NXP JCOP 7.x with eUICC extension on SN300 B1.1 Secure Element

Security Target Lite

Rev. 6.6 — 18 August 2025 **Product evaluation document**
EUCC-2500052-01 **PUBLIC**

Document information

Information	Content
Keywords	NXP, ASE, JCOP 7.x with eUICC extension on SN300 B1.1 Secure Element, Single Chip Secure Element and NFC Controller, JCOP, Common Criteria, EAL4 augmented
Abstract	This document is the Security Target of NXP JCOP 7.x with eUICC extension on SN300 B1.1 Secure Element, developed and provided by NXP Semiconductors. The TOE complies with Evaluation Assurance Level 4 of the Common Criteria for Information Technology Security Evaluation CC:2022 with augmentations.

Revision History

Revision history

Revision number	Date	Description
6.6	2025-08-18	Derived from full Security Target.

1 ST Introduction (ASE_INT)

1.1 ST Reference

"NXP JCOP 7.x with eUICC extension on SN300 B1.1 Secure Element", Security Target Lite, Revision 6.6, 18 August 2025.

1.2 TOE Reference

Table 1. TOE Reference

Content	Version
Product Type	eUICC
TOE name	NXP JCOP 7.x with eUICC extension on SN300 B1.1 Secure Element
TOE version(s)	JCOP 7.0 R1.64.0.2 JCOP 7.0 R2.04.0.2 JCOP 7.1 R1.04.0.2 JCOP 7.2 R1.09.0.2 JCOP 7.3 R1.07.0.2

1.3 TOE Overview

1.3.1 Usage and Major Security Features of the TOE

The JCOP 7.x with eUICC extension on SN300 B1.1 Secure Element combines an eUICC Application, a Java Card Operating System, and an Embedded Secure Element with a NFC Controller on a single die. It also provides a Power Management Unit and IC specific software services.

The Main Operating System creates separate and independently updatable secondary Operating Systems providing a converged product.

All Secondary OSs at the platform level are underpinned by the same Java Card and Global Platform technology, but are dedicated to their own application.

The TOE is the Java Card eUICC OS embedded on the SN300 Secure Element with IC Dedicated Software. It excludes the NFC Controller, the Power Management Unit and Java Card eSE OS.

The component of the SN300 on which the TOE executes is the embedded Secure Element, abbreviated to SN300_SE. Figure 1 provides an overview of the TOE and the place in the system.

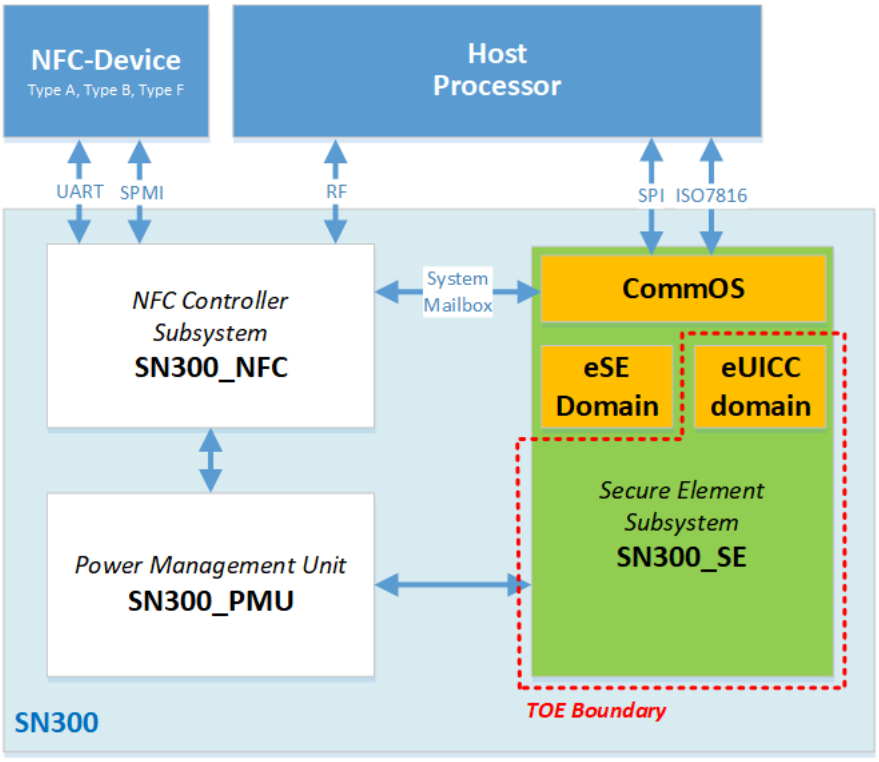

Figure 1. Place in the System

The TOE communicates, via the CommOS, with the Host Processor through an ISO7816 interface and with the integrated NFC controller through the System Mailbox. The integrated NFC controller is not in scope of this evaluation.

The TOE is the embedded UICC (eUICC) software that implements the GSMA Remote SIM Provisioning (RSP) Architecture for Consumer Devices [30], [31].

The TOE provides a variety of security features. The hardware of the Micro Controller already protects against physical attacks by applying various sensors to detect manipulations. Hardware accelerators process data in ways protected against leakage by side channel analysis. With the software stack the TOE provides many cryptographic primitives for encryption, decryption, signature generation, signature verification, key generation, secure management of PINs and secure storage of confidential data (e.g. keys, PINs). Also the software stack implements several countermeasures to protect the TOE against attacks.

Secure Element Hardware:

The TOE incorporates an high frequency clocked ARM Cortex M33 processor augmented with its dedicated coprocessor (SYM-lite), a secure copy machine (SMA), and a Public-Key Cryptography (PKC) coprocessor, which are all connected to a bus system. This bus system gives access to memories, hardware peripherals and communication interfaces. The PKC coprocessor provides large integer arithmetic operations, which can be used by Security IC Embedded Software for asymmetric-key cryptography. Hardware peripherals include coprocessors for symmetric-key cryptography and for calculation of error-detecting codes, and also a random number generator. On-chip memories are Flash memory, ROM and RAMs. The Flash memory can be used to store data and code of Security IC Embedded Software. It is designed for reliable non-volatile storage.

