up into smaller chunks and transmitted as payload within the OSU Commands to the TOE.

Decrypting the D.UPDATE_IMAGE with the Package Decryption Key prevents the authorization of the D.UPDATE_IMAGE for the update process on a not certified system. The Package Decryption Key is only available on a certified TOE.

FMT_MSA.3 /OSU Static attribute initialisation

FMT_MSA.3.1/OSU The TSF shall enforce the [assignment: OS Update information flow control SFP] to provide [selection: restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/OSU The TSF shall allow the [assignment: S.OSU] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1/OSU Management of security attributes

FMT_MSA.1.1/OSU The TSF shall enforce the [assignment: OS Update information flow control SFP] to restrict the ability to [selection: modify] the security attributes [assignment: Current Version Number] to [assignment: S.OSU].

FMT_SMR.1/OSU Security roles

FMT_SMR.1.1/OSU The TSF shall maintain the roles [assignment: S.OSU].

FMT_SMR.1.2/OSU The TSF shall be able to associate users with roles.

Application note:

The roles defined by the TMS OSU authentication key and cipher key.

FMT_SMF.1/OSU Specification of Management Functions

FMT_SMF.1.1/OSU The TSF shall be capable of performing the following management functions:

[assignment:

- query Current Version Number

].

Application note:

After the atomic activation of the additional code the Final version Number is returned on querying the Current version Number.

FIA_UID.1/OSU Timing of Identification

FIA_UID.1.1/OSU The TSF shall allow [assignment: OP.TRIGGER_OSU] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 /OSU The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.1/OSU Timing of authentication

FIA_UAU.1.1/OSU The TSF shall allow **[assignment: OP.TRIGGER_OSU]** on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2/OSU The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.4/OSU Single-use authentication mechanisms

FIA_UAU.4.1/OSU The TSF shall prevent reuse of authentication data related to **[assignment: the authentication mechanism used to load D.UPDATE_IMAGE]**.

FPT_FLS.1/OSU Failure with preservation of secure state

FPT_FLS.1.1/OSU The TSF shall preserve a secure state when the following types of failures occur:
[assignment:

- **Corrupted D.UPDATE_IMAGE is received.**
- **Unauthorized D.UPDATE_IMAGE is received.**
- **The OS Update Process is interrupted.**
- **The activation of the additional code failed.**

].

7.1.2 Security Requirements Rationale

7.1.2.1 Identification

O.SID Subjects' identity is AID-based (applets, packages and CAP files), and is met by the following SFRs: FDP_ITC.2/Installer, FIA_ATD.1/AID, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_MSA.1/ADEL, FMT_MSA.1/CM, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.3/CM, FMT_SMF.1/CM, FMT_SMF.1/ADEL, FMT_MTD.1/JCRE and FMT_MTD.3/JCRE.

Installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities (FIA_UID.2/AID, FIA_USB.1/AID).

7.1.2.2 Execution

O.FIREWALL This objective is met by the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL, the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) and the functional requirement FDP_ITC.2/Installer. The functional requirements of the class FMT (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1/Installer, FMT_SMR.1/JCS, FMT_SMF.1/JCS, FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_SMF.1/CM, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMR.1/CM, FMT_MSA.2/FIREWALL_JCVM, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM) also indirectly contribute to meet this objective.

O.GLOBAL_ARRAYS_CONFID Only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer, the global byte array input parameter (bArray) to an applet's install method and the global arrays created by the JCSYSTEM.makeGlobalArray(...) method. The clearing requirement of these arrays is met by (FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray and FDP_RIP.1/bArray respectively). The JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.

If the TOE provides JCRMI functionality, protection of the array parameters of remotely invoked methods, which are global as well, is covered by the general initialization of method parameters (FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL and FDP_RIP.1/TRANSIENT).

O.GLOBAL_ARRAYS_INTEG This objective is met by the JCVM information flow control policy

(FDP_IFF.1/JCVM, FDP_IFC.1/JCVM), which prevents an application from keeping a pointer to the APDU buffer of the card, to the global byte array of the applet's install method or to the global arrays created by the JCSYSTEM.makeGlobalArray(...) method. Such a pointer could be used to access and modify it when the buffer is being used by another application.

O.ARRAY_VIEWS_CONFID Array views have security attributes of temporary objects where the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from storing a reference to the array view. Furthermore, array views may not have ATTR_READABLE_VIEW security attribute which ensures that no application can read the contents of the array view.

O.ARRAY_VIEWS_INTEG Array views have security attributes of temporary objects where the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from storing a reference to the array view. Furthermore, array views may not have ATTR_WRITABLE_VIEW security attribute which ensures that no application can alter the contents of the array view.

O.NATIVE This security objective is covered by FDP_ACF.1/FIREWALL: the only means to execute native code is the invocation of a Java Card API method. This objective mainly relies on the environmental objective OE.CAP_FILE, which uphold the assumption A.CAP_FILE.

O.OPERATE_JCS The TOE is protected in various ways against applets' actions (FPT_TDC.1), the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL, and is able to detect and block various failures or security violations during usual working (FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FPT_FLS.1/Installer, FAU_ARP.1). Its security-critical parts and procedures are also protected: safe recovery from failure is ensured (FPT_RCV.3/Installer), applets' installation may be cleanly aborted (FDP_ROL.1/FIREWALL), communication with external users and their internal subjects is well-controlled (FDP_ITC.2/Installer, FIA_ATD.1/AID, FIA_USB.1/AID) to prevent alteration of TSF data (also protected by components of the FPT class).

Almost every objective and/or functional requirement indirectly contributes to this one too.

Application note:

Startup of the TOE (TSF-testing) can be covered by FPT_TST.1. This SFR component is not mandatory in [JCRE], but appears in most of security requirements documents for masked applications. Testing could also occur randomly. Self-tests may become mandatory in order to comply with FIPS certification [FIPS 140-2].

O.REALLOCATION This security objective is satisfied by the following SFRs: FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ADEL, which imposes that the contents of the re-allocated block shall always be cleared before delivering the

block.

O.RESOURCES The TSFs detects stack/memory overflows during execution of applications (FAU_ARP.1, FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FPT_FLS.1/Installer). Failed installations are not to create memory leaks (FDP_ROL.1/FIREWALL, FPT_RCV.3/Installer) as well. Memory management is controlled by the TSF (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1/Installer, FMT_SMR.1/JCS, FMT_SMF.1/JCS, FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_SMF.1/CM and FMT_SMR.1/CM).

Additionally, if the TOE provides JCRMI functionality, memory management is controlled by the TSF FMT_SMR.1/JCRMI, and FMT_SMF.1/JCRMI.

7.1.2.3 Services

O.ALARM This security objective is met by FPT_FLS.1/Installer, FPT_FLS.1/JCS, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL which guarantee that a secure state is preserved by the TSF when failures occur, and FAU_ARP.1 which defines TSF reaction upon detection of a potential security violation.

O.CIPHER This security objective is directly covered by FCS_CKM.1, FCS_CKM.6, FCS_CKM.2, FCS_CKM.3 and FCS_COP.1. The SFR FPR_UNO.1 contributes in covering this security objective and controls the observation of the cryptographic operations which may be used to disclose the keys.

O.RNG This security objective is directly covered by FCS_RNG.1 which ensures the cryptographic quality of random number generation.

O.KEY-MNGT This relies on the same security functional requirements as O.CIPHER, plus FDP_RIP.1 and FDP_SDI.2/DATA as well. Precisely it is met by the following components: FCS_CKM.1, FCS_CKM.6, FCS_CKM.2, FCS_CKM.3, FCS_COP.1, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL and FDP_RIP.1/TRANSIENT.

O.PIN-MNGT This security objective is ensured by FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FPR_UNO.1, FDP_ROL.1/FIREWALL and FDP_SDI.2/DATA security functional requirements. The TSFs behind these are implemented by API classes. The firewall security functions FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL shall protect the access to private and internal data of the objects.

O.TRANSACTION Directly met by FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT,

FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT and FDP_RIP.1/OBJECTS (more precisely, by the element FDP_RIP.1.1/ABORT).

7.1.2.4 Object Deletion

O.OBJ-DELETION This security objective specifies that deletion of objects is secure. The security objective is met by the security functional requirements FDP_RIP.1/ODEL and FPT_FLS.1/ODEL.

7.1.2.5 Applet Management

O.DELETION This security objective specifies that applet and CAP file deletion must be secure. The non-introduction of security holes is ensured by the ADEL access control policy (FDP_ACC.2/ADEL, FDP_ACF.1/ADEL). The integrity and confidentiality of data that does not belong to the deleted applet or CAP file is a by-product of this policy as well. Non-accessibility of deleted data is met by FDP_RIP.1/ADEL and the TSFs are protected against possible failures of the deletion procedures (FPT_FLS.1/ADEL, FPT_RCV.3/Installer). The security functional requirements of the class FMT (FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL) included in the group ADELG also contribute to meet this objective.

O.LOAD This security objective specifies that the loading of a CAP file into the card must be secure. Evidence of the origin of the CAP file is enforced (FCO_NRO.2/CM) and the integrity of the corresponding data is under the control of the CAP FILE LOADING information flow policy (FDP_IFC.2/CM, FDP_IFF.1/CM) and FDP_UTI.1/CM. Appropriate identification (FIA_UID.1/CM) and transmission mechanisms are also enforced (FTP_ITC.1/CM).

O.INSTALL This security objective specifies that installation of applets must be secure. Security attributes of installed data are under the control of the FIREWALL access control policy (FDP_ITC.2/Installer), and the TSFs are protected against possible failures of the installer (FPT_FLS.1/Installer, FPT_RCV.3/Installer).

7.1.2.6 Card Manager

O.CARD_MANAGEMENT This security objective specifies that the access control to card management functions. This is enforced by FDP_ACC.1/CMGR, FDP_ACF.1/CMGR, FMT_MSA.1/CMGR, FMT_MSA.3/CMGR, FMT_SMF.1/CM.

7.1.2.7 Smart Card Platform

O.SCP.RECOVERY This security objective specifies that the platform must behave securely if an

unexpected loss of power occurs. This is covered by FPT_RCV.4/SCP which specifies the recovery after unexpected power failure.

O.SCP.SUPPORT This security objective specifies that the SCP provides security features to the JCS. This is provided by FPT_TST.1/SCP. This is also provided by requirements of the IC, which are described in [[THN31 1.0.2](#)].

O.SCP.IC This security objective specifies that the IC must provide mechanisms to protect itself against physical attacks. This is provided by FPT_PHP.3/SCP. This is also provided by requirements of the IC, which are described in [[THN31 1.0.2](#)].

7.1.2.8 OS Update

O.CONFID-UPDATE-IMAGE.LOAD This security objective specifies that the D.UPDATE_IMAGE transferred to the device is not disclosed during the installation, FPR_UNO.1 contributes by ensuring the non-observability of the keys. FIA_UID.1/OSU contributes by requiring identification and FIA_UAU.1/OSU contributes by requiring authentication.

O.AUTH-LOAD-UPDATE-IMAGE This security objective specifies that the D.UPDATE_IMAGE must be authenticated. The OSU information flow control policy(FDP_IFC.2/OSU, FDP_IFF.1/OSU) prevents the information or processes from being modified. FMT_MSA.3/OSU contributes by enforcing restrictive default values for the attributes of the OS Update information flow control SFP. FMT_SMR.1/OSU contributes by letting S.OSU handle the OS Update procedure. FIA_UID.1/OSU contributes by requiring identification of the authorized images. FIA_UAU.1/OSU contributes by requiring authentication of the authorized images.

O.SECURE_LOAD_ACODE This security objective specifies that :

- the Loader shall check an evidence of authenticity and integrity of the loaded Additional Code. This is provided by FDP_IFC.2/OSU.
- the TOE shall remain secure state when Load Phase interrupted. This is provided by FPT_FLS.1/OSU.
- the authenticity and integrity of D.UPDATE_IMAGE, this is provided by FIA_UAU.4/OSU.

O.SECURE_AC_ACTIVATION This security objective specifies that the activation of the Additional Code and update of the Identification Data shall be performed at the same time in an Atomic way. This is provided by FPT_FLS.1/OSU. FMT_MSA.1/OSU contributes by enforcing OS Update information flow control SFP to restrictive the ability to modify current version number.

O.TOE_IDENTIFICATION This security objective specifies that :

- the TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data. This is provided by FDP_SDI.2/DATA.

- the user shall be able to uniquely identify Initial TOE and Additional Code(s) which are embedded in the Final TOE. This is provided by FMT_SMF.1/OSU.