The security functionality of the TOE is designed to act as an integral part of a security system composed of hardware and Security IC Embedded Software to strengthen it as a whole. Several security mechanisms of the TOE are completely implemented in and controlled by the SN300 Secure Element. Other security mechanisms is treated by Security IC Embedded Software. All security functionality is targeted for use in a potential insecure environment, in which the TOE maintains

- correct operation of the security functionality
- integrity and confidentiality of data and code stored to its memories and processed in the device
- controlled access to memories and hardware components supporting separation of different applications.

This is ensured by the construction of TOE and its security functionality.

The following list contains the main features of the TOE:

- hardware to perform computations on multiprecision integers, which are suitable for public-key cryptography
- hardware to calculate the Data Encryption Standard with up to three keys
- hardware to calculate the Advanced Encryption Standard (AES) with different key lengths
- hardware to support Cipher Block Chaining (CBC), Cipher Feedback (CFB) and Counter (CTR) modes of operation for symmetric-key cryptographic block ciphers
- hardware to support Galois/Counter Mode (GCM) of operation for symmetric-key cryptographic block ciphers
- hardware to calculate Cyclic Redundancy Checks (CRC)
- hardware to serve with True Random Numbers
- hardware to control access to memories and hardware components

In addition, the hardware embeds sensors, which ensure proper operating conditions of the device. Integrity protection of data and code involves error correction and error detection codes, light sensing and other security functionality. Memory encryption and masking mechanisms are implemented to preserve confidentiality of data. The IC hardware is shielded against physical attacks.

Cryptographic algorithms and functionality:

- AES
- Triple-DES (3DES)
- RSA for en-/decryption and signature generation and verification
- RSA key generation
- ECDSA signature generation and verification
- ECDH key exchange
- ECC key generation
- ECC point operations and key validation
- Diffie Hellman key exchange on Montgomery Curves over GF(p)
- Key generation for the Diffie Hellman key exchange on Montgomery Curves over GF(p))
- EdDSA signature generation and verification
- EdDSA key generation
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 algorithms

- HMAC algorithms
- eUICC authentication functions (MILENAGE, TUAK and CAVE)
- Multi-precision arithmetic operations including exact division, modular addition, modular subtraction, modular multiplication, modular inversion, arithmetic comparison and exact addition and subtraction.
- Data Protection Module for a secure storage of the the sensitive data.
- Random number generation according to class DRG.3 or DRG.4 of AIS20 [7] and initialized (seeded) by the hardware random number generator of the TOE.

Java Card 3.1 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 compactification.
- Support for Extended Length APDUs.
- Support for Extended CAP file format.
- Persistent Memory Management and Transaction Mechanism.
- Optional JC3.1 Cryptographic APIs [13] are not implemented. A call to those APIs throw an exception of type ISO7816.SW_FUNC_NOT_SUPPORTED in this case.

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 (SCP03).
- Delegated Management, DAP (RSA 1024 and ECC 256).
- Supported Amendments A, B, D, E.
- GSMA Remote SIM Provisionning Architecture for consumer Devices [31]

NXP Proprietary Functionality

- Runtime Configuration Interface: Config Applet that can be used for configuration of the TOE.
- OS Update Component: Proprietary functionality that can update JCOP eUICC, Shared code (including Crypto Lib), FlashOS, SystemOS, CommOS, SMK. This component allows only NXP authorised updates to the product.
- Restricted Mode: In Restricted Mode only very limited functionality of the TOE is available such as reading logging information or resetting the Attack Counter.

- Image4 (IM4) : Software which ensures the customer authorisation of any product updates using OS update features, and provides features to make the update management easier.
- Error Detection Code (EDC) API.

Functionality without specific security claims

- 5G features as per SIM Alliance 2.3
- Programmable Timeout for SMB with Limitations in UGM [45], [51], [57], [63], [69] Section 6
- CPLC data made available through SystemInfo, see UGM [45], [51], [57], [63], [69] Section 1.3.3.
- Compliance to Secure Element configuration, Common Implementation Configuration, UICC Configuration, and UICC Configuration Contactless Extension.

The TOE is offered with the NXP Trust Provisioning Service, which involves secure reception, generation, treatment and insertion of customer data and code at NXP.

1.3.2 TOE Type

The TOE is the eUICC Application on top of the Java Card Operating System and the SN300 Secure Element (including Dedicated Software) on which it is running. It excludes the NFC Controller, the Power Management Unit, as well as other Guest Operating Systems (like JCOP eSE or CommOS) that are considered as domains external to the TOE.

The eUICC is a UICC embedded in a consumer device and may be in a removable form factor or otherwise. It connects to a given mobile network, by means of its currently enabled MNO profile. The eUICC domain is directly accessible by the ISO-7816 interface.

1.3.3 Required non-TOE Hardware/Software/Firmware

As an eUICC product for customer devices, the TOE needs to work together with the other non-TOE parts in the remote provisioning infrastructure for customer devices. Non-TOE parts listed in [12] section 1.2.4 for eUICC for consumer device are applicable, except the Embedded Software, Runtime Environment and IC that are parts of the TOE. Non-TOE parts defined in [11] are applicable, except the Card Manager and the Smart Card Platform that are parts of the TOE.

Three groups of users with their requirements shall be distinguished here.

1. **End-users** group, which uses the TOE eventually with one or more loaded applets in the final form factor as an embedded Secure Element. These users only require a communication device to be able to communicate with the TOE.
   The eUICC TOE communicates via the Secure Mail Box, which is connected to the Integrated NFC controller, and via SPI and ISO7816 direct interfaces. The NFC controller facilitates contactless or wired interfaces supporting:
   - Card Emulation Type A, Type B and Type F according to ETSI 102 622 [33].
   - Wired Mode by using the APDUCard Gate according to ETSI 102 622 [34]. The wired interface is expected to be connected to an applications processor.
2. **Administrators of cards** can configure the TOE by using the Config Applet or install additional applets. These users require the same equipment as end-users.

3. **Applet developers** which develop Java Card applets and executes them on the TOE. These applet developers need in addition to the communication device a set of tools for the development of applets. This set of tools can be obtained from the TOE vendor and comprises elements such as PC development environment, byte code verifier, compiler, linker and debugger.

1.4 TOE Description

The JCOP 7.x with eUICC extension on SN300 B1.1 Secure Element consists of the following components that are part of the TOE:

- SN300 Secure Element excluding NFC (see Section 1.4.1)
- SMK (see Section 1.4.5)
- JCOP eUICC (see Section 1.4.6.1)
- SystemOS (see Section 1.4.4)
- FlashOS (see Section 1.4.3)
- Shared Code
  - Crypto Library (see Section 1.4.2.1)
  - Other Shared Code (see Section 1.4.2)

Furthermore there are components on the platform that are **not part of the TOE**:

- NFC Controller Subsystem
- Power Management Unit
- JCOP eSE
- JCOP xxx (optional - not present in the TOE)
- CommOS

All components and the TOE boundaries are depicted in Figure 2. The components are described in more detail in the following sections.

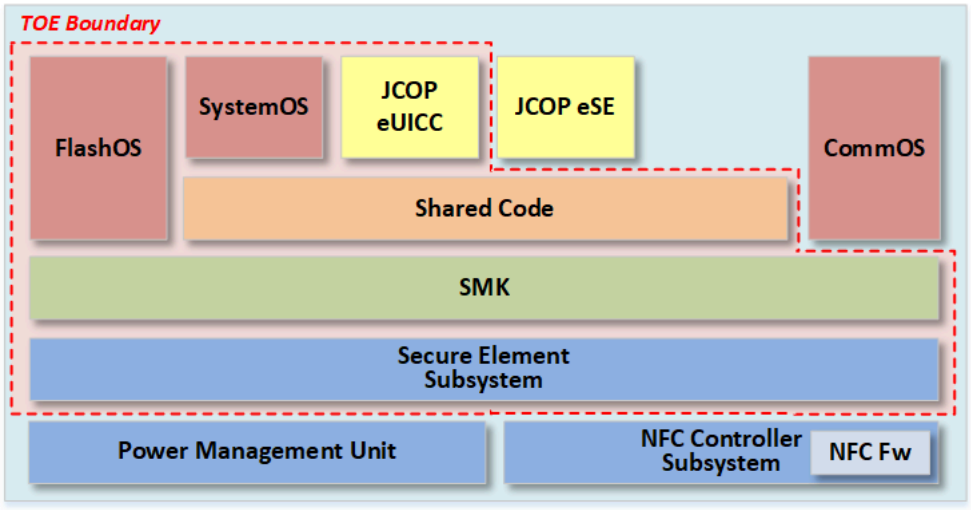

Figure 2. Components of the TOE

The certification of this TOE is a composite certification. This means that for the certification of this TOE other certifications of components which are part of this TOE are re-used. In the following sections more detailed descriptions of the components of

Figure 2 are provided. In the description it is also made clear whether a component is covered by a previous certification or whether it is covered in the certification of this TOE.

1.4.1 Secure Element Subsystem

The SN300 B1.1 is a hardware platform designed to meet the developing needs of the mobile communications market. It embeds a Secure Element Subsystem (SN300_SE), supported by an integrated NFC Controller Subsystem (SN300_NFC) and Power Management Unit (SN300_PMU).

The hardware part of the SN300_SE is referred to as Secure Element Hardware in the following.

The SN300 Secure Element has been certified in a previous certification and the results are reused for this certification. The exact reference to the previous certification is given in the following Table 2:

Table 2. Reference to certified Secure Element Hardware

Hardware Commercial Name	SN300 Series
Certified HW version	SN300_SE B1.1 J9
Certification ID	NSCIB-CC-2300122-02
Security Target Reference	NXP SN300 Series - Secure Element Security Target, Revision 1.0.6, 19 February 2025 [76]

1.4.1.1 Hardware Description

The separation of operating systems is based on the ARM Trustzone-M concept. One Main OS (the SMK) operates in secured privileged state, several Guest OSs operate in separated non-secured states. Each state comes along with an assigned "Context". This Context aware system allows for Virtualization of its components to build an access control mechanism for memories and peripherals that can also be used to share software components between different Operating systems. The separation is enforced throughout the whole system by the Memory Protection Unit, Secure Cache Controller and peripheral bridges.

The SN300 Secure Element implements 512 Kbytes ROM, 2.5 Mbytes Flash, 96 Kbytes System RAM, 5 Kbytes PKC RAM and a Buffer RAM for Flash erase/programming and for Flash read caching. All these memories are accessible over the bus system on data/address busses, and the PKC RAM can also be directly accessed by the PKC coprocessor on a separate data/address bus.

The hardware controls write, read and execute access to the memories over the bus system against system operation modes. Context information is attached to all bus transactions throughout the whole system. Any peripheral on the bus can use the context information to check if access is allowed for the actual context, apply context specific cyphering or to assign associated errors or interrupts to a particular context.

The SN300 Secure Element implements a wide range of hardware components. It embeds the Fast Accelerator for Modular Exponentiation of 3rd Generation (Fame3.5), which can be utilized by the software to accelerate computations required for public-key cryptography like such related to RSA, Elliptic Curve Cryptography (ECC) .

The Secure Generic Interface (SGI) is a symmetric crypto engine that serves the IC Security Embedded Software with interfacing to a DES coprocessor, an AES coprocessor and a GCM coprocessor. The DES coprocessor provides Triple-DES encryption and

decryption in 2-key or 3-key operation with cryptographic key sizes 112 and 168 bits. The AES coprocessor performs AES encryption and decryption calculations with key lengths of 128, 192 or 256 bits. The GCM coprocessor implements a Galois Field Multiplier to support Galois/Counter Mode (AES-GCM) of operation performed by the Crypto Library. Besides ECB mode, the SGI hardware supports chaining mode for e.g. Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB) and Counter Mode (CTR).

The SYM-Lite is a CPU co-processor providing crypto-supporting general purpose operations over sensitive data, outside - but under control of - the CPU.