7.1.3 Rationale Tables of Security Objectives and SFRs

Security Objectives	Security Functional requirements
O.SID	FIA_ATD.1/AID, FIA_UID.2/AID, FMT_MSA.1/JCRE, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.1/CM, FMT_MSA.3/CM, FDP_ITC.2/Installer, FMT_SMF.1/CM, FMT_SMF.1/ADEL, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FIA_USB.1/AID, FMT_MSA.1/JCVM, FMT_MSA.3/JCVM
O.FIREWALL	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FMT_SMR.1/Installer, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMR.1/CM, FMT_MSA.3/FIREWALL, FMT_SMR.1/JCS, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL, FMT_MSA.1/JCRE, FDP_ITC.2/Installer, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FMT_SMF.1/ADEL, FMT_SMF.1/CM, FMT_SMF.1/JCS, FMT_MSA.2/FIREWALL_JCVM, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_MSA.1/JCVM, FMT_MSA.3/JCVM
O.GLOBAL_ARRAYS_CONFID	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FDP_RIP.1/bArray, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT
O.GLOBAL_ARRAYS_INTEG	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM
O.ARRAY_VIEWS_CONFID	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FDP_ACC.2/Firewall, FDP_ACF.1/Firewall
O.ARRAY_VIEWS_INTEG	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FDP_ACC.2/Firewall, FDP_ACF.1/Firewall
O.NATIVE	FDP_ACF.1/FIREWALL
O.OPERATE_JCS	FAU_ARP.1, FDP_ROL.1/FIREWALL, FIA_ATD.1/AID, FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FPT_FLS.1/Installer, FDP_ITC.2/Installer, FPT_RCV.3/Installer,

	FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FPT_TDC.1, FIA_USB.1/AID
O.REALLOCATION	FDP_RIP.1/ABORT, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ADEL, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS
O.RESOURCE	FAU_ARP.1, FDP_ROL.1/FIREWALL, FMT_SMR.1/Installer, FMT_SMR.1/JCS, FMT_SMR.1/ADEL, FPT_FLS.1/Installer, FPT_FLS.1/ODEL, FPT_FLS.1/JCS, FPT_FLS.1/ADEL, FPT_RCV.3/Installer, FMT_SMR.1/CM, FMT_SMF.1/ADEL, FMT_SMF.1/CM, FMT_SMF.1/JCS, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE
O.ALARM	FPT_FLS.1/Installer, FPT_FLS.1/JCS, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL, FAU_ARP.1
O.CIPHER	FCS_CKM.1, FCS_CKM.6, FCS_CKM.2, FCS_CKM.3, FCS_COP.1, FPR_UNO.1
O.RNG	FCS_RNG.1
O.KEY-MNGT	FCS_CKM.1, FCS_CKM.6, FCS_CKM.2, FCS_CKM.3, FCS_COP.1, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_SDI.2/DATA, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT
O.PIN-MNGT	FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FPR_UNO.1, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_ROL.1/FIREWALL, FDP_SDI.2/DATA, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL
O.TRANSACTION	FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT, FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_RIP.1/OBJECTS
O.OBJ-DELETION	FDP_RIP.1/ODEL, FPT_FLS.1/ODEL
O.DELETION	FDP_ACC.2/ADEL, FDP_ACF.1/ADEL, FDP_RIP.1/ADEL, FPT_FLS.1/ADEL, FPT_RCV.3/Installer, FMT_MSA.1/ADEL,

	FMT_MSA.3/ADEL, FMT_SMR.1/ADEL
O.LOAD	FCO_NRO.2/CM, FDP_IFC.2/CM, FDP_IFF.1/CM, FDP_UTI.1/CM, FIA_UID.1/CM, FTP_ITC.1/CM
O.INSTALL	FDP_ITC.2/Installer, FPT_RCV.3/Installer, FPT_FLS.1/Installer
O.CARD-MANAGEMENT	FDP_ACC.1/CMGR, FDP_ACF.1/CMGR, FMT_MSA.1/CMGR, FMT_MSA.3/CMGR, FMT_SMF.1/CM
O.SCP.IC	FPT_RCV.4/SCP
O.SCP.RECOVERY	FPT_TST.1/SCP
O.SCP.SUPPORT	FPT_PHP.3/SCP
O.SECURE_LOAD_ACODE	FDP_IFC.2/OSU, FPT_FLS.1/OSU, FIA_UAU.4/OSU
O.SECURE_AC_ACTIVATION	FPT_FLS.1/OSU, FMT_MSA.1/OSU
O.TOE_IDENTIFICATION	FMT_SMF.1/OSU, FDP_SDI.2/DATA
O.CONFID-UPDATE-IMAGE.LOAD	FIA_UID.1/OSU, FIA_UAU.1/OSU, FPR_UNO.1
O.AUTH-LOAD-UPDATE-IMAGE	FDP_IFC.2/OSU, FDP_IFF.1/OSU, FMT_SMR.1/OSU, FMT_MSA.3/OSU, FIA_UID.1/OSU, FIA_UAU.1/OSU

Table 36: Security Objectives and SFRs - Coverage

Security Functional Requirements	Security Objectives
FDP_ACC.2/FIREWALL	O.FIREWALL, O.OPERATE_JCS, O.PIN-MNGT, O.ARRAY_VIEWS_CONFID, O.ARRAY_VIEWS_INTEG
FDP_ACF.1/FIREWALL	O.FIREWALL, O.NATIVE, O.OPERATE_JCS, O.PIN-MNGT, O.ARRAY_VIEWS_CONFID, O.ARRAY_VIEWS_INTEG
FDP_IFC.1/JCVM	O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_CONFID, O.ARRAY_VIEWS_INTEG
FDP_IFF.1/JCVM	O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_CONFID, O.ARRAY_VIEWS_INTEG
FDP_RIP.1/OBJECTS	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION
FMT_MSA.1/JCRE	O.SID, O.FIREWALL
FMT_MSA.1/JCVM	O.SID, O.FIREWALL
FMT_MSA.2/FIREWALL_JCVM	O.FIREWALL
FMT_MSA.3/FIREWALL	O.SID, O.FIREWALL

FMT_MSA.3/JCVM	O.SID, O.FIREWALL
FMT_SMF.1/JCS	O.FIREWALL, O.RESOURCES
FMT_SMR.1/JCS	O.FIREWALL, O.RESOURCES
FCS_CKM.1	O.CIPHER, O.KEY-MNGT
FCS_CKM.6	O.CIPHER, O.KEY-MNGT
FCS_COP.1	O.CIPHER, O.KEY-MNGT
FCS_RNG.1	O.RNG
FDP_RIP.1/ABORT	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION
FDP_RIP.1/APDU	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION
FDP_RIP.1/bArray	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION
FDP_RIP.1/GlobalArray	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION
FDP_RIP.1/KEYS	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION
FDP_RIP.1/TRANSIENT	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION
FDP_ROL.1/FIREWALL	O.OPERATE_JCS, O.RESOURCES, O.PIN-MNGT, O.TRANSACTION
FAU_ARP.1	O.OPERATE_JCS, O.RESOURCES, O.ALARM
FDP_SDI.2/DATA	O.KEY-MNGT, O.PIN-MNGT, O.TOE_IDENTIFICATION
FPR_UNO.1	O.CIPHER, O.KEY-MNGT, O.PIN-MNGT, O.CONFID-UPDATE-IMAGE.LOAD
FPT_FLS.1/JCS	O.OPERATE_JCS, O.RESOURCES, O.ALARM
FPT_TDC.1	O.OPERATE_JCS
FIA_ATD.1/AID	O.SID, O.OPERATE_JCS
FIA_UID.2/AID	O.SID
FIA_USB.1/AID	O.SID, O.OPERATE_JCS
FMT_MTD.1/JCRE	O.SID, O.FIREWALL, O.RESOURCES
FMT_MTD.3/JCRE	O.SID, O.FIREWALL, O.RESOURCES
FDP_ITC.2/Installer	O.SID, O.FIREWALL, O.OPERATE_JCS, O.INSTALL
FMT_SMR.1/Installer	O.FIREWALL, O.RESOURCES

FPT_FLS.1/Installer	O.OPERATE_JCS, O.RESOURCES, O.ALARM, O.INSTALL
FPT_RCV.3/Installer	O.OPERATE_JCS, O.RESOURCES, O.DELETION, O.INSTALL
FDP_ACC.2/ADEL	O.DELETION
FDP_ACF.1/ADEL	O.DELETION
FDP_RIP.1/ADEL	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.DELETION
FMT_MSA.1/ADEL	O.SID, O.FIREWALL, O.DELETION
FMT_MSA.3/ADEL	O.SID, O.FIREWALL, O.DELETION
FMT_SMF.1/ADEL	O.SID, O.FIREWALL, O.RESOURCES
FMT_SMR.1/ADEL	O.FIREWALL, O.RESOURCES, O.DELETION
FPT_FLS.1/ADEL	O.OPERATE_JCS, O.RESOURCES, O.ALARM, O.DELETION
FDP_RIP.1/ODEL	O.GLOBAL_ARRAYS_CONFID, O.REALLOCATION, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.OBJ-DELETION
FPT_FLS.1/ODEL	O.OPERATE_JCS, O.RESOURCES, O.ALARM, O.OBJ-DELETION
FCO_NRO.2/CM	O.LOAD
FDP_IFC.2/CM	O.LOAD
FDP_IFF.1/CM	O.LOAD
FDP_UIT.1/CM	O.LOAD
FIA_UID.1/CM	O.LOAD
FMT_MSA.1/CM	O.SID, O.FIREWALL
FMT_MSA.3/CM	O.SID, O.FIREWALL
FMT_SMF.1/CM	O.SID, O.FIREWALL, O.RESOURCES, O.CARD-MANAGEMENT
FMT_SMR.1/CM	O.FIREWALL, O.RESOURCES
FTP_ITC.1/CM	O.LOAD
FDP_ACC.1/CMGR	O.CARD-MANAGEMENT
FDP_ACF.1/CMGR	O.CARD-MANAGEMENT
FMT_MSA.1/CMGR	O.CARD-MANAGEMENT
FMT_MSA.3/CMGR	O.CARD-MANAGEMENT
FPT_RCV.4/SCP	O.SCP.IC
FPT_TST.1/SCP	O.SCP.RECOVERY
FPT_PHP.3/SCP	O.SCP.SUPPORT
FDP_IFC.2/OSU	O.AUTH-LOAD-UPDATE-IMAGE O.SECURE_LOAD_ACODE
FDP_IFF.1/OSU	O.AUTH-LOAD-UPDATE-IMAGE
FMT_MSA.3 /OSU	O.AUTH-LOAD-UPDATE-IMAGE

FMT_MSA.1/OSU	O.SECURE_AC_ACTIVATION
FMT_SMR.1/OSU	O.AUTH-LOAD-UPDATE-IMAGE
FMT_SMF.1/OSU	O.TOE_IDENTIFICATION
FIA_UID.1/OSU	O.CONFID-UPDATE-IMAGE.LOAD O.AUTH-LOAD-UPDATE-IMAGE
FIA_UAU.1 /OSU	O.CONFID-UPDATE-IMAGE.LOAD O.AUTH-LOAD-UPDATE-IMAGE
FIA_UAU.4 /OSU	O.SECURE_LOAD_ACODE
FPT_FLS.1/OSU	O.SECURE_LOAD_ACODE O.SECURE_AC_ACTIVATION
FCS_CKM.2	O.CIPHER, O.KEY-MNGT
FCS_CKM.3	O.CIPHER, O.KEY-MNGT

Table 37:SFRs and Security Objectives

7.1.4 Dependencies for Java Card System

7.1.4.1 SFRs Dependencies for Java Card System

Requirements	CC Dependencies	Satisfied Dependencies
FDP_ACC.2/FIREWALL	(FDP_ACF.1)	FDP_ACF.1/FIREWALL
FDP_ACF.1/FIREWALL	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2/FIREWALL, FMT_MSA.3/FIREWALL
FDP_IFC.1/JCVM	(FDP_IFF.1)	FDP_IFF.1/JCVM
FDP_IFF.1/JCVM	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1/JCVM, FMT_MSA.3/JCVM
FDP_IFF.1/OSU	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.2/OSU, FMT_MSA.3/OSU
FDP_RIP.1/OBJECTS	No Dependencies	
FMT_MSA.1/JCRE	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL, FMT_SMR.1/JCS, Note: The dependency FMT_SMF.1 is discarded, see the notes below for more details.
FMT_MSA.1/JCVM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL, FDP_IFC.1/JCVM, FMT_SMF.1/JCS, FMT_SMR.1/JCS
FMT_MSA.2/FIREWALL_JCVM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_MSA.1) and	FDP_ACC.2/FIREWALL, FDP_IFC.1/JCVM, FMT_MSA.1/JCRE,

	(FMT_SMR.1)	FMT_MSA.1/JCVM, FMT_SMR.1/JCS
FMT_MSA.3/FIREWALL	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_SMR.1/JCS
FMT_MSA.3/JCVM	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCVM, FMT_SMR.1/JCS
FMT_SMF.1/JCS	No Dependencies	
FMT_SMR.1/JCS	(FIA_UID.1)	FIA_UID.2/AID
FCS_CKM.1	(FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1) and (FCS_RBG.1 or FCS_RNG.1) and (FCS_CKM.6)	FCS_COP.1, FCS_CKM.6, FCS_RNG.1, FCS_CKM.2
FCS_CKM.6	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5)	FCS_CKM.1
FCS_COP.1	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.5) and (FCS_CKM.6)	FCS_CKM.1, FCS_CKM.6
FCS_RNG.1	No Dependencies	
FDP_RIP.1/ABORT	No Dependencies	
FDP_RIP.1/APDU	No Dependencies	
FDP_RIP.1/bArray	No Dependencies	
FDP_RIP.1/GlobalArray	No Dependencies	
FDP_RIP.1/KEYS	No Dependencies	
FDP_RIP.1/TRANSIENT	No Dependencies	
FDP_ROL.1/FIREWALL	(FDP_ACC.1 or FDP_IFC.1)	FDP_ACC.2/FIREWALL, FDP_IFC.1/JCVM
FAU_ARP.1	(FAU_SAA.1)	See note below
FDP_SDI.2/DATA	No Dependencies	
FPR_UNO.1	No Dependencies	
FPT_FLS.1/JCS	No Dependencies	
FPT_TDC.1	No Dependencies	
FIA_ATD.1/AID	No Dependencies	
FIA_UID.2/AID	No Dependencies	
FIA_USB.1/AID	(FIA_ATD.1)	FIA_ATD.1/AID
FMT_MTD.1/JCRE	(FMT_SMF.1) and (FMT_SMR.1)	FMT_SMF.1/JCS, FMT_SMR.1/JCS

FMT_MTD.3/JCRE	(FMT_MTD.1)	FMT_MTD.1/JCRE
FDP_ITC.2/Installer	(FDP_ACC.1 or FDP_IFC.1) and (FPT_TDC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.2/CM, FTP_ITC.1/CM, FPT_TDC.1
FMT_SMR.1/Installer	(FIA_UID.1)	See note below
FPT_FLS.1/Installer	No Dependencies	
FPT_RCV.3/Installer	(AGD_OPE.1)	AGD_OPE.1
FDP_ACC.2/ADEL	(FDP_ACF.1)	FDP_ACF.1/ADEL
FDP_ACF.1/ADEL	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2/ADEL, FMT_MSA.3/ADEL
FDP_RIP.1/ADEL	No Dependencies	
FMT_MSA.1/ADEL	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/ADEL, FMT_SMF.1/ADEL, FMT_SMR.1/ADEL
FMT_MSA.3/ADEL	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/ADEL, FMT_SMR.1/ADEL
FMT_SMF.1/ADEL	No Dependencies	
FMT_SMF.1/OSU	No dependencies	
FMT_SMR.1/ADEL	(FIA_UID.1)	See note below
FPT_FLS.1/ADEL	No Dependencies	
FPT_FLS.1/OSU	No dependencies	
FDP_RIP.1/ODEL	No Dependencies	
FPT_FLS.1/ODEL	No Dependencies	
FCO_NRO.2/CM	(FIA_UID.1)	FIA_UID.1/CM
FDP_IFC.2/CM	(FDP_IFF.1)	FDP_IFF.1/CM
FDP_IFC.2/OSU	(FDP_IFF.1)	FDP_IFF.1/OSU
FDP_IFF.1/CM	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.2/CM, FMT_MSA.3/CM
FDP_UIT.1/CM	(FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.2/CM, FTP_ITC.1/CM
FIA_UID.1/CM	No Dependencies	
FIA_UID.1/OSU	No Dependencies	
FIA_UAU.1/OSU	(FIA_UID.1)	FIA_UID.1/OSU
FIA_UAU.4/OSU	No Dependencies	
FMT_MSA.1/CM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_IFC.2/CM, FMT_SMF.1/CM, FMT_SMR.1/CM
FMT_MSA.3/CM	(FMT_MSA.1) and	FMT_MSA.1/CM,

	(FMT_SMR.1)	FMT_SMR.1/CM
FMT_SMF.1/CM	No Dependencies	
FMT_SMR.1/CM	(FIA_UID.1)	FIA_UID.1/CM
FMT_SMR.1/OSU	(FIA_UID.1)	FIA_UID.1/OSU
FTP_ITC.1/CM	No Dependencies	
FDP_ACC.1/CMGR	FDP_ACF.1	FDP_ACF.1/CMGR
FDP_ACF.1/CMGR	FDP_ACC.1, FMT_MSA.3	FDP_ACC.1/CMGR, FMT_MSA.3/CMGR
FMT_MSA.1/CMGR	(FDP_ACC.1 or FDP_IFC.1), FMT_SMF.1, FMT_SMR.1	FDP_ACC.1/CMGR, FMT_SMF.1/CM, FMT_SMR.1/CM
FMT_MSA.1/OSU	(FDP_ACC.1 or FDP_IFC.1), FMT_SMF.1, FMT_SMR.1	FDP_IFC.2/OSU FMT_SMR.1/OSU FMT_SMF.1/OSU
FMT_MSA.3/CMGR	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/CMGR, FMT_SMR.1/CM
FMT_MSA.3/OSU	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/OSU FMT_SMR.1/OSU
FPT_RCV.4/SCP	No Dependencies	
FPT_TST.1/SCP	No Dependencies	
FPT_PHP.3/SCP	No Dependencies	
FCS_CKM.3	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5)	FCS_CKM.1
FCS_CKM.2	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5)	FCS_CKM.1

Table 38:SFRs Dependencies for Java Card System

Note:

The dependency FIA_UID.1 of FMT_SMR.1/Installer is discarded. This [PP JCS] does not require the identification of the "installer" since it can be considered as part of the TSF.

The dependency FIA_UID.1 of FMT_SMR.1/ADEL is discarded. This [PP JCS] does not require the identification of the "deletion manager" since it can be considered as part of the TSF.

The dependency FMT_SMF.1 of FMT_MSA.1/JCRE is discarded. The dependency between FMT_MSA.1/JCRE and FMT_SMF.1 is not satisfied because no management functions are required for the Java Card RE.

The dependency FAU_SAA.1 of FAU_ARP.1 is discarded. The dependency of FAU_ARP.1 on FAU_SAA.1 assumes that a "potential security violation" generates an audit event. On the

contrary, the events listed in FAU_ARP.1 are self-contained (arithmetic exception, ill-formed bytecodes, access failure) and ask for a straightforward reaction of the TSFs on their occurrence at runtime. The JCVM or other components of the TOE detect these events during their usual working order. Thus, there is no mandatory audit recording in [\[PP JCS\]](#).

7.2 Security Requirements for eUICC

The Security Functional Requirements for the eUICC component of the TOE are defined in strict compliance with the Security Functional Requirements described in the [\[PP eUICC\]](#).

7.2.1 Security Policies

The TOE of this security target defines the following security policies:

- Secure Channel Protocol information flow control SFP,
- Platform services information flow control SFP,
- ISD-R content access control SFP,
- ECASD access control SFP.

7.2.2 Security attributes used in SFRs

Security attribute	Details	Relationship to assets
AID	The AID is an identifier for the applications in a JCS runtime environment.	The AID belongs to the runtime environment (is an asset of the JCS Protection Profile [PP JCS])
S.ISD-P state	The state of the subject S.ISD-P. The possible value for this state are: ● ENABLED ● DISABLED ● INSTALLED ● SELECTABLE	This attribute is a part of the D.PLATFORM_DATA described in section 4.2.1
PPR	The Profile Policy Rules are associated to a given S.ISD-P and are used by the TOE to assess whether an ISD-P disabling or deletion is authorized. PPR may include one or several of the following rules: ● (PPR1) ‘Disabling of this Profile is not allowed’	This attribute is described as part of D.PROFILE_RULES in section 4.2.1

	(PPR2) ‘Deletion of this Profile is not allowed’	
Reference Enterprise Rule	The Reference Enterprise Rule is associated to a given S.ISD-P and is used by the TOE to assess whether other ISD-Ps enablement is authorized. If MEP is supported, it is possible that multiple Enterprise profiles are enabled at a time. Then, only the Reference Enterprise Rule is considered.	This attribute is described as part of D.PROFILE_RULES in section 4.2.1
Enterprise Rule	A rule stored in an Enterprise Profile that can be used by the Profile Owner to restrict End User controllability for enabling and installing Profiles on Enterprise Capable Devices.	This attribute is described as part of D.PROFILE_RULES in section 4.2.1
RAT	The Rules Authorisation Table is installed at eUICC personalization time and is used by the PPE and the LPA to determine whether or not a Profile that contains PPRs is authorised and can be installed on the eUICC.	
Keysets and session keys (D.MNO_KEYS, D.SECRETS)	Keysets are used by the TOE to build secure channels between remote actors and their local counterparts on the eUICC.	These attributes (D.MNO_KEYS, D.SECRETS) are defined in section 4.2.1
CERT.DPauth. E CDSA CERT.DPpb.EC DSA	Certificates of U.SM-DP+ that are used by the TOE to authenticate this user. These certificates are issued by the eSIM CA. The TOE can verify this certificate using the eSIM CA public key.	These attributes are not assets of this Protection Profile. The eSIM CA public key is described as the asset D.PK.CI.ECDSA in section 4.2.1
SM-DP+ OID MNO OID	SM-DP+ OID is the identification of the default SM-DP+. This value can be empty, in which case either the SM-DS discovery procedure or the SM-DP+ address contained in an Activation Code have to be used. The default SM-DP+ address can be modified or deleted during the lifetime of the eUICC;	These attributes included in the D.PLATFORM_DATA described in section 4.2.1

	Memory Reset resets the SM-DP+ OID to its initial value. MNO OID is the identification of the MNO owner of the Profile. Once this information is associated to the Profile, it remains unchanged during the Profile's life-time.	
CERT.DSauth.ECDSA	Certificates of U.SM-DS that are used by the TOE to authenticate this user. These certificates are issued by the eSIM CA. The TOE can verify this certificate using the eSIM CA public key.	These attributes are not assets of this Protection Profile. The eSIM CA public key is described as the asset D.PK.CI.ECDSA in section 4.2.1
SM-DS OID(s)	SM-DS OID is the identification Root SMDS. The Root SM-DS address(es) are unique and filled in the eUICC. The Root SM-DS(s) are configured at the time of Device manufacture and is invariant.	These attribute is included in the D.PLATFORM_DATA described in section 4.2.1
EID	The EID is the identifier of the physical eUICC on which the TOE is implemented.	The EID is a hardware identifier and is not part of the assets of this protection profile.

Table 39:Definition of the security attributes

7.2.3 Security Functional Requirements for eUICC

7.2.3.1 Identification and authentication

FIA_UID.1/EXT Timing of identification

FIA_UID.1.1/EXT The TSF shall allow

- application selection
- requesting data that identifies the eUICC
- [assignment: *none*].

on behalf of the user to be performed before the user is identified.

FIA_UID.1.2/EXT The TSF shall require each user to be successfully identified before

allowing any other TSF-mediated actions on behalf of that user.

Application Note:

This SFR is related to the identification of the following external (remote) users of the TOE:

- U.SM-DP+;
- U.MNO-OTA;

The identification of the only local user (U.MNO-SD) is addressed by the FIA_UID.1/MNO-SD SFR.

Application selection is authorized before identification since it may be required to provide the identification of the eUICC to the remote user.

FIA_UAU.1/EXT Timing of authentication

FIA_UAU.1.1/EXT The TSF shall allow

- **application selection**
- **requesting data that identifies the eUICC**
- **user identification**
- **[assignment: none]**

on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2/EXT The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Application Note:

This SFR is related to the authentication of the following external (remote) users of the TOE:

- U.SM-DP+;
- U.MNO-OTA;

FIA_USB.1/EXT User-subject binding

FIA_USB.1.1/EXT The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

- **SM-DP+ OID is associated to S.ISD-R, acting on behalf of U.SM-DP+;**
- **MNO OID is associated to U.MNO-SD, acting on behalf of U.MNO-OTA;**
- **SM-DS OID is associated to S.ISD-R, acting on behalf of U.SM-DS;**
- **[selection: no other associations].**

FIA_USB.1.2/EXT The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

- **Initial association of SM-DP+ OID and MNO OID requires U.SM-DP+ to be authenticated via “CERT.DPauth.ECDSA”;**
- **Initial association of SM-DS OID requires U.SM-DS to be authenticated via “CERT.DSauth.ECDSA”;**

- **[selection: *no other initial associations*].**

FIA_USB.1.3/EXT The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

- **change of SM-DP+ OID requires U.SM-DP+ to be authenticated via “CERT.DPauth.ECDSA”**
- **change of MNO OID is not allowed.**
- **change of SM-DS OID requires U.SM-DS to be authenticated via “CERT.DSauth.ECDSA”;**
- **[selection: *no other changes*].**

Application Note:

This SFR is related to the binding of external (remote) users to local subjects or users of the TOE:

- U.SM-DP+ binds to a subject (S.ISD-R);
- U.SM-DS binds to a subject (S.ISD-R)
- U.MNO-OTA binds to an on-card user (U.MNO-SD);

It should be noted that the U.MNO-SD is not a subject of the TOE, but an external on-card user acting on behalf of U.MNO-OTA, which is an external off-card user.

This SFR is related to the following commands:

- Initial association of the D.MNO_KEYS keyset is performed by the ES8+.ConfigureISDP command.

FIA_UAU.4/EXT Single-use authentication mechanisms

FIA_UAU.4.1/EXT The TSF shall prevent reuse of authentication data related to **the authentication mechanism used to open a secure communication channel between the eUICC and**

- **U.SM-DP+**
- **U.MNO-OTA**
- **[Selection: *none*]**

Application Note:

This SFR is related to the authentication of external (remote) users of the TOE:

- U.SM-DP+;
- U.MNO-OTA;

FIA_UID.1/MNO-SD Timing of identification

FIA_UID.1.1/MNO-SD The TSF shall allow **[assignment: *none*]** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2/MNO-SD The TSF shall require each user to be successfully identified before allowing

any other TSF-mediated actions on behalf of that user.

Application Note:

This SFR is related to the identification of the local user U.MNO-SD only. The identification of remote users is addressed by the FIA_UID.1/EXT SFR.

It should be noted that the U.MNO-SD is identified but not authenticated. However, U.MNO-SD is installed on the TOE by the U.SM-DP+ via the subject S.ISD-R (see FDP_ACF.1/ISDR), and the binding between U.SM-DP+ and S.ISD-R requires authentication of U.SM-DP+, as described in FIA_USB.1/EXT.

FIA_USB.1/MNO-SD User-subject binding

FIA_USB.1.1/MNO-SD The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **The AID is associated to the S.ISD-P acting on behalf of U.MNO-SD.**

FIA_USB.1.2/MNO-SD The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **Initial association of AID requires U.SM-DP+ to be authenticated via CERT.DPauth.ECDSA.**

FIA_USB.1.3/MNO-SD The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **no change of AID is allowed.**

Application Note:

This SFR is related to the identification of the local user U.MNO-SD.

FIA_ATD.1/Base User attribute definition

FIA_ATD.1.1/Base The TSF shall maintain the following list of security attributes belonging to individual users:

- **CERT.DPauth.ECDSA, CERT.DPpb.ECDSA, and SM-DP+ OID belonging to U.SM-DP+;**
- **MNO OID belonging to U.MNO-OTA;**
- **AID belonging to U.MNO-SD;**
- **CERT.DSauth.ECDSA and SM-DS OID belonging to U.SM-DS;**
- **[selection: no additional attributes].**

FIA_API.1 Authentication Proof of Identity

FIA_API.1.1 The TSF shall provide a **cryptographic authentication mechanism based on the EID of the eUICC** to prove the identity of the TOE by including the following properties **the EID value in the eUICC certificate** to an external entity.

Application Note:

This proof is obtained by including the EID value in the eUICC certificate, which is signed by the eUICC Manufacturer.

7.2.3.2 Communication

FDP_IFC.1/SCP Subset information flow control

FDP_IFC.1.1/SCP The TSF shall enforce the **Secure Channel Protocol information flow control SFP** on

- **users/subjects/objects:**
 - **U.SM-DP+ and SO.ISD-R, SO.ISD-P**
 - **U.MNO-OTA and U.MNO-SD**
- **information: transmission of commands.**

FDP_IFE.1/SCP Simple security attributes

FDP_IFE.1.1/SCP The TSF shall enforce the **Secure Channel Protocol information flow control SFP** based on the following types of subject and information security attributes:

- **users/subjects/objects:**
 - **U.SM-DP+, SO.ISD-P and SO.ISD-R, with security attribute D.SECRETS**
 - **U.MNO-OTA and U.MNO-SD, with security attribute D.MNO_KEYS**
- **information: transmission of commands.**

FDP_IFE.1.2/SCP The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **The TOE shall permit communication between U.MNO-OTA and U.MNOSD in a SCP80 or SCP81 secure channel.**

FDP_IFE.1.3/SCP The TSF shall enforce the [assignment: *none*].

FDP_IFE.1.4/SCP The TSF shall explicitly authorise an information flow based on the following rules:
[assignment: *none*].

FDP_IFE.1.5/SCP The TSF shall explicitly deny an information flow based on the following rules:

- **The TOE shall reject communication between U.SM-DP+ and S.ISD-R if it is not performed in a SCP-SGP22 secure channel.**

Application Note:

More details on the secure channels can be found in [\[SGP 22\]](#)

- For SM-DP+: [\[SGP 22\]](#) § 5.5
- For MNO-SD: [\[SGP 22\]](#) § 5.4

FTP_ITC.1/SCP Inter-TSF trusted channel
--

FTP_ITC.1.1/SCP The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/SCP The TSF shall permit another trusted IT product to initiate communication via the trusted channel.