The Secure Copy Machine (SMA) is a secure DMA. Purpose of the SMA is to copy data between memories and between memories and peripherals in a secure way.

Two CRC coprocessors each serve with checksum computation based on CRC generation polynomials CRC-16 and CRC-32. The Random Number Generator generates true random numbers, which are compliant to AIS31 and FIPS 140-3.

SN300 Secure Element also implements a watchdog counter with time-out mechanism that can be utilized by the software to abort irregular program executions, and provides a CPU Guard with several security functionality, which can be utilized by the software to secure its execution.

The Hardware components can be controlled by the IC Security Embedded Software via Special Function Registers, which are accessible over the bus system on two separate busses. The peripheral control bus is provided for communication and thus gives access to the Special Function Registers of the DMA controller, the communication interfaces, the I/O switch matrix and a component for checksum computations over data streams of the communication interfaces. The Special Function Registers of all other hardware components are accessed on the control bus.

The SN300 Secure Element implements complex security functionality to protect code and data during processing and while stored to the device. This includes appropriate memory encryptions and masking schemes to preserve confidentiality. This also includes error detection codes (the Flash Secure Fetch Plus) to protect against integrity and manifold light sensing to detect perturbations which can lead to integrity violation. Active shielding is present and operating conditions are monitored by sensors on temperature, power supplies and frequencies.

The TOE hardware operates with a power supply provided by the shared Power Management Unit ("SN300_PMU"). The device can be set into sleep and power-down modes, which have different levels of reduced availability of hardware components with appropriately reduced power consumption.

1.4.1.2 IC Dedicated Support Software

The IC Dedicated Support Software of the SN300_SE comprises:

- Test software named *FactoryOS*
- Boot software named *BootOS*
- Memory Driver software named *Flash Driver Software*

BootOS, FactoryOS and Flash Driver Software are stored to ROM. Patches to the BootOS are stored to Flash.

The BootOS is executed during start-up after power-on or reset of the TOE. It sets up the device and its configuration, and finally jumps to a start address in either Mission Mode or Test Mode (if not finally locked).

The FactoryOS is used during manufacturing to load the whole software stack into Flash. The FactoryOS also provides controlled access to different levels of testing capabilities of SN300 Secure Element. Full testing capabilities are under restricted access to NXP for production testing of the TOE and also for in-depth analysis of field returns. In addition, limited testing capabilities are accessible to NXP for basic analysis of field returns, which target to preserve the product in its original condition. Beyond that, the FactoryOS provides some basic functional testing of the SN300 Secure Element and also with a readout of the TOE IC hardware identification flags (if enabled via OEF option). The FactoryOS implements security functionality to protect from unauthorized access and ensures that also authorized access cannot compromise confidentiality of content stored to access controlled Flash areas as well as System Pages. Factory OS implements security functionality against unauthorized access in the field.

Flash Driver Software provides a Hardware Abstraction Layer that is stored to ROM. It supports basic operation of the Flash memory to enable usage of the Flash during Boot Mode and Test Mode.

1.4.2 Shared Code

The Shared Code Subsystem is a software layer containing software components that can be accessed by several OSs. This Shared Code is executed by inheriting the access rights from the caller OS.

The Shared Code of the JCOP OSes comprises the following components:

- Common Native Code (JCOP and Crypto Library Section 1.4.2.1)
- Common JavaCard implementation
- Common GlobalPlatform implementation
- Common JCOPX implementation

1.4.2.1 Crypto Library

The Crypto Library (or parts thereof) comprises a set of cryptographic functions.

AES

- The AES algorithm is intended to provide encryption and decryption functionality.
- The following modes of operation are supported for AES: ECB, CBC, CFB, CTR, GCM, XTS, CBC-MAC, CCM and CMAC.

TDES

- The Triple-DES (TDES) algorithm is intended to provide encryption and decryption functionality.
- The following modes of operation are supported for Triple-DES: ECB, CBC, CFB, CTR, CBC-MAC, RetailMAC and CMAC.

RSA Plain/CRT

- The RSA algorithm can be used for encryption and decryption as well as for signature generation, signature verification, message and signature encoding EME-OAEP, EMSA-PSS, EME-PKCS1- v1_5, EMSA-PKCS1-v1_5 and EMSA-ISO/IEC9796-2.
- The RSA decryption/signature generation can be calculated using keys either in "Straight Forward" format or in CRT format.
- The RSA key generation can be used to generate key pairs either in "Straight Forward" format (i.e. using the "Simple Straight Forward Method") or in CRT format (i.e. using the "Chinese-Remainder-Theorem" method).

- The RSA public key generation can be used to compute the public key that belongs to a given private CRT key.

The TOE supports various key sizes for RSA from 512 to 4096 bits.

ECDSA (ECC over GF(p))

- The ECDSA algorithm can be used for signature generation and signature verification.
- The ECC key generation algorithm can be used to generate key pairs for ECDSA and ECDH.
- The ECDH key exchange algorithm can be used to establish cryptographic keys. It can be also used as secure point multiplication.
- Provide ECC point operations and key validation.

The TOE supports various key sizes for ECC over GF(p) from 128 to 640 bits.

EdDSA & MontDH

- The EdDSA and MontDH over GF(p) library component implements the EdDSA and MontDH over GF(p) related functions:
  - EdDSA key generation, signature generation and signature verification (generalization of Ed25519 and Ed448), support for filling of EdDSA domain parameters
  - MontDH key generation and key exchange for the DH key exchange scheme MontDH (generalization of Curve25519 and Curve448).

The TOE supports various key sizes for EdDsa and MontDH from 128 to 640 bits.

eUICC

- The eUICC library component implements the following MILENAGE and TUAK USIM modes: 3G authentication mode, 3G resynchronization mode, Virtual 2G mode, 3G + Kc mode, Anonymity keys for the 3G modes
- Support of the following CAVE operations: SSD generation, Authentication signature generation, CMEA Key and VPM generation

SHA

- The SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 algorithms can be used for different purposes such as computing hash values in the course of digital signature creation or key derivation.
- The Crypto Library implements two versions of each algorithm with different security level: Standard SHA and Secured SHA. The difference between the standard and high security level of the SHA implementations is that the high security level is protected against differential side channel attacks.

HMAC

- The HMAC algorithm can be used to calculate Keyed-Hash Authentication code. The TOE supports the calculation of HMAC authentication code with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 hash algorithms. The HMAC algorithm can use either the high security level or standard security level version of SHA, depending on required security level.

Random number generation

- Library component to access random numbers generated by a software pseudo random number generator (DRNG). The DRNG is used to fulfill the random numbers Java Card API. It is used as a general purpose random source, i.e. for the generation of cryptographical challenges, generation of session keys, generation of random IVs,

etc. A hardware TRNG is used to seed the DRNG internally to the crypto library with no other usage.

Multi-precision Arithmetic

- The Crypto Library provides functions to implement various arithmetic operations including exact division, secure modular addition, secure modular subtraction, secure modular multiplication, secure modular inversion, secure arithmetic comparison and secure exact addition.

Data Protection Module

- The Crypto Library provides functions to store sensitive data, e.g. symmetric and asymmetry keys, required by the Crypto Library components.

Resistance of cryptographic algorithms against attacks

The cryptographic algorithms are resistant against attacks as described in JIL [8], which include Side Channel Attacks, Perturbation attacks, Differential Fault Analysis (DFA) and timing attacks, except for standard/high security level SHA and HMAC, which are only resistant against Side Channel Attacks and timing attacks.

1.4.3 FlashOS

The FlashOS subsystem consists of the following components

- Service Core

The Services Software comprises the Flash Services Software, the Services Framework Software and the part of the Services HAL (Hardware Abstraction Layer) that is not stored to ROM.

Flash Services Software

- The Flash Services Software manages technical demands of the Flash memory and serves the Security IC Embedded Software with an interface for Flash erase and/or programming.
- The Flash Services Software maintains the Flash with re-freshing, tearing-safe updates of Flash contents and wear leveling techniques to ensure integrity and consistency of its content and optimize its endurance.

Services Framework Software

- The Services Framework Software provides the utility functionality and interface for actual services. This comprises the control of services related functionality such as the resource management, patch handling, service and system configurations functionality.

1.4.4 SystemOS

The SystemOS is a standalone operating system. The SystemOS Component can be used by NXP (or by the customer through Image4 - IM4) to update any software component (JCOP eUICC, , FlashOS, CommOS, Shared code, SMK, SystemOS). It is accessed as decribed in [50], [56], [62] and [68] and its version can be queried independently. SystemOS shares parts of its code with other OSs.

The SystemOS is always booted by SMK, while JCOP eSE is booted (by the SMK) only if it is activated. The SystemOS contains the following functionalities:

- OS updater

- Common Log Interface providing all the logging information supported by the system.
- Get Log Status providing status indicators of the logs
- Reset Attack Counters providing an interface to reset the attack counters of the system
- Get Data Commands of system properties
- Runtime Configuration Interface providing read and write access the configuration items of the system

1.4.4.1 OS Updater Feature

The OS Updater provides the following functionalities:

- it handles APDUs to write a new OS (including Shared Code and SMK) to flash.
- it verifies integrity of the new OS before updating.
- it decrypts the new OS before updating.
- it checks if the new OS can be authenticated and checks if the update can be authorized.
- it ensures that the activation and setting of the information that identifies the new OS is done atomically.
- if the update fails the system stays in a secure state.

1.4.4.2 Image4 (IM4) Feature

The IM4 feature provides control over the OS update processes. This control consists in enforcing that

- OS update steps can be performed only in a particular state of the TOE.
- only allowed OS update plans can be applied to the TOE.

No security claims are made for IM4. The use of IM4 does not compromise any of the security of the OS update mechanisms and all restrictions implied by these mechanisms remain in force. The IM4 implementation does not replace or modify the existing mechanisms by which the SE decrypts, authenticates and authorizes JCOP updates.

1.4.5 SMK

The SMK is a secure microkernel. It is responsible to schedule several Guest Operating Systems. The arrangement of operating systems is determined by a static system configuration passed to the SMK. The SMK is in charge of:

- Providing messaging and scheduling APIs to guest OSs (eSE, eUICC, FlashOS, SystemOS, CommOS,..)
- Providing services APIs to Guest OSs (like Memory Management, Fault/Errors handling...)
- Providing specific APIs available to SystemOS, CommOS, or FlashOS
- Providing APIs for Hardware Peripherals virtualization

Messaging and scheduling allows Guest OSs to exchange messages, recieve signals, handle interrupts, and manage their background activity. The scheduling consists of evaluating the priority of the Guest OSs and the messages, transmitting the message, and switching the context.

The virtualization APIs provide a complete virtualisation of the hardware peripherals like Random Number Generator (TRNG), symmetric/asymmetric crypto accelerators, PUF,...

1.4.6 JCOP eUICC

1.4.6.1 JCOP OS

JCOP OS consists of Native OS, JCVM, JCRE, JCAPI, Extension API, GP framework and Config Applet and IM4. JCVM, JCRE, JCAPI and GP framework are implemented according to the Java Card Specification listed in Table 3 and the applicable Global Platform Specifcation and Amendments are listed in Table 4.

Table 3. Java Card Specification Version

Name	Version
JCVM and JCRE version	Version 3.1 Classic Edition [14] [15]
JC API version	Version 3.1 Classic Edition [13]

Table 4. Global Platform and Amendments

Name	Version
GP Framework	Version 2.3.1 [18]
Amendment A, Confidential Card Content Management	Version 1.1.1 [19]
Amendment B, Remote Application Management over HTTP	Version 1.1.3 [20]
Amendment D, Secure Channel Protocol 03	Version 1.1.2 [21]
Amendment E, Security Upgrade for Card Content Management	Version 1.1 [22]
Common Implementation Configuration	Version 2.1 [24]
GP Card API	Version 1.7 [28]
UICC Configuration	Version 2.0 [26]
Contactless Extension	Version 2.0 [27]

JCOP 7.x eUICC OS identification is obtained using the Version Query command that provides the Platform ID and the Platform Release (a.k.a. Platform String; see UGM [45], [51], [57], [69]). The Platform Identification data, which includes the Hardware Type, JCOP Version, Build Number, Mask ID, a Patch ID and Non-Volatile Memory Size, identifies the JCOP 7.x platform (combination of HW and SW). The Platform Release is a data string that allows to identify the eUICC OS component. Table 13 in Section 1.6 lists all possible values for the Platform ID that are valid for this TOE.