FTP_ITC.1.3/SCP The TSF shall initiate communication via the trusted channel for **[assignment: The TSF shall permit the SM DP+ to open a SCP-SGP22 secure channel to transmit the following operations:**

- *ES8+.InitialiseSecureChannel*
- *ES8+.ConfigureISDP*
- *ES8+.StoreMetadata*
- *ES8+.ReplaceSessionKeys*
- *ES8+.LoadProfileElements.*

The TSF shall permit the LPAd/IPAd to transmit the following operations:

- *ES10a.GetEuiccConfiguredData (SGP.22 v3.1 or higher)*
- *ES10a.GetEuiccConfiguredAddresses (SGP.22 v2.x)*
- *ES10a.SetDefaultDpAddress (SGP.22)*
- *ES10b.PrepareDownload*
- *ES10b.LoadBoundProfilePackage*
- *ES10b.GetEUICCChallenge*
- *ES10b.GetEUICCInfo*
- *ES10b.ListNotification*
- *ES10b.RetrieveNotificationsList*
- *ES10b.RemoveNotificationFromList*
- *ES10b.LoadCRL (SGP.22 V2.x)*
- *ES10b.AuthenticateServer*
- *ES10b.CancelSession*
- *ES10c.GetProfilesInfo (SGP.22)*
- *ES10c.EnableProfile (SGP.22)*
- *ES10c.DisableProfile (SGP.22)*
- *ES10c.DeleteProfile (SGP.22)*
- *ES10c.eUICCMemoryReset (SGP.22)*
- *ES10c.GetEID (SGP.22)*
- *ES10c.SetNickname (SGP.22)*
- *ES10b.GetRAT*

- *ES10c.LPA alerting (SGP.22 v3.1 or higher)*
- *ES10c.VerifySmsResponse (SGP.22 v3.1 or higher)*
- *ES10b.LoadRPMPackage (SGP.22 v3.1 or higher)*
- *ES10b.PrepareDeviceChange (SGP.22 v3.1 or higher)*
- *ES10b.VerifyDeviceChange (SGP.22 v3.1 or higher)*

The TSF shall permit the remote OTA Platform to open a SCP80 or SCP81 secure channel to transmit the following operation:

- *ES6.UpdateMetadata.*
-].

Application Note:

Related keys are:

- either generated on-card (D.SECRETS);
- or distributed along with the profile (D.MNO_KEYS);

FDP_ITC.2/SCP Import of user data with security attributes

FDP_ITC.2.1/SCP The TSF shall enforce the **Secure Channel Protocol information flow control SFP** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/SCP The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/SCP The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/SCP The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/SCP The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [**assignment: none**].

FPT_TDC.1/SCP Inter-TSF basic TSF data consistency

FPT_TDC.1.1/SCP The TSF shall provide the capability to consistently interpret

- **Commands from U.SM-DP+ and U.MNO-OTA**
- **Downloaded objects from U.SM-DP+ and U.MNO-OTA**

when shared between the TSF and another trusted IT product.

FPT_TDC.1.2/SCP The TSF shall use [**assignment: none**] when interpreting the TSF data from another trusted IT product.

Application Note:

The commands related to the SFRs FPT_TDC.1/SCP, FDP_IFC.1/SCP, FDP_IFF.1/SCP and the Downloaded objects related to this SFR FPT_TDC.1/SCP are listed below:

- SM-DP+ commands
 - ES8+.InitialiseSecureChannel
 - ES8+.ConfigureISDP
 - ES8+.StoreMetadata
 - ES8+.ReplaceSessionKeys
 - ES8+.LoadProfileElements
- LPA/IPAd commands
 - ES10a.GetEuiccConfiguredData (SGP.22 v3.0 or higher)
 - ES10a.GetEuiccConfiguredAddresses (SGP.22 v2.x)
 - ES10a.SetDefaultDpAddress
 - ES10b.PrepareDownload
 - ES10b.LoadBoundProfilePackage
 - ES10b.GetEUICCCChallenge
 - ES10b.GetEUICCInfo
 - ES10b.ListNotification
 - ES10b.RetrieveNotificationsList
 - ES10b.RemoveNotificationFromList
 - ES10b.LoadCRL (SGP.22 V2.x)
 - ES10b.AuthenticateServer
 - ES10b.CancelSession
 - ES10c.GetProfilesInfo
 - ES10c.EnableProfile
 - ES10c.DisableProfile
 - ES10c.DeleteProfile
 - ES10c.eUICCMemoryReset (SGP.22)
 - ES10c.GetEID (SGP.22)
 - ES10c.SetNickname
 - ES10c.GetRAT
 - ES10c.LPA alerting (SGP.22 v3.1 or higher)
 - ES10c.VerifySmdsResponse (SGP.22 v3.1 or higher)
 - ES10b.LoadRPMPackage (SGP.22 v3.1 or higher)
 - ES10b.PrepareDeviceChange (SGP.22 v3.1 or higher)
 - ES10b.VerifyDeviceChange (SGP.22 v3.1 or higher)
- Downloaded objects from SM-DP+
 - Session keys
 - Profile Metadata (including PPR data)
- MNO commands
 - ES6.UpdateMetadata
- Downloaded objects from MNO OTA Platform
 - Profile Metadata (including PPR data and Enterprise Rules).

FDP_UCT.1/SCP Basic data exchange confidentiality

FDP_UCT.1.1/SCP The TSF shall enforce the **Secure Channel Protocol information flow control SFP** to receive user data in a manner protected from unauthorised disclosure.

Application Note:

This SFR is related to the protection of:

- Profiles downloaded from SM-DP+.

Related keys are:

- either generated on-card (D.SECRETS);
- or distributed along with the profile (D.MNO_KEYS).

FDP_UIT.1/SCP Data exchange integrity

FDP_UIT.1.1/SCP The TSF shall enforce the **Secure Channel Protocol information flow control SFP** to receive user data in a manner protected from modification, deletion, insertion and replay errors.

FDP_UIT.1.2/SCP The TSF shall be able to determine on receipt of user data, whether modification, deletion, insertion and replay has occurred.

Application Note:

This SFR is related to the protection of:

- Profiles downloaded from SM-DP+;
- Commands received from SM-DP+ and MNO OTA Platform;
- PPR and Enterprise Rules (optional) received from the MNO OTA Platform.

Related keys are:

- either generated on-card (D.SECRETS);
- or distributed along with the profile (D.MNO_KEYS).

FCS_CKM.1/SCP-SM Cryptographic key generation

FCS_CKM.1.1/SCP-SM The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **Elliptic Curves Key agreement (ECKA)** and specified cryptographic key sizes **256** that meet the following: [assignment: *ECKA-EG using one of the following standards:*

- *NISTP-256(FIPSPUB186-3DigitalSignatureStandard)*
- *brainpoolP256r1(BSITR-03111,Version1.11,RFC5639)*

].

Application Note:

This key generation mechanism is used to generate

- D.SECRETS keyset via the ES8+.InitialiseSecureChannel command, using the U.SM-DP+ public

key of PK.DP.ECKA

FCS_CKM.2/SCP-MNO Cryptographic key distribution

FCS_CKM.2.1/SCP-MNO The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [**assignment: *distribution method from SCP-SGP22 (SCP03)***] that meets the following: [**assignment: *[SGP.02] standard***].

Application Note:

This SFR is related to the distribution of

- D.MNO_KEYS during profile download.

Note: this SFR does not apply to the private keys loaded pre-issuance of the TOE (D.SK.EUICC.ECDSA).

FCS_CKM.6/SCP-SM Cryptographic key destruction

FCS_CKM.6.1/SCP-SM The TSF shall destroy **D.SECRETS, CERT.DPauth.ECDSA, CERT.DPpb.ECDSA, CERT.DSauth.ECDSA, D.CERT.EUICC.ECDSA, D.SK.EUICC.ECDSA and D.PK.CLECDSA** when [**selection: *no longer needed***].

FCS_CKM.6.2/SCP-SM The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1/SCP-SM in accordance with a specified cryptographic key destruction method [**assignment: *clean the buffer with random bytes***] that meets the following: [**assignment: *none***].

FCS_CKM.6/SCP-MNO Cryptographic key destruction

FCS_CKM.6.1/SCP-MNO The TSF shall destroy **D.MNO_KEYS** when [**selection: *no longer needed***].

FCS_CKM.6.2/SCP-MNO The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1/SCP-MNO in accordance with a specified cryptographic key destruction method [**assignment: *clean the buffer with random bytes***] that meets the following: [**assignment: *none***].

7.2.3.3 Security Domains

FDP_ACC.1/ISDR Subset access control

FDP_ACC.1.1/ISDR The TSF shall enforce the **ISD-R content access control SFP** on

- **subjects: S.ISD-R**
- **objects: SO.ISD-P**
- **operations:**

- Create and configure profile
- Store profile metadata
- Enable profile
- Disable profile
- Delete profile
- Perform a Memory reset.

Application Note:

This policy describes the rules to be applied to access Platform Management operations. It covers the access to operations by ISD-R required by [\[SGP 22\]](#), §5.x.

FDP_ACE1/ISDR Security attribute based access control
--

FDP_ACE1.1/ISDR The TSF shall enforce the **ISD-R content access control SFP** to objects based on the following:

- **subjects: S.ISD-R**
- **objects:**
 - SO.ISD-P with security attributes “state” “PPR”, and [Selection: *no additional attributes*]
- **operations:**
 - Create and configure profile
 - Store profile metadata
 - Enable profile
 - Disable profile
 - Delete profile
 - Perform a Memory reset

FDP_ACE1.2/ISDR The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **Authorized states:**

- **Enabling a S.ISD-P is authorized only if**
 - the corresponding S.ISD-P is in the state “DISABLED” and
 - in case a currently enabled S.ISD-P has to be disabled, the PPR data of this S.ISD-P allows its disabling, and
 - [Selection: *no additional conditions*].
- **Disabling a S.ISD-P is authorized only if**
 - the corresponding S.ISD-P is in the state “ENABLED” and
 - the corresponding S.ISD-P’s PPR data allows its disabling.
- **Deleting a S.ISD-P is authorized only if**
 - the corresponding S.ISD-P is not in the state “ENABLED” and
 - the corresponding S.ISD-P’s PPR data allows its deletion.
- **Performing a S.ISD-P Memory reset is authorized regardless of the involved S.ISD-P’s state or PPR attribute.**

FDP_ACF.1.3/ISDR The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment: none]**.

FDP_ACF.1.4/ISDR The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment: none]**.

Application Note:

This policy describes the rules to be applied to access Platform Management or eUICC Management operations. It covers the access to the following operations by ISD-R required by [\[SGP 22\]](#), §5.x:

- ES8+.ConfigureISDP (Create and configure profile)
- ES8+.StoreMetadata (Store profile metadata)
- ES10c.EnableProfile (Enable profile)
- ES10c.DisableProfile (Disable profile)
- ES10c.DeleteProfile (Delete profile)
- ES10c.eUICCMemoryReset (Perform a Memory reset)
- ES10b.LoadRpmPackage (Enable/Disable/Delete profile) (SGP.22 v3.1 or higher)

FDP_ACC.1/ECASD Subset access control
--

FDP_ACC.1.1/ECASD The TSF shall enforce the **ECASD access control SFP** on

- **subjects: S.ISD-R, S.ECASD**
- **objects: data and attributes of ECASD,**
- **operations:**
 - **execution of an ECASD function**
 - **access to output data of these functions,**
- **[assignment: none]**.

FDP_ACF.1/ECASD Security attribute based access control
--

FDP_ACF.1.1/ECASD The TSF shall enforce the **ECASD access control SFP** to objects based on the following:

- **subjects: S.ISD-R, with security attribute “AID”, S.ECASD**
- **objects: data and attributes of ECASD**
- **operations:**
 - **execution of a ECASD function**
 - **Verification of the off-card entities Certificates (SM-DP+, SM-DS), provided by an ISD-R, with the CI public key (D.PK.CIECDSA)**
 - **Creation of an eUICC signature on material provided by an ISD-R.**
 - **access to output data of these functions.**
- **[assignment: none]**.

FDP_ACF.1.2/ECASD The TSF shall enforce the following rules to determine if an operation among

controlled subjects and controlled objects is allowed:

- **Authorized users: only S.ISD-R, identified by its AID, shall be authorized to execute the following S.ECASD functions:**
 - **Verification of a certificate CERT.DPauth.ECDSA, CERT.DPpb.ECDSA, or CERT.DSauth.ECDSA provided by an ISD-R, with the eSIM CA public key (D.PK.CIECDSA)**
 - **Creation of an eUICC signature, using D.SK.EUICC.ECDSA, on material provided by an ISD-R.**
- **[assignment: none].**

FDP_ACF1.3/ECASD The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment: none]**.

FDP_ACF1.4/ECASD The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment: none]**.

7.2.3.4 Platform Services

FDP_IFC.1/Platform_services Subset information flow control
--

FDP_IFC.1.1/Platform_services The TSF shall enforce the **Platform services information flow control SFP** on

- **users/subjects:**
 - **S.ISD-R, S.ISD-P, U.MNO-SD**
 - **Platform code (S.PRE, S.PPI, S.TELECOM)**
- **information:**
 - **D.PROFILE_NAA_PARAMS**
 - **D.PROFILE_RULES**
 - **D.PLATFORM_RAT**
- **operations:**
 - **installation of a profile**
 - **PPR and RAT enforcement**
 - **network authentication.**
 - **[selection: no additional operations]**

FDP_IFE.1/Platform_services Simple security attributes

FDP_IFE.1.1/Platform_services The TSF shall enforce the **Platform services information flow control SFP** based on the following types of subject and information security attributes:

- **users/subjects:**
 - **S.ISD-R, S.ISD-P, U.MNO-SD, with security attribute “application identifier (AID)”**
 - **Platform code (S.PRE, S.PPI, S.TELECOM)**

- **information:**
 - **D.PROFILE_NAA_PARAMS**
 - **D.PROFILE_RULES**
 - **D.PLATFORM_RAT**
- **operations:**
 - **installation of a profile**
 - **PPR and RAT enforcement**
 - **network authentication.**
 - **[selection: *no additional operations*]**

FDP_IFE.1.2/Platform_services The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **D.PROFILE_NAA_PARAMS shall be transmitted only:**
 - **by U.MNO-SD to S.TELECOM in order to execute the network authentication function**
 - **by S.ISD-R to S.PPI using the profile installation function**
- **D.PROFILE_RULES shall be transmitted only**
 - **by S.ISD-R to S.PRE in order to execute the PPR enforcement function**
 - **[selection: *no additional information flows*]**
- **D.PLATFORM_RAT shall be transmitted only**
 - **by S.ISD-R to S.PRE in order to execute the RAT enforcement function.**

FDP_IFE.1.3/Platform_services The TSF shall enforce the **[assignment: *none*]**.

FDP_IFE.1.4/Platform_services The TSF shall explicitly authorise an information flow based on the following rules: **[assignment: *none*]**.

FDP_IFE.1.5/Platform_services The TSF shall explicitly deny an information flow based on the following rules: **[assignment: *none*]**.

Application Note:

This SFR aims to control which subject is able to transmit Profile Policy Rules, Enterprise Rules, Rules Authorisation Table or network authentication keys to the PRE, PPI, and Telecom Framework. Differences in implementation are allowed, since this PP requires demonstrable conformance. It is consequently possible for the ST writer to replace this SFR by another instance of FDP_IFE.1 as long as it addresses the control of information flow for these data. Examples of such adaptations could be due to cases such as:

- D.PROFILE_RULES transmitted from S.ISD-P to S.ISD-R, then from S.ISD-R to S.PRE;
- D.PROFILE_NAA_PARAMS transmitted from U.MNO-SD to S.ISD-P, then by S.ISD-P to S.TELECOM.

FPT_FLS.1/Platform_services Failure with preservation of secure state
--

FPT_FLS.1.1/Platform_services The TSF shall preserve a secure state when the following types of failures occur:

- **failure that lead to a potential security violation during the processing of a S.PRE, S.PPI or S.TELECOM API specific functions:**
 - **Installation of a profile**
 - **PPR and RAT enforcement**
 - **Network authentication**
 - **[selection: *no additional functions*]**
- **[assignment: *none*].**

7.2.3.5 Security Management

FPT_EMS.1/Base TOE Emanation of TSF and User data

FPT_EMS.1.1/Base The TSF shall ensure that the TOE does not emit emissions over its attack surface in such amount that these emissions enable access to TSF data and user data as specified in <table>

ID	Emission	Attack surface	TSF data	User data
1	side channels (power consumptions, the timing of transitions of internal states and electromagnetic fluctuations)	Any	-	• D.SECRETS; • D.SK.EUICC.ECDSA and the secret keys which are part of the following keysets: • D.MNO_KEYS, • D.PROFILE_NAA_PARAMS.

Table 40:emissions enable access to TSF data and user data

Application Note:

The TOE shall prevent attacks against the secret data of the TOE where the attack is based on external observable physical phenomena of the TOE. Such attacks may be observable at the interfaces of the TOE or may originate from internal operation of the TOE or may originate from an attacker that varies the physical environment under which the TOE operates. The set of measurable physical phenomena is influenced by the technology employed to implement the TOE.

Examples of measurable phenomena are variations in the power consumption, the timing of transitions of internal states, electromagnetic radiation due to internal operation, radio emission. Due to the heterogeneous nature of the technologies that may cause such emanations, evaluation against state-of-the-art attacks applicable to the technologies employed by the TOE is assumed. Examples of such attacks are, but are not limited to, evaluation of TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, and so on.

FDP_SDI.1/Base Stored data integrity monitoring

FDP_SDI.1.1/Base The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors** on all objects, based on the following attributes: **integrity-sensitive data**.

Refinement:

The notion of integrity-sensitive data covers the assets of the Security Target TOE that require to be protected against unauthorized modification, including but not limited to the assets of this PP that require to be protected against unauthorized modification:

- D.MNO_KEYS
- Profile data
 - D.PROFILE_NAA_PARAMS
 - D.PROFILE_IDENTITY
 - D.PROFILE_RULES
 - D.PROFILE_USER_CODES
- Management data
 - D.PLATFORM_DATA
 - D.DEVICE_INFO
 - D.PLATFORM_RAT
- Identity management data
 - D.SK.EUICC.ECDSA
 - D.CERT.EUICC.ECDSA
 - D.PK.CI.ECDSA
 - D.EID
 - D.SECRETS
 - D.CERT.EUM.ECDSA
 - D.CRLs if existing

FDP_RIP.1/Base Subset residual information protection

FDP_RIP.1.1/Base The TSF shall ensure that any previous information content of a resource is made unavailable upon the **deallocation of the resource from and allocation of the resource** to the following objects:

- **D.SECRETS;**
- **D.SK.EUICC.ECDSA;**
- **The secret keys which are part of the following keysets:**
 - **D.MNO_KEYS,**
 - **D.PROFILE_NAA_PARAMS.**

FPT_FLS.1/Base Failure with preservation of secure state

FPT_FLS.1.1/Base The TSF shall preserve a secure state when the following types of failures occur:

- failure of creation of a new ISD-P by ISD-R
- failure of installation of a profile by ISD-R.

FMT_MSA.1/PLATFORM_DATA Management of security attributes

FMT_MSA.1.1/PLATFORM_DATA The TSF shall enforce the **ISD-R content access control policy** to restrict the ability to modify the security attributes **the following parts of D.PLATFORM_DATA** of:

- **ISD-P state**

to

- **S.ISD-R.**

Application Note:

- The following changes are possible:
 - Modify ISD-P state
 - from “INSTALLED” to “SELECTABLE” (during ISD-P creation)
 - from “ENABLED” to “DISABLED” (during profile disabling)
 - Modify ISD-P state
 - from “DISABLED” to “ENABLED” (during profile enabling)
- In case part of the Platform functionality is performed by GlobalPlatform packages, the role of S.PRE may for instance be partly attributed to the OPEN.

FMT_MSA.1/RULES Management of security attributes

FMT_MSA.1.1/RULES The TSF shall enforce the **Secure Channel protocol information flow SFP** to restrict the ability to change default, query, modify and delete the security attributes

- **D.PROFILE_RULES**

to

- **S.ISD-R for change_default, via function “ES8+.ConfigureISDP”**
- **S.ISD-R for query**
- **S.ISD-P for modify, via function “ES6.UpdateMetadata”**
- [selection:
 - *S.ISD-R to modify, via function “ES10b.LoadRPMPackage (UpdateMetadataRequest)” (SGP.22 v3.1 or higher),*
 - *S.ISD-R to delete, via function “ES10c.DeleteProfile”*].

FMT_MSA.1/CERT_KEYS Management of security attributes

FMT_MSA.1.1/CERT_KEYS The TSF shall enforce the **ECASD access control SFP** to restrict the ability to query and delete the security attributes

- **D.CERT.EUICC.ECDSA**
- **D.PK.CI.ECDSA**
- **D.CERT.EUM.ECDSA**

- **D.MNO_KEYS**
- to
- **S.ISD-R for:**
 - **query D.PK.CI.ECDSA**
 - **delete D.MNO_KEYS, via function [selection: *ES10c.DeleteProfile(SGP.22)*]**
 - **no actor for other operations.**

Application Note:

The modification of D.MNO_KEYS keysets is forbidden. To modify the keysets, one must delete the profile and load another profile.

FMT_SME.1/Base Specification of Management Functions

FMT_SME.1.1/Base The TSF shall be capable of performing the following management functions:
[assignment:

- *SCP information flow control (linked to roles S.ISD-R, U.SM-DP+, S.ISD-P, U.MNO-SD, U.MNO-OTA)*
- *Platform services information flow control (linked to roles S.PPI, S.ISD-P, S.ISD-R, U.MNO-SD)*
- *ISD R access control (linked to role S.ISD-R, U.SM-DP+)*
- *ISD P content access control (linked to roles S.ISD-P, U.MNO-SD, U.MNO-OTA)*
- *ECASD access control (linked to roles S.ECASD)*

].

FMT_SMR.1/Base Security roles

FMT_SMR.1.1/Base The TSF shall maintain the roles

- **External users:**
 - **U.SM-DP+**
 - **U.MNO-SD**
 - **U.MNO-OTA**
 - **U.SM-DS**
 - **[selection: *none*]**
- **Subjects:**
 - **S.ISD-R**
 - **S.ISD-P**
 - **S.ECASD**
 - **S.PPI**
 - **S.PRE**
 - **S.TELECOM.**

FMT_SMR.1.2/Base The TSF shall be able to associate users with roles.

Application Note:

The roles defined here correspond to the users and subjects defined in [\[PP eUICC\]](#) §3.2.

FMT_MSA.1/RAT Management of security attributes

FMT_MSA.1.1/RAT The TSF shall enforce the **Platform services information flow SFP** to restrict the ability to query the security attributes

- **D.PLATFORM_RAT**
- **D.PROFILE_NAA_PARAMS**
- **D.PROFILE_RULES**

to

- **S.ISD-R for query**
- **S.PRE for query**

FMT_MSA.3 Static attribute initialisation

FMT_MSA.3.1 The TSF shall enforce the **Security Channel Protocol information flow control SFP**, **ISD-R content access control SFP**, **ECASD access control SFP** and **Platform services information flow control SFP** to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the **no actor** to specify alternative initial values to override the default values when an object or information is created.

7.2.3.6 Mobile Network authentication

FCS_COP.1/Mobile_network Cryptographic operation

FCS_COP.1.1/Mobile_network The TSF shall perform **Network authentication** in accordance with a specified cryptographic algorithm **MILENAGE**, **Tuak**, [selection: *[assignment: no other algorithm]*] and cryptographic key sizes **according to the corresponding standard** that meet the following:

- **MILENAGE according to standard [\[MILENAGE\]](#) with the following restrictions:**
 - **Only use 128-bit AES as the kernel function. Do not support other choices**
 - **Allow any value for the constant OP**
 - **Allow any value for the constants C1-C5 and R1-R5, subject to the rules and recommendations in section 5.3 of the standard [\[MILENAGE\]](#)**
- **Tuak according to [\[Tuak\]](#) with the following restrictions:**
 - **Allow any value of TOP**
 - **Allow multiple iterations of Keccak**
 - **Support 256-bit K as well as 128-bit**
 - **Restrict supported sizes for RES, MAC, CK and IK to those currently supported in 3GPP**

standards.

- [selection: *no additional standards*]

FCS_CKM.2/Mobile_network Cryptographic key distribution

FCS_CKM.2.1/Mobile_network The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: Network Authentication keys as part of the D.PROFILE_NAA_PARAMS loaded during profile installation] that meets the following: [assignment: [\[SGP 02\]](#), [\[SGP 21\]](#), [\[SGP 22\]](#)].

Application Note:

The keys in this SFR are the Mobile Network authentication keys included in the asset D.PROFILE_NAA_PARAMS. These keys are distributed as a part of the MNO profile during profile download.

FCS_CKM.6/Mobile_network Cryptographic key destruction

FCS_CKM.6.1/Mobile_network The TSF shall destroy **MILENAGE keys, TUA keys and** [selection: *no other keys of the cryptographic algorithm*] when [selection: *no longer needed*].

FCS_CKM.6.2/Mobile_network The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1/Mobile_network in accordance with a specified cryptographic key destruction method [assignment: *clean the buffer with random bytes*] that meets the following: [assignment: *none*].

7.2.4 Security Requirements Rationale for eUICC

The rationale of Security Functional Requirements for the eUICC component is strictly the same as in the [\[PP eUICC\]](#).

7.2.5 Dependencies for eUICC

Requirements	CC Dependencies	Satisfied Dependencies
FIA_UID.1/EXT	No Dependencies	
FIA_UAU.1/EXT	(FIA_UID.1)	FIA_UID.1/EXT
FIA_USB.1/EXT	(FIA_ATD.1)	FIA_ATD.1/Base
FIA_UAU.4/EXT	No Dependencies	

FIA_UID.1/MNO-SD	No Dependencies	
FIA_USB.1/MNO-SD	(FIA_ATD.1)	FIA_ATD.1/Base
FIA_ATD.1/Base	No Dependencies	
FIA_API.1	No Dependencies	
FDP_IFC.1/SCP	(FDP_IFF.1)	FDP_IFF.1/SCP
FDP_IFF.1/SCP	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1/SCP, FMT_MSA.3
FTP_ITC.1/SCP	No Dependencies	
FDP_ITC.2/SCP	(FDP_ACC.1 or FDP_IFC.1) and (FPT_TDC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.1/SCP, FTP_ITC.1/SCP, FPT_TDC.1/SCP
FPT_TDC.1/SCP	No Dependencies	
FDP_UCT.1/SCP	(FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.1/SCP, FTP_ITC.1/SCP
FDP_UIT.1/SCP	(FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.1/SCP, FTP_ITC.1/SCP
FCS_CKM.1/SCP-SM	(FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1) and (FCS_RBG.1	FCS_COP.1 FCS_CKM.6/SCP-SM FCS_RNG.1

	or FCS_RNG.1) and (FCS_CKM.6)	
FCS_CKM.2/SCP-MNO	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.5)	FDP_ITC.2/SCP,
FCS_CKM.6/SCP-SM	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) or FCS_CKM.5)	FDP_ITC.2/SCP
FCS_CKM.6/SCP-MNO	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) or FCS_CKM.5)	FDP_ITC.2/SCP
FDP_ACC.1/ISDR	(FDP_ACF.1)	FDP_ACF.1/ISDR
FDP_ACF.1/ISDR	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.1/ISDR, FMT_MSA.3
FDP_ACC.1/ECASD	(FDP_ACF.1)	FDP_ACF.1/ECASD
FDP_ACF.1/ECASD	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.1/ECASD, FMT_MSA.3
FDP_IFC.1/Platform_services	(FDP_IFF.1)	FDP_IFF.1/Platform_services
FDP_IFF.1/Platform_services	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1/Platform_services, FMT_MSA.3
FPT_FLS.1/Platform_services	No Dependencies	
FPT_EMS.1/Base	No Dependencies	
FDP_SDI.1/Base	No Dependencies	

FDP_RIP.1/Base	No Dependencies	
FPT_FLS.1/Base	No Dependencies	
FMT_MSA.1/PLATFORM_DATA	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.1/ISDR, FMT_SMF.1/Base, FMT_SMR.1/Base
FMT_MSA.1/RULES	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_IFC.1/SCP, FMT_SMF.1/Base, FMT_SMR.1/Base
FMT_MSA.1/CERT_KEYS	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.1/ECASD, FMT_SMF.1/Base, FMT_SMR.1/Base
FMT_SMF.1/Base	No Dependencies	
FMT_SMR.1/Base	(FIA_UID.1)	FIA_UID.1/EXT, FIA_UID.1/MNO-SD
FMT_MSA.1/RAT	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_IFC.1/Platform_services, FMT_SMF.1/Base, FMT_SMR.1/Base
FMT_MSA.3	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/PLATFORM_DATA, FMT_MSA.1/RULES, FMT_MSA.1/CERT_KEYS, FMT_SMR.1/Base, FMT_MSA.1/RAT

FCS_COP.1/Mobile_network	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) or FCS_CKM.5) and FCS_CKM.6	FDP_ITC.2/SCP FCS_CKM.6/Mobile_network
FCS_CKM.2/Mobile_network	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5)	FDP_ITC.2/SCP
FCS_CKM.6/Mobile_network	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) or FCS_CKM.5)	FDP_ITC.2/SCP

Table 41: SFRs Dependencies for eUICC

8 Security Assurance Requirements

This Security target conforms to the assurance package EAL4 augmented with ALC_DVS.2 and AVA_VAN.5.