1.4.6.1.1 Native Applications

Historically MIFARE & FELICA have been implemented as native applications on smartcard controllers, but it is no longer the case. JCOP 7.x provides Java Card APIs providing specific support for MIFARE & FELICA standards with access to MIFARE dedicated accelerators and Felica specific Cryptography. The complete implementation of the MIFARE and Felica standards in JCOP 7.x rely upon applets using these accelerated APIs. JCOP 7.x receives, processes and routes commands from the NFC controller according to the pipe used, with MIFARE being received as Type-A APDUs,

either Level 4 ISO wrapped or MIFARE raw commands, and FELICA coming through as raw Type-F commands requiring JCOP to decode, process and route correctly.

1.4.6.2 eUICC component details

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

The eUICC component provides interpretation of the Telecom commands defined in ETSI TS 102 222, ETSI TS 102 221, ETSI TS 131 102, ETSI TS 131 103, 3GPP2 C.S00065-0, and all the support of CAT API (defined in ETSI TS 143 019, ETSI TS 102 241, ETSI TS 131 130). This component uses the security and cryptographic services provided by the JCOP platform.

The eUICC application will contain several MNO Profiles, each of them being associated with a given International Mobile Subscriber Identity (IMSI). The primary function of the MNO Profile is to authenticate the validity of a Device when accessing the network. The Profile is MNO property, and stores MNO specific information. An eUICC with an enabled operational Profile provides the same functionality as a SIM or USIM card.

1.4.6.2.1 eUICC Application Layer

The goal of the Application layer is to implement the eUICC functionalities described in [31] and [29], 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, but one and only one is activated at a time. Each Profile is controlled by a unique ISD-P; consequently, there is one and only one enabled ISD-P at a time on the eUICC.

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 [31] and [29].

In addition to Profile data, the eUICC itself has a Rules Authorisation 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 authorised and can be installed on the eUICC. The RAT is initialised 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.

1.4.6.2.2 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 [29], the ISD-P shall ensure that:

- It hosts a unique Profile;
- 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.
- 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;
- Deletion of a Profile shall remove the containing ISD-P and all Profile components of the Profile.

1.4.6.2.3 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 [29]:

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

1.4.6.2.4 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 LPAd or the LPAe PP-Module-TOE-element LPAe). These services are:

- Transfer Bound Profile Package from the LPAd to the ISD-P;
- Provide list of installed Profiles;
- Retrieve EID;
- Provide Local Profile Management Operations.

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

1.4.6.2.5 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.

1.4.6.2.6 ECA-SD

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.

The ECASD is the representative of the off-card entity CI root. As defined in [31], the ECASD has the following properties:

- There can only be one ECASD on an eUICC;
- It is installed and personalised by the EUM during the eUICC manufacturing as described in [18];
- It has eUICC private key(s) for creating signatures;
- It has associated certificate(s) for eUICC authentication;
- It has the Certificate Issuers (CI) root public key(s) for verifying SM-DP+ and SM-DS certificates;
- It has the certificate of the EUM.

1.4.6.2.7 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 SIMalliance eUICC Profile Package Specification [25] into an installed Profile using the specific internal format of the target eUICC.
- The Profile Policy Enabler, which has two functions:
  - Verification that a Profile containing PPRs is authorised by the RAT;
  - Enforcement of the PPRs of a Profile.

1.4.6.2.8 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 property, and stores MNO specific information.

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

1.4.7 Interfaces of the TOE

Electrical interface

The electrical interface of the TOE are the lines between the I/O interface of the SN300_SE and the communication pads, that are exclusively used by the SN300_SE subsystem. The interface can be configured to establish communication with the TOE via the following interfaces:

- Serial Peripheral Interface (SPI)
- 2x I^2C interfaces
- I^3C interface (shared pins with second I^2C interface)
- ISO/IEC 7816 compliant interface by use of ISO/IEC 7816 UART
- SPMI Interface
- GPIO interface by use of Special Function Registers

The TOE also provides an electrical interface to the SN300_PMU subsystem, which connects power supply voltage input and ground as reference voltage, and an interface to the Power-Clock-Reset Module of the SN300_NFC subsystem. Communication between SN300_SE and SN300_NFC supported by System Mailbox interface.

Logical interface

The logical interface of the TOE accessible to the Security IC Embedded Software is implemented via the CommOS. It provides the following communication channels:

- Secure System Mailbox interface for data exhange with SN300_NFC subsystem
- interface to each Guest JCOP for data exchange
- external interface to access Host Processor

Physical interface

The chip surface must be considered as an interface of the TOE as well. This interface could be exposed to environmental stress or physically manipulated by an attacker.

1.5 TOE Life Cycle

The life cycle for this Java Card is based on the general smart card life cycle defined in the Java Card Protection Profile - Open Configuration [11], but also considers the life-cycle presented by the GSMA Embedded UICC for Consumer Devices Protection Profile (see [12]), both of which are mapped to the lifecycle presented in BSI-PP-0084 [10] (see Figure 3). Authentic delivery of the TOE is supported by the NXP Trust Provisioning Service.