8.1 SARs Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
ADV_ARC.1	(ADV_FSP.1) and (ADV_TDS.1)	ADV_FSP.4, ADV_TDS.3
ADV_FSP.4	(ADV_TDS.1)	ADV_TDS.3
ADV_IMP.1	(ADV_TDS.3) and (ALC_TAT.1)	ADV_TDS.3, ALC_TAT.1
ADV_TDS.3	(ADV_FSP.4)	ADV_FSP.4
AGD_OPE.1	(ADV_FSP.1)	ADV_FSP.4
AGD_PRE.1	No Dependencies	
ALC_CMC.4	(ALC_CMS.1) and (ALC_DVS.1) and (ALC_LCD.1)	ALC_CMS.4, ALC_DVS.2, ALC_LCD.1
ALC_CMS.4	No Dependencies	
ALC_DEL.1	No Dependencies	
ALC_DVS.2	No Dependencies	
ALC_LCD.1	No Dependencies	
ALC_TAT.1	(ADV_IMP.1)	ADV_IMP.1
ASE_CCL.1	(ASE_ECD.1) and (ASE_INT.1) and (ASE_REQ.1)	ASE_ECD.1, ASE_INT.1, ASE_REQ.2
ASE_ECD.1	No Dependencies	
ASE_INT.1	No Dependencies	
ASE_OBJ.2	(ASE_SPD.1)	ASE_SPD.1
ASE_REQ.2	(ASE_ECD.1) and (ASE_OBJ.2)	ASE_ECD.1, ASE_OBJ.2
ASE_SPD.1	No Dependencies	
ASE_TSS.1	(ADV_FSP.1) and (ASE_INT.1) and (ASE_REQ.1)	ADV_FSP.4, ASE_INT.1, ASE_REQ.2
ATE_COV.2	(ADV_FSP.2) and (ATE_FUN.1)	ADV_FSP.4, ATE_FUN.1
ATE_DPT.1	(ADV_ARC.1) and (ADV_TDS.2) and (ATE_FUN.1)	ADV_ARC.1, ADV_TDS.3, ATE_FUN.1
ATE_FUN.1	(ATE_COV.1)	ATE_COV.2
ATE_IND.2	(ADV_FSP.2) and (AGD_OPE.1)	ADV_FSP.4,

	and (AGD_PRE.1) and (ATE_COV.1) and (ATE_FUN.1)	AGD_OPE.1, AGD_PRE.1, ATE_COV.2, ATE_FUN.1
AVA_VAN.5	(ADV_ARC.1) and (ADV_FSP.4) and (ADV_IMP.1) and (ADV_TDS.3) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_DPT.1)	ADV_ARC.1, ADV_FSP.4, ADV_IMP.1, ADV_TDS.3, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1

Table 42: SARs Dependencies

8.2 Rationale for the Security Assurance Requirements

EAL4 is required for this type of TOE and product since it is intended to defend against sophisticated attacks. This evaluation assurance level allows a developer to gain maximum assurance from positive security engineering based on good practices. EAL4 represents the highest practical level of assurance expected for a commercial grade product. In order to provide a meaningful level of assurance that the TOE and its embedding product provide an adequate level of defense against such attacks: the evaluators should have access to the low level design and source code. The lowest for which such access is required is EAL4

8.2.1 ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE and the embedding product. Due to the nature of the TOE and embedding product, ALC_DVS.2 is necessary to justify the sufficiency of these procedures to protect their confidentiality and integrity. ALC_DVS.2 has no dependencies.

8.2.2 AVA_VAN.5 Advanced methodical vulnerability analysis

The TOE is intended to operate in hostile environments. AVA_VAN.5 "Advanced methodical vulnerability analysis" is considered as the expected level for Java Card technology-based products hosting sensitive applications, in particular in payment and identity areas. AVA_VAN.5 has dependencies on ADV_ARC.1, ADV_FSP.1, ADV_TDS.3, ADV_IMP.1, AGD_PRE.1 and AGD_OPE.1. All of them are satisfied by EAL4.

9 TOE Summary Specification

9.1 TOE Security Functionality

The table below shows the mapping between security functions and SFRs for both JavaCard and eUICC.

Security Functions	SFRs
SF.FW	FDP_ACC.2/FIREWALL,FDP_ACF.1/FIREWALL, FDP_IFC.1/JCVM,FDP_IFF.1/JCVM, FDP_RIP.1/OBJECTS,FMT_MSA.1/JCRE, FMT_MSA.1/JCVM,FMT_MSA.2/FIREWALL_JCVM, FMT_MSA.3/FIREWALL,FMT_MSA.3/JCVM, FMT_SMF.1/JCS,FMT_SMR.1/JCS
SF.API	FCS_CKM.1,FCS_CKM.6,FCS_COP.1, FDP_RIP.1/ABORT,FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray,FDP_RIP.1/bArray, FDP_RIP.1/KEYS,FDP_RIP.1/TRANSIENT, FDP_ROL.1/FIREWALL,FCS_CKM.2,FCS_CKM.3
SF.CSM	FAU_ARP.1,FDP_SDI.2/DATA,FPR_UNO.1, FPT_FLS.1/JCS,FPT_TDC.1
SF.AID	FIA_ATD.1/AID,FIA_UID.2/AID,FIA_USB.1/AID, FMT_MTD.1/JCRE,FMT_MTD.3/JCRE
SF.INST	FDP_ITC.2/Installer,FMT_SMR.1/Installer, FPT_FLS.1/Installer,FPT_RCV.3/Installer
SF.ADEL	FDP_ACC.2/ADEL,FDP_ACF.1.1/ADEL, FDP_RIP.1/ADEL,FMT_MSA.1/ADEL, FMT_MSA.3/ADEL,FMT_SMF.1/ADEL, FMT_SMR.1/ADEL,FPT_FLS.1/ADEL
SF.ODEL	FDP_RIP.1/ODEL,FPT_FLS.1/ODEL
SF.CAR	FCO_NRO.2/CM,FDP_IFC.2/CM,FDP_IFF.1/CM, FDP_UIT.1/CM,FIA_UID.1/CM,FMT_MSA.1/CM, FMT_MSA.3/CM,FMT_SMF.1/CM,FMT_SMR.1/CM, FTP_ITC.1/CM
SF.SCP	FPT_TST.1/SCP,FPT_PHP.3/SCP,FPT_RCV.4/SCP, FCS_RNG.1
SF.CM	FDP_ACC.1/CMGR,FDP_ACF.1/CMGR, FMT_MSA.1/CMGR,FMT_MSA.3/CMGR
SF.OSU	FDP_IFC.2/OSU,FDP_IFF.1/OSU,FMT_MSA.3/OSU, FMT_MSA.1/OSU,FMT_SMR.1/OSU,FMT_SMF.1/OSU, FIA_UID.1/OSU,FIA_UAU.1/OSU,FIA_UAU.4/OSU,

	FPT_FLS.1/OSU
SF.PLATFORM_EUICC	FIA_UID.1/EXT,FIA_UAU.1/EXT,FIA_USB.1/EXT, FIA_UAU.4/EXT,FIA_UID.1/MNO-SD, FIA_USB.1/MNO-SD,FIA_ATD.1,FIA_API.1, FDP_ACC.1/ISDR,FDP_ACF.1/ISDR,FDP_ACC.1/ECASD, FDP_ACF.1/ECASD,FDP_IFC.1/Platform_services, FDP_IFF.1/Platform_services,FPT_FLS.1/Platform_services, FPT_EMS.1/Base,FDP_SDI.1/Base,FDP_RIP.1/Base, FMT_MSA.1/PLATFORM_DATA,FMT_MSA.1/RULES, FMT_MSA.1/CERT_KEYS,FMT_SMF.1/Base,FMT_SMR.1/Base, FMT_MSA.1/RAT,FMT_MSA.3,FPT_FLS.1/Base
SF.COMM_EUICC	FDP_IFC.1/SCP,FDP_IFF.1/SCP,FPT_ITC.1/SCP, FDP_ITC.2/SCP,FPT_TDC.1/SCP,FDP_UCT.1/SCP, FDP_UIT.1/SCP,FCS_CKM.1/SCP-SM, FCS_CKM.2/SCP-MNO,FCS_CKM.6/SCP-SM, FCS_CKM.6/SCP-MNO
SF.NETWORK	FCS_COP.1/Mobile_network,FCS_CKM.2/Mobile_network, FCS_CKM.6/ Mobile_network

Table 43: Security Functions and SFRs

9.1.1 SFs For Java Card System

9.1.1.1 SF.FW: Firewall Policy

The SF.FW provides the Firewall Security Function which enforces firewall policy to ensure applet isolation and object sharing. Isolation means that one applet cannot access the fields or objects of an applet in another context unless the other applet explicitly provides an interface for access. To be more specific, firewalls essentially partition the Java Card platform's object system into separate protected object spaces called contexts. The firewall is the boundary between one context and another. The Java Card RE shall allocate and manage a context for each Java Card CAP File containing applets. All applet instances within a single Java Card CAP File shares the same context. There is no firewall between individual applet instances within the same CAP File. That is, an applet instance can freely access objects belonging to another applet instance that resides in any package in the same Java Card CAP File.

The following SFR are fulfilled by SF.FW:

- FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL
- FDP_IFC.1/JCVM and FDP_IFF.1/JCVM
- FDP_RIP.1/OBJECTS
- FMT_MSA.1/JCRE
- FMT_MSA.1/JCVM
- FMT_MSA.2/FIREWALL_JCVM

- FMT_MSA.3/FIREWALL
- FMT_MSA.3/JCVM
- FMT_SMF.1/JCS
- FMT_SMR.1/JCS

9.1.1.2 SF.API: Application Programming Interface

The SF.API Security Function provides cryptographic operations and key management according to [\[JCAPI\]](#) and [\[GP\]](#). It also ensures that the resource is made unavailable upon the deallocation of that resource according to [\[JCRE\]](#).

The following key generation algorithms, key sizes and standards are supported:

Iteration	Algorithm	Key size	Standards
ECC	ECC	ansix9p224r1, ansix9p256r1, ansix9p384r1, ansix9p521r1, brainpoolP224r1, brainpoolP256r1, brainpoolP320r1, brainpoolP384r1 and brainpoolP512r1 curves	ANSI X9.62-2005, RFC 5639
TDES	GP SCP02 session key	112, 168 bit(for SCP02)	[GP] (for SCP02)
AES	GP SCP03 session key	128,192, 256 bit(for SCP03)	[GP] (for SCP03)

The following cryptographic operations with supported algorithms, key sizes and standards are listed below:

Iteration	Operation	Algorithm	Key Size	Standards
TDES	Encryption and Decryption	ALG_DES_CBC_ISO9797_M1 ALG_DES_CBC_ISO9797_M2 ALG_DES_CBC_NOPAD ALG_DES_CBC_PKCS5 ALG_DES_ECB_ISO9797_M1 ALG_DES_ECB_ISO9797_M2 ALG_DES_ECB_NOPAD ALG_DES_ECB_PKCS5	LENGTH_DES3_2KEY,	NIST-SP800-67 ISO9797-1 DES NOPAD DES PKCS#5 DES 9797 M1 M2
TDES_MA C	Generation and Verification	ALG_DES_MAC4_ISO9797_1_M1_ALG3 ALG_DES_MAC4_ISO9797_1_M2_ALG3	LENGTH_DES3_2KEY,	[JCAPI]

		ALG_DES_MAC4_ISO9797_M1 ALG_DES_MAC4_ISO9797_M2 ALG_DES_MAC8_ISO9797_1_M1_ALG3 ALG_DES_MAC8_ISO9797_1_M2_ALG3 ALG_DES_MAC8_ISO9797_M1 ALG_DES_MAC8_ISO9797_M2 ALG_DES_MAC4_NOPAD ALG_DES_MAC8_NOPAD ALG_DES_MAC4_PKCS5 ALG_DES_MAC8_PKCS5		
ECDSA	Signature and Verification	ALG_ECDSA_SHA_256	256 bits	[JCAPI]
AES_CIPHER	Encryption and Decryption	ALG_AES_BLOCK_128_CBC_NOPAD ALG_AES_CBC_ISO9797_M1 ALG_AES_CBC_ISO9797_M2 ALG_AES_CBC_PKCS5 ALG_AES_BLOCK_128_ECB_NOPAD ALG_AES_ECB_ISO9797_M1 ALG_AES_ECB_ISO9797_M2 ALG_AES_ECB_PKCS5	128, 192, 256 bits	[JCAPI]
AES_MAC	Signature and Verification	ALG_AES_CMAC_128 ALG_AES_MAC_128_NOPAD ALG_AES_MAC_192_NOPAD ALG_AES_MAC_256_NOPAD	128, 256 bits	[JCAPI]
ECDH	Key Agreement	ALG_EC_SVDP_DH ALG_EC_SVDP_DHC ALG_EC_SVDP_DH_PLAIN ALG_EC_SVDP_DHC_PLAIN ALG_EC_SVDP_DH_PLAIN_XY	-	[JCAPI]
SHA	Secure Hash Computation	SHA-256	256 bits	[JCAPI]
HAMC	Signature	HAMC	256 bits	FIPS 198 The Keyed-Hash Message Authentication Code(HMAC)

The following cryptographic key distribution with supported algorithms, key distribution method and standards are listed below:

Iteration	Distribution method	Standard
-----------	---------------------	----------

TDES	JC API setKey()	[JCAPI]
AES	JC API setKey()	[JCAPI]

The following cryptographic key access with supported algorithms, key access method and standards are listed below:

Iteration	Key access method	Standard
TDES	JC API getKey()	[JCAPI]
AES	JC API getKey()	[JCAPI]
ECC	JC API: getA(), getB(), getG(), getR(), getK(),getS(),getW()	[JCAPI]

The following SFR are fulfilled by SF.API:

- FCS_CKM.1
- FCS_CKM.6
- FCS_COP.1
- FDP_RIP.1/ABORT
- FDP_RIP.1/APDU
- FDP_RIP.1/GlobalArray
- FDP_RIP.1/bArray
- FDP_RIP.1/KEYS
- FDP_RIP.1/TRANSIENT
- FDP_ROL.1/FIREWALL
- FCS_CKM.2
- FCS_CKM.3

9.1.1.3 SF.CSM: Card Security Management

The Card Security Management Security Function ensures the security policy for card content management, lifecycle management and the related cryptographic operations according to [\[JCVM\]](#) and [\[GP\]](#).