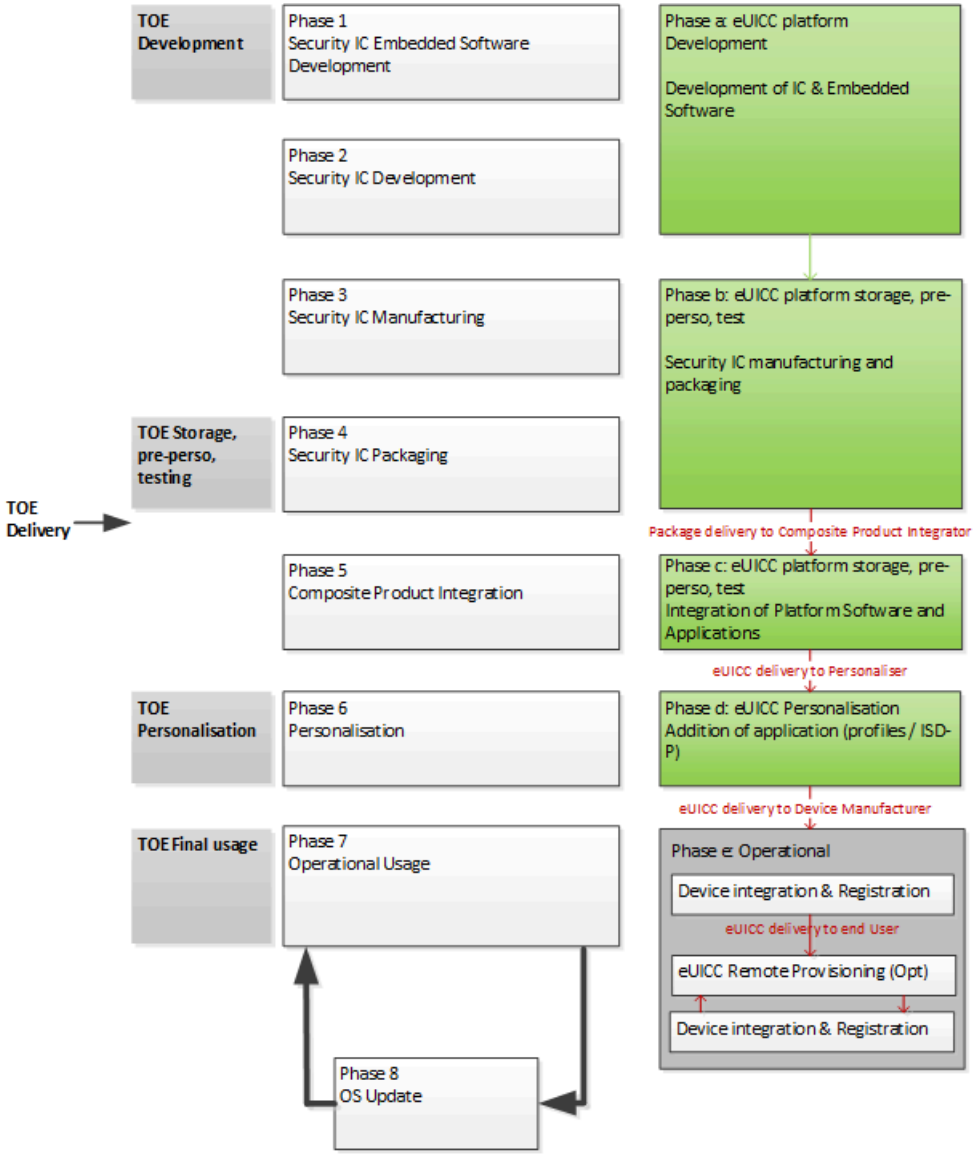

Figure 3. TOE Life Cycle within Product Life Cycle

Table 5.

Phase	Name	Description
1	Security IC Embedded Software Development	The IC Embedded Software Developer is in charge of • smartcard embedded software development including the development of Java Card applets and • specification of IC pre-personalization requirements, though the actual data for IC pre-personalization comes from phase 4, 5, or 6.

Table 5. *...continued*

Phase	Name	Description
2	Security IC Development	The IC Developer • designs the IC, • develops IC Dedicated Software, • provides information, software or tools to the IC Embedded Software Developer, and • receives the embedded software from the developer, through trusted delivery and verification procedures. From the IC design, IC Dedicated Software and Smartcard Embedded Software, the IC Developer • constructs the smartcard IC database, necessary for the IC photomask fabrication.
3	Security IC Manufacturing	The IC Manufacturer is responsible for • producing the IC through three main steps: IC manufacturing, IC testing, and IC pre-personalization. The IC Mask Manufacturer • generates the masks for the IC manufacturing based upon an output from the smarcard IC database. Configuration items may be changed/deleted. The NXP Trust Provisioning Service ensures confidentiality and integrity of any customer data in this phase. This includes secure treatment and insertion of data and code received from the customer as well as random or derived data, which are generated by NXP. JCOP is loaded onto the chip during phase 3.
4	Security IC Packaging	The IC Packaging Manufacturer is responsible for • IC packaging and testing. The delivery processes between all involved sites provide accountability and traceability of the dies. Authentic delivery of the TOE is supported by its NXP Trust Provisioning Service.
5	Composite Product Integration	The Composite Product Manufacturer is responsible for the smartcard product finishing process.
6	Personalization	The Personalizer is responsible for • smartcard (including applet) personalization and final tests. User Applets may be loaded onto the chip at the personalization process and configuration items may be changed/deleted. The Config Applet can be used to set Configuration Items.
7	Operational Usage	The Consumer (e.g. Original Equipment Manufacturer) of Composite Product is responsible for • smartcard product delivery to the smartcard end-user, and the end of life process. • applets may be loaded onto the chip. • triggering an OS update. • Config Applet: changing Config Items. • perform card content management according to Global Platform and Amendments specifications.
8	OS Update	The IC Developper is responsible for providing an updated IC Dedicated Software. The OS update in the field is performed by the consumer as per step 7.

The TOE is delivered to the customer at the end of Phase 4, meaning the evaluation process is limited to phases 1 to 4. User Applet development is outside the scope of this evaluation. Applets can be loaded into Flash memory. Applet loading into Flash memory can be done in phases 3, 4, 5, and 6. Applet loading in phase 7 is also allowed. This means post-issuance loading of applets is allowed. The certification is only valid for platforms that return the Platform Identifier as stated in Table 7 Table 8 Table 9 Table 10 Table 11 . The delivery process from NXP to their customers (to phase 4 or phase 5 of the life cycle) guarantees, that the customer is aware of the exact versions of the different parts of the TOE as outlined above. TOE documentation is delivered in electronic form (encrypted according to defined mailing procedures).