Upon detection of a potential security violation, one of the following actions is invoked:

- throw an exception,
- lock the card session,
- reinitialize the Java Card System and its data.

The "potential security violation" stands for one of the following events:

- CAP file inconsistency,
- typing error in the operands of a bytecode,
- applet life cycle inconsistency,
- card manager life cycle inconsistency,
- card tearing (unexpected removal of the Card out of the CAD) and power failure,
- abort of a transaction in an unexpected context,
- violation of the Firewall or JCVM SFPs,
- unavailability of resources,
- array overflow,
- integrity error on sensitive data including cryptographic key and PIN.

The THN31 IC provides hardware security mechanisms including Bus Masking, Data Masking and dummy operation to ensure that it is unable to observe the cryptographic key values and PIN values in terms of cryptographic operation and comparison.

The following SFR are fulfilled by SF.CSM:

- FAU_ARP.1
- FDP_SDL.2/DATA
- FPR_UNO.1
- FPT_FLS.1/JCS
- FPT_TDC.1

9.1.1.4 SF.AID: AID Management

The AID Security Function enforces rules for the AID as defined in [\[JCRE\]](#) with respect to the following security attributes:

- CAP File AID,
- Package AID,
- Applet's version number,
- Registered applet AID,
- Applet Selection Status.

Each user is identified before allowing any other actions on behalf of that user. Upon applet selection, the rules for initial selection and applet selection are enforced according to [\[JCRE\]](#), §4. Furthermore, the modification on the AID list that is maintained by JCRE is restricted.

The following SFR are fulfilled by SF.AID:

- FIA_ATD.1/AID
- FIA_UID.2/AID

- FIA_USB.1/AID
- FMT_MTD.1/JCRE
- FMT_MTD.3/JCRE

9.1.1.5 SF.INST: Installer

The Installer Security Function enforces rules for loading and installation of a package and applet respectively as defined in [\[JCVM\]](#).

CAP file loading is allowed only if, for each dependent package, its AID is equal to a resident package AID, the major version associated to the dependent package file is equal to the major version of the resident package and the minor version is equal to or less than the minor version associated to the resident package ([\[JCVM\]](#), §4.5.2).

For failures during load / installation of a package/applet and deletion of a package / applet / object, SF.INST ensures the return of the TOE to a secure state using automated procedures.

The SF.INST fulfils the following SFR:

- FDP_ITC.2/Installer
- FMT_SMR.1/Installer
- FPT_FLS.1/Installer
- FPT_RCV.3/Installer

9.1.1.6 SF.ADEL: Applet Deletion

The ADEL Security Function enforces rules for applet deletion and resource management after the applet deletion as defined in [\[JCVM\]](#).

The Cap file and applet can be deleted only if the operation meets the rules according to [\[JCVM\]](#). Once deleted, any previous information content of a resource is made unavailable upon the deallocation of the resource from the deleted objects.

A secure state is preserved when the applet deletion manager fails to delete a CAP file/applet as described in [\[JCRE\]](#), §11.3.4.

The following SFR are fulfilled by SF.ADEL:

- FDP_ACC.2/ADEL and FDP_ACF.1.1/ADEL
- FDP_RIP.1/ADEL
- FMT_MSA.1/ADEL
- FMT_MSA.3/ADEL
- FMT_SMF.1/ADEL
- FMT_SMR.1/ADEL

- FPT_FLS.1/ADEL

9.1.1.7 SF.ODEL: Object Deletion

The ODEL Security Function enforces rules for object deletion and resource management after the object deletion as defined in [\[JCVM\]](#).

Any previous information content of a resource is made unavailable upon the deallocation of the resource from the deleted objects.

A secure state is preserved when the object deletion functions fail to delete all the unreferenced objects owned by the applet that requested the deletion.

The following SFR are fulfilled by SF.ODEL:

- FDP_RIP.1/ODEL
- FPT_FLS.1/ODEL

9.1.1.8 SF.CAR: Secure Carrier

The Secure Carrier Security Function ensures the confidentiality, integrity and authenticity for applet package loading between on-card entity and off-card entity. This is achieved by the supporting of DAP verification and secure channel protocols including SCP02 and SCP03.

The following SFR are fulfilled by SF.CAR:

- FCO_NRO.2/CM
- FDP_IFC.2/CM and FDP_IFF.1/CM
- FDP_UIT.1/CM
- FIA_UID.1/CM
- FMT_MSA.1/CM
- FMT_MSA.3/CM
- FMT_SMF.1/CM
- FMT_SMR.1/CM
- FTP_ITC.1/CM

9.1.1.9 SF.SCP: Smart Card Platform

The SCP Security Function provides capabilities for self-protection and data integrity based on the security services provided by THN31 IC hardware and Native COS.

The THN31 IC hardware is tampering-resistant with a set of security mechanisms to detect and react upon malicious operations. Moreover, the hardware is capable of verifying the integrity of sensitive data like Keys and PIN. Lastly, a secure random number generator is provided as well.

The Native COS provides transaction mechanism to ensure that reading from and writing to static and objects' fields interrupted by power loss have the property that the operation either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

The following SFR are fulfilled by SF.SCP:

- FPT_TST.1/SCP
- FPT_PHP.3/SCP
- FPT_RCV.4/SCP
- FCS_RNG.1

9.1.1.10 SF.CM: Card Manager

The Card Manager Security Function enforce the CARD CONTENT MANAGEMENT rules as defined in [\[GP\]](#). The CARD CONTENT MANAGEMENT rules include the runtime behaviors of loading, installation, extradition, personalization, and deletion according to [\[GP\]](#).

The following SFR are fulfilled by SF.CM:

- FDP_ACC.1/CMGR and FDP_ACF.1/CMGR
- FMT_MSA.1/CMGR
- FMT_MSA.3/CMGR

9.1.1.11 SF.OSU: OS update

The SF.OSU provides secure functionality to update the eUICC OS or eSE OS with an image created by a trusted off-card entity.

The SF.OSU employs two distinct sets of keys to protect the integrity and confidentiality of the UPDATE_IMAGE.

- The ECC key used for Integrity protection,
- The AES encryption used for Confidentiality protection.

The SF.OSU is a standalone subsystem and is selected by a strict process.

The SF.OSU ensures:

- the Update Image must be kept confidential;
- the Update Image to be protected from unauthorized modification;
- the system stays in a secure state when the update fails.

The following SFR are fulfilled by SF.OSU:

- FDP_IFC.2/OSU
- FDP_IFF.1/OSU
- FMT_MSA.3 /OSU
- FMT_MSA.1/OSU
- FMT_SMR.1/OSU
- FMT_SMF.1/OSU
- FIA_UID.1/OSU
- FIA_UAU.1 /OSU
- FIA_UAU.4 /OSU
- FPT_FLS.1/OSU

9.1.2 SFs for eUICC

9.1.2.1 SF.PLATFORM_EUICC: eUICC platform management

The SF.PLATFORM_EUICC provides the Profile Rules Enforcer, Profile Package Interpreter and the Telecom Framework.

The SF. PLATFORM_EUICC enforce the rules as defines in [[SGP21](#)] and [[SGP22](#)].

The following SFR are fulfilled by SF.PLATFORM_EUICC:

- FIA_UID.1/EXT
- FIA_UAU.1/EXT
- FIA_USB.1/EXT
- FIA_UAU.4/EXT
- FIA_UID.1/MNO-SD
- FIA_USB.1/MNO-SD
- FIA_ATD.1/Base
- FIA_API.1
- FDP_ACC.1/ISDR
- FDP_ACF.1/ISDR
- FDP_ACC.1/ECASD
- FDP_ACF.1/ECASD
- FDP_IFC.1/Platform_services
- FDP_IFF.1/Platform_services
- FPT_FLS.1/Platform_services
- FPT_EMS.1/Base
- FDP_SDI.1/Base
- FDP_RIP.1/Base

- FMT_MSA.1/PLATFORM_DATA
- FMT_MSA.1/RULES
- FMT_MSA.1/CERT_KEYS
- FMT_SMF.1/Base
- FMT_SMR.1/Base
- FMT_MSA.1/RAT
- FMT_MSA.3
- FPT_FLS.1/Base

9.1.2.2 SF.COMM_EUICC: eUICC Communication

The TSF build secure communications with external users. The TSF enforce communication

- between U.SM-DP+ and S.ISD-R in a SCP-SGP22 secure channel
- between U.MNO-OTA and U.MNO-SD in a SCP80 or SCP81 secure channel.

These secure channels are used to import commands and objects, thus requiring that these commands and objects are consistently interpreted by the TSF.

These secure channels are established according to a security policy and this policy specifically requires protection of the confidentiality and integrity of transmitted information.

The following SFR are fulfilled by SF.COMM_EUICC:

- FDP_IFC.1/SCP
- FDP_IFF.1/SCP
- FTP_ITC.1/SCP
- FDP_ITC.2/SCP
- FPT_TDC.1/SCP
- FDP_UCT.1/SCP
- FDP_UIT.1/SCP
- FCS_CKM.1/SCP-SM
- FCS_CKM.2/SCP-MNO
- FCS_CKM.6/SCP-SM
- FCS_CKM.6/SCP-MNO

9.1.2.3 SF.NETWORK: Mobile Network authentication

This TSF provides the authentication of the eUICC on MNO networks.

This TSF implements cryptographic mechanisms for the authentication on the MNO network meet the following:

- MILENAGE with the following restrictions:

- Only use 128-bit AES as the kernel function. Do not support other choices
 - Allow any value for the constant OP
 - Allow any value for the constants C1-C5 and R1-R5, subject to the rules.
- Tuak with the following restrictions:
 - Allow any value of TOP
 - Allow multiple iterations of Keccak
 - Support 256-bit K as well as 128-bit
 - To restrict supported sizes for RES, MAC, CK and IK to those currently supported in 3GPP standards.

The following SFR are fulfilled by SF.NETWORK:

- FCS_COP.1/Mobile_network
- FCS_CKM.2/Mobile_network
- FCS_CKM.6/ Mobile_network

10 Composition with IC

10.1 Statement of compatibility - Threats part

IC Threats	Rationale
Part of [ST/IC]	
T.Leak-Inherent	This threat is related to the information which is leaked from the TOE during usage of the Security IC in order to disclose sensitive data of the TOE. <i>It is considered in the TOE evaluation.</i>
T.Phys-Probing	This threat is related to physical probing of the TOE to disclose relevant information. <i>It is considered in the TOE evaluation.</i>
T.Malfunction	This threat is related to force malfunctions of the TSF due to environmental stress that could lower or bypass the implemented security mechanisms. <i>It is considered in the TOE evaluation.</i>
T.Phys-Manipulation	This threat is related to physical manipulation of the Security IC. <i>It is covered by the IC evaluation.</i>

T.Leak-Forced	This threat is related to information which is leaked from the TOE during usage of the Security IC in order to disclose confidential user data of the composite TOE. <i>It is covered by the IC evaluation.</i>
T.Abuse-Func	This threat is related to the usage of functions of the TOE that are not allowed once the TOE Delivery and can impact the security of the TOE. <i>It is considered in the TOE evaluation.</i>
T.RND	This threat is related to the deficiency of random numbers. <i>It is covered by the IC evaluation.</i>

Table 44: Statement of compatibility - Threats part

10.2 Statement of compatibility - OSPs part

IC SFRs	Rationale
Part of [ST/IC]	
P.Process-TOE	This policy is related to the accurate unique identification during IC Development and Production. <i>It is covered by the IC evaluation.</i>
P.Crypto-Service	This policy is related to Cryptographic services (ECC, TDES and AES). <i>It is covered by the ALC_DVS.2 activity of the TOE evaluation.</i>

Table 45: Statement of compatibility - OSPs part

10.3 Statement of compatibility - Assumptions part

IC Assumptions	Rationale
Part of [ST/IC]	
A.Process-Sec-IC	This assumption ensures the security of the delivery and storage of the IC. <i>It is covered by the ALC_DVS.2 activity of the TOE evaluation.</i>
A.Resp-Appl	This assumption ensures that security relevant data of the current TOE are properly treated according to the IC security needs. <i>It is covered by the ADV_IMP.1 activity of the TOE evaluation.</i>

Table 46: Statement of compatibility - Assumptions part

10.4 Statement of compatibility - Security objectives for the environment part

IC OEs are separated in the following groups as defined in appendix 1.1 of [CEM]:

- **IrOE:** IC OE being not relevant for the current TOE.
- **CfPOE:** IC OE being fulfilled by the current TOE automatically.
- **SgOE:** The remaining IC OE which shall be addressed by the current TOE.

IC OEs	Rationale
Part of [ST/IC]	
OE.Resp-Appl	This objective deals with the treatment of TOE user data by the TOE itself. <i>It is covered by the ADV_IMP.1 activity of the TOE evaluation.</i> ● During phase c: CfPOE
OE.Process-Sec-IC	This objective is covered by the IC evaluation and by the ALC_DVS.2 activity of the TOE evaluation. ● During phase c: CfPOE

Table 47: Statement of compatibility - Security objectives for the environment part

10.5 Statement of compatibility - Security objectives part

IC Security objectives	Rationale
Part of [ST/IC]	
O.Identification	<i>This objective is covered by the IC evaluation.</i>
O.Leak-Inherent	<i>This objective is covered by TOE evaluation.</i>
O.Phys-Probing	<i>This objective is covered by TOE evaluation.</i>
O.Malfunction	<i>This objective is covered by TOE evaluation.</i>
O.Phys-Manipulation	<i>This objective is covered by the IC evaluation.</i>
O.Leak-Forced	<i>This objective is covered by the IC evaluation.</i>
O.Abuse-Func	<i>This objective is covered by the TOE evaluation.</i>
O.RND	<i>This objective is covered by the IC evaluation.</i>
O.TDES	<i>This objective is covered by the IC evaluation.</i>
O.AES	<i>This objective is covered by the IC evaluation.</i>
O.RSA	<i>This objective is covered by the IC evaluation.</i>
O.ECC	<i>This objective is covered by the IC evaluation.</i>

Table 48: Statement of compatibility - Security objectives part

10.6 Statement of compatibility - SFRs part

IC SFRs are separated in the following groups as defined in [CEM]:

- **IP_SFR** : Irrelevant IC SFR not being used by the current TOE.
- **RP_SFR SERV** : Relevant IC SFR being used by the current TOE to implement a security service with associated TSFL.
- **RP_SFR MECH** : Relevant IC SFR being used by the current evaluation because of its security properties providing protection attacks to the TOE as a whole and are addressed in ADV_ARC. These required security properties are a result of the security mechanisms and services that are implemented in the IC

IC SFRs	TMC_OS SFRs	Rationale
FRU_FLT.2	-	RP_SFR-MECH
FPT_FLS.1	FPT_FLS.1/Platform_services FPT_FLS.1/Installer FPT_FLS.1/JCS FPT_FLS.1/ADEL FPT_FLS.1/ODEL FPT_FLS.1/OSU FPT_FLS.1/Base	RP_SFR-MECH
FDP_ITT.1 FDP_IFC.1 FPT_ITT.1	FPT_EMS.1/Base FDP_IFC.1/JCVM FDP_IFC.1/SCP FDP_IFC.1/Platform_services	RP_SFR-SERV
FPT_PHP.3	-	RP_SFR-MECH
FDP_SDC.1	-	RP_SFR-MECH
FDP_SDI.2	FDP_SDI.2/DATA	RP_SFR-MECH
FMT_LIM.1	-	RP_SFR-MECH
FMT_LIM.2	-	RP_SFR-MECH
FAU_SAS.1	-	RP_SFR-SERV
FCS_RNG.1[PTG.2]	FCS_RNG.1	RP_SFR-SERV
FCS_COP.1/TDES	FCS_COP.1/TDES FCS_COP.1/TDES_MAC	RP_SFR-SERV
FCS_COP.1/AES	FCS_COP.1/AES_CIPHER FCS_COP.1/AES_MAC	RP_SFR-SERV
FCS_COP.1/RSA-CRT	-	IP_SFR
FCS_COP.1/ECC	FCS_COP.1/ECDSA FCS_COP.1/ECDH	RP_SFR-SERV

Table 49: Statement of compatibility - SFRs part

11 References

11.1 Glossary of vocabulary

Term	Definition
AID	Application identifier, an ISO-7816 data format used for unique identification of Java Card applets (and certain kinds of files in card file systems). The Java Card platform uses the AID data format to identify applets, packages and CAP files. AIDs are administered by the International Opens Organization (ISO), so they can be used as unique identifiers. AIDs are also used in the security policies (see “Context” below): applets’ AIDs are related to the selection mechanisms; Package AIDs are used in the enforcement of the firewall. Note: although they serve different purposes, they share the same namespace.
APDU	Application Protocol Data Unit, an ISO 7816-4 defined communication format between the card and the off-card applications. Cards receive requests for service from the CAD in the form of APDUs. These are encapsulated in Java Card System by the javacard.framework.APDU class (JCAPII). APDUs manage both the selection-cycle of the applets (through Java Card RE mediation) and the communication with the Currently selected applet.
APDU buffer	The APDU buffer is the buffer where the messages sent (received) by the card depart from (arrive to). The Java Card RE owns an APDU object (which is a Java Card RE Entry Point and an instance of the javacard.framework.APDU class) that encapsulates APDU messages in an internal byte array, called the APDU buffer. This object is made accessible to the currently selected applet when needed, but any permanent access (out of selection-scope) is strictly prohibited for security reasons.
Applet	The name given to any Java Card technology-based application. An applet is the basic piece of code that can be selected for execution from outside the card. Each applet on the card is uniquely identified by its AID.
Applet deletion manager	The on-card component that embodies the mechanisms necessary to delete an applet or library and its associated data on smart cards using Java Card technology.
BCV	The bytecode verifier is the software component performing a static analysis of the code to be loaded on the card. It checks several kinds of properties, like the correct format of CAP files and the enforcement of the typing rules associated to bytecodes. If the component is placed outside the card, in a secure environment, then it is called an off-card verifier. If the component is part of the embedded software of the card it is called an on-card verifier.
CAD	Card Acceptance Device or card reader. The device where the card is inserted,

	and which is used to communicate with the card. Unless explicitly said otherwise, in this document, CAD covers PCD.
CAP file	A file in the Converted applet format. A CAP file contains a binary representation of one or several packages of classes that can be installed on a device and used to execute the packages' classes on a Java Card virtual machine. A CAP file can contain a user library, or the code of one or more applets.
Class	In object-oriented programming languages, a class is a prototype for an object. A class may also be considered as a set of objects that share a common structure and behavior. Each class declares a collection of fields and methods associated to its instances. The contents of the fields determine the internal state of a class instance, and the methods the operations that can be applied to it. Classes are ordered within a class hierarchy. A class declared as a specialization (a subclass) of another class (its super class) inherits all the fields and methods of the latter. Java platform classes should not be confused with the classes of the functional requirements (FIA) defined in the CC.
Context	A context is an object-space partition associated to a CAP file. Applets in Java technology-based packages contained within the same CAP file belong to the same context. The firewall is the boundary between contexts (see “Current context”).
Current context	The Java Card RE keeps track of the current Java Card System context (also called “the active context”). When a virtual method is invoked on an object, and a context switch is required and permitted, the current context is changed to correspond to the context of the applet that owns the object. When that method returns, the previous context is restored. Invocations of static methods have no effect on the current context. The current context and sharing status of an object together determine if access to an object is permissible.
Currently selected applet	The applet has been selected for execution in the current session. The Java Card RE keeps track of the currently selected Java Card applet. Upon receiving a SELECT command from the CAD or PCD with this applet's AID, the Java Card RE makes this applet the currently selected applet over the I/O interface that received the command. The Java Card RE sends all further APDU commands received over each interface to the currently selected applet on this interface ([JCRE], Glossary).
Default applet	The applet that is selected after a card reset or upon completion of the PICC activation sequence on the contactless interface ([JCRE], §4.1).
DPA	Differential Power Analysis is a form of side channel attack in which an attacker studies the power consumption of a cryptographic hardware device such as a smart card.
Embedded Software	Pre-issuance loaded software.
Firewall	The mechanism in the Java Card technology for ensuring applet isolation and object sharing. The firewall prevents an applet in one context from unauthorized access to objects owned by the Java Card RE or by an applet in another context.

Installer	The installer is the on-card application responsible for the installation of applets on the card. It may perform (or delegate) mandatory security checks according to the card issuer policy (for bytecode-verification, for instance), loads and link CAP files on the card to a suitable form for the Java Card VM to execute the code they contain. It is a subsystem of what is usually called “card manager”; as such, it can be seen as the portion of the card manager that belongs to the TOE. The installer has an AID that uniquely identifies him, and may be implemented as a Java Card applet. However, it is granted specific privileges on an implementation-specific manner ([JCRE], §10).
Interface	A special kind of Java programming language class, which declares methods, but provides no implementation for them. A class may be declared as being the implementation of an interface, and in this case must contain an implementation for each of the methods declared by the interface (See also shareable interface).
Java Card RE	The runtime environment under which Java programs in a smart card are executed. It is in charge of all the management features such as applet lifetime, applet isolation, object sharing, applet loading, applet initializing, transient objects, the transaction mechanism and so on.
Java Card RE Entry Point	An object owned by the Java Card RE context but accessible by any application. These methods are the gateways through which applets request privileged Java Card RE services: the instance methods associated to those objects may be invoked from any context, and when that occurs, a context switch to the Java Card RE context is performed. There are two categories of Java Card RE Entry Point Objects: Temporary ones and Permanent ones. As part of the firewall functionality, the Java Card RE detects and restricts attempts to store references to these objects.
Java Card RMI	Java Card Remote Method Invocation is the Java Card System version 2.2 and 3 Classic Edition mechanism enabling a client application running on the CAD platform to invoke a method on a remote object on the card. Notice that in Java Card System, version 2.1.1, the only method that may be invoked from the CAD is the process method of the applet class and that in Java Card System, version 3 Classic Edition, this functionality is optional.
Java Card System	Java Card System includes the Java Card RE, the Java Card VM, the Java Card API and the installer.
Java Card VM	The embedded interpreter of bytecodes. The Java Card VM is the component that enforces separation between applications (firewall) and enables secure data sharing.
Logical channel	A logical link to an application on the card. A new feature of the Java Card System, version 2.2 and 3 Classic Edition, that enables the opening of simultaneous sessions with the card, one per logical channel. Commands issued to a specific logical channel are forwarded to the active applet on that logical channel. Java Card platform, version 2.2.2 and 3 Classic Edition, enables opening up to twenty logical channels over each I/O interface (contacted or

	contactless).
NVRAM	Non-Volatile Random-Access Memory, a type of memory that retains its contents when power is turned off.
Object deletion	The Java Card System version 2.2 and 3 Classic Edition mechanism ensures that any unreferenced persistent (transient) object owned by the current context is deleted. The associated memory space is recovered for reuse prior to the next card reset.
Package	A package is a namespace within the Java programming language that may contain classes and interfaces. A package defines either a user library, or one or more applet definitions. A package is divided in two sets of files: export files (which exclusively contain the public interface information for an entire package of classes, for external linking purposes; export files are not used directly in a Java Card virtual machine) and CAP files where CAP file may contain one or more packages.
PCD	Proximity Coupling Device. The PCD is a contactless card reader device.
PICC	Proximity Card. The PICC is a card with contactless capabilities.
RAM	Random Access Memory, is a type of computer memory that can be accessed randomly.
SCP	Smart Card Platform. It is comprised of the integrated circuit, the operating system and the dedicated software of the smart card.
Shareable interface	An interface declaring a collection of methods that an applet accepts to share with other applets. These interface methods can be invoked from an applet in a context different from the context of the object implementing the methods, thus “traversing” the firewall.
SIO	An object of a class implementing a shareable interface.
Subject	An active entity within the TOE that causes information to flow among objects or change the system’s status. It usually acts on the behalf of a user. Objects can be active and thus are also subjects of the TOE.
SWP	The Single Wire Protocol is a specification for a single-wire connection between the SIM card and a Near Field Communication (NFC) chip in a mobile handset.
Transient object	An object whose contents are not preserved across CAD sessions. The contents of these objects are cleared at the end of the current CAD session or when a card reset is performed. Writes to the fields of a transient object are not affected by transactions.
User	Any application interpretable by the Java Card RE. That also covers the packages. The associated subject(s), if applicable, is (are) an object(s) belonging to the javacard.framework.applet class.

11.2 References

Agent	Name
[CC1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model. CC:2022 Revision 1. November 2022.
[CC2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements. CC:2022 Revision 1. November 2022.
[CC3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance requirements. CC:2022 Revision 1. November 2022.
[CC5]	Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements. CC:2022 Revision 1. November 2022.
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. CC:2022 Revision 1. November 2022. CCMC-2023-04-001.
[CC-COMP]	Common Criteria Supporting Document, Mandatory Technical Document Composite product evaluation and certification, version 1.6, April 2024.
[GP]	GlobalPlatform Card Specification Version 2.3.1
[GP_Amendment D]	Card Technology Secure Channel Protocol '03' Card Specification v2.2 – Amendment D V1.1.1
[PP JCS]	Java Card System Protection Profile, Version 3.1, April 2020, registered and certified by the German certification body (BSI) under the following references: [BSI-CC-PP-0099-V2-2020].
[PP0084b]	BSI-CC-PP-0084-2014, Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, 13 January 2014.
[AGD_PRE]	TMS_TMCOS4.00.3_eSA_AGD_PRE_V1.9.
[AGD_OPE]	TMS_TMCOS4.00.3_eSA_AGD_OPE_V1.11.
[PP eUICC]	GSMA SGP.25 - eUICC for Consumer and IoT Devices Protection Profile, Version 2.1, 3 February 2025.
[PP IC]	Security IC Platform Protection Profile with Augmentation Packages, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, Version 1.0, 13 January 2014.
[SGP 22]	GSMA Remote SIM Provisioning (RSP) Technical Specification – v3.1, SGP.22, December 2023.
[SGP 21]	GSMA Remote SIM Provisioning (RSP) Architecture – v3.0, SGP.21, 28 March 2022.
[Profile PKG]	Trusted Connectivity Alliance (formerly SIMalliance) eUICC Profile Package: Interoperable Format Technical Specification, Version 2.3.1, September 2020
[SGP 02]	GSMA Remote Provisioning Architecture for Embedded UICC Technical Specification - v4.0, SGP.02, February 2019.
[JAVASPEC]	The Java Language Specification. Third Edition, May 2005. Gosling, Joy, Steele and Bracha. ISBN 0-321-24678-0.

[JCRE]	Published by Oracle. Java Card Platform, Runtime Environment Specification, Classic Edition, Version 3.1, February 2021
[JVM]	The Java Virtual Machine Specification. Lindholm, Yellin. ISBN 0-201-43294-3.
[JCVM]	Published by Oracle. Java Card Platform, Virtual Machine Specification, Classic Edition, Version 3.1, February 2021
[JCAPI]	Published by Oracle. Java Card Platform, Application Programming Interface, Classic Edition, Version 3.1, February 2021
[THN31P]	THN31 1.0.2 Secure Element V1.0.
[MILENAGE]	3GPP TS 35.205, 3GPP TS 35.206, 3GPP TS 35.207, 3GPP TS 35.208, 3GPP TR 35.909 (Release 11): "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; 3G Security; Specification of the MILENAGE Algorithm Set: An example algorithm set for the 3GPP authentication and key generation functions f_1, f_1^*, f_2, f_3, f_4, f_5 and f_5^*; • Document 1: General; • Document 2: Algorithm Specification; • Document 3: Implementors Test Data; • Document 4: Design Conformance Test Data; Document 5: Summary and results of design and evaluation
[Tuak]	3GPP TS 35.231, 3GPP TS 35.232, 3GPP TS 35.233, version 12.1.0, release 12, December 2014. • Document 1: Algorithm specification; • Document 2: Implementors' test data; Document 3: Design conformance test data.
[PP USIM]	(U)SIM Java Card Platform Protection Profile Basic and SCWS Configurations, version 2.0.2, July 2010, ANSSI-CC-PP-2010/05
[GP Applications]	GlobalPlatform Card Composition Model Security Guidelines for Basic Applications, version 2.0, December 2014 – ref. GPC_GUI_050.
[ETSI_TS_102221]	ETSI TS 102 221 - Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 9).