Note: Phases 1 to 3 are under the TOE developer scope of control. Therefore, the objectives for the environment related to phase 1 to 3 are covered by Assurance measures, which are materialized by documents, process and procedures evaluated through the TOE evaluation process. During phases 4 to 7 the TOE is no more under the developer control. In this environment, the TOE protects itself with its own Security functions. But some additional usage recommendation must also be followed in order to ensure that the TOE is correctly and securely handled, and protected against damage or comprimission. This ST assumes (A.USE_DIAG, A.USE_KEYS) that users handle securely the TOE and related Objectives for the environment are defined (OE.USE_DIAG, OE.USE_KEYS).

1.5.1 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 [12]:

Table 6. eUICC lifecycle stages and Delivery Options

Phase	Description
a	Development corresponds to the first two stages of the IC development;
b	Storage, pre-personalisation and test cover the stages related to manufacturing and packaging of the IC; • TOE Delivery [optional]: At this phase the delivery of the TOE to the customer of the eUICC manufacturer could happen, if the TOE is already self-protected;
c	eUICC platform storage, pre-personalization, test covers the stage of the embedding of software products onto the eUICC; • TOE Delivery [optional]: At this phase the delivery of the TOE to the customer of the eUICC manufacturer could happen, if the TOE is already self-protected;
d	eUICC personalization covers the insertion of provisioning Profiles and Operational Profiles onto the eUICC; • TOE Delivery [optional]: 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.

As NXP is both, the IC manufacturer and the eUICC platform developer, Phase 5 Composite Product Integration [10] is not required and done in earlier phases: The TOE

is delivered in Phase 4 while the activities related to eUICC integration ([10] phase 5) already take place in phase 3 in the TOE lifecycle.

Concerning the eUICC product phases as described in [12], the TOE will be delivered at the end of [12] Phase C.

1.6 TOE Identification

The delivery comprises the following items:

Table 7. Delivery items for JCOP 7.0 R1.64.0.2

Type	Name	Identification	Delivery form
IC Hardware	NXP SN300 Series - Secure Element	SN300_SE B1.1 J9 (see UGM, Table 20, and SN300_SE Security Target [76])	Package WLCSP
Embedded Software	JCOP 7.0 R1.64.0.2 OS with eUICC functionalities and including Shared Code (with Cryptolib), FlashOS, CommOS, SystemOS, and SMK.	1.64.0.2 (see UGM below and Table 13)	On-chip software stored into the FLASH area of the TOE.
Document	JCOP 7.0 R1.64.0.2 User Guidance Manual (UGM)	[45]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R1.64.0.2 UGM Addendum	[46]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R1.64.0.2 UGM Anomaly	[47]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R1.64.0.2 (JCOP 7.0 17.4-2.64) UGM for JCOP eUICC	[48]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R1.64.0.2 UGM Addendum UICC	[49]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R1.64.0.2 UGM Addendum System Management	[50]	Electronic Document (PDF via NXP Docstore)

Table 8. Delivery items for JCOP 7.0 R2.04.0.2

Type	Name	Identification	Delivery form
IC Hardware	NXP SN300 Series - Secure Element	SN300_SE B1.1 J9 (see UGM, Table 20, and SN300_SE Security Target [76])	Package WLCSP
Embedded Software	JCOP 7.0 R2.04.0.2 OS with eUICC functionalities and including Shared Code (with Cryptolib), FlashOS, CommOS, SystemOS, and SMK.	2.04.0.2 (see UGM below and Table 13)	On-chip software stored into the FLASH area of the TOE.

Table 8. Delivery items for JCOP 7.0 R2.04.0.2*...continued*

Type	Name	Identification	Delivery form
Document	JCOP 7.0 R2.04.0.2 User Guidance Manual (UGM)	[51]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R2.04.0.2 UGM Addendum	[52]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R2.04.0.2 UGM Anomaly	[53]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R2.04.0.2 (JCOP 7.0 18.4-2.04) UGM for JCOP eUICC	[54]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R2.04.0.2 UGM Addendum UICC	[55]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.0 R2.04.0.2 UGM Addendum System Management	[56]	Electronic Document (PDF via NXP Docstore)

Table 9. Delivery items for JCOP 7.1 R1.04.0.2

Type	Name	Identification	Delivery form
IC Hardware	NXP SN300 Series - Secure Element	SN300_SE B1.1 J9 (see UGM, Table 20, and SN300_SE Security Target [76])	Package WLCSP
Embedded Software	JCOP 7.1 R1.04.0.2 OS with eUICC functionalities and including Shared Code (with Cryptolib), FlashOS, CommOS, SystemOS, and SMK.	1.04.0.2 (see UGM below and Table 13)	On-chip software stored into the FLASH area of the TOE.
Document	JCOP 7.1 R1.04.0.2 User Guidance Manual (UGM)	[57]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.1 R1.04.0.2 UGM Addendum	[58]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.1 R1.04.0.2 UGM Anomaly	[59]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.1 R1.04.0.2 (JCOP 7.1 19.4-2.04) UGM for JCOP eUICC	[60]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.1 R1.04.0.2 UGM Addendum UICC	[61]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.1 R1.04.0.2 UGM Addendum System Management	[62]	Electronic Document (PDF via NXP Docstore)

Table 10. Delivery items for JCOP 7.2 R1.09.0.2

Type	Name	Identification	Delivery form
IC Hardware	NXP SN300 Series - Secure Element	SN300_SE B1.1 J9 (see UGM, Table 20, and SN300_SE Security Target [76])	Package WLCSP
Embedded Software	JCOP 7.2 R1.09.0.2 OS with eUICC functionalities and including Shared Code (with Cryptolib), FlashOS, CommOS, SystemOS, and SMK.	1.09.0.2(see UGM below and Table 13)	On-chip software stored into the FLASH area of the TOE.
Document	JCOP 7.2 R1.09.0.2 User Guidance Manual (UGM)	[63]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.2 R1.09.0.2 UGM Addendum	[64]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.2 R1.09.0.2 UGM Anomaly	[65]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.2 R1.09.0.2 (JCOP 7.2 20.4-2.06) UGM for JCOP eUICC	[66]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.2 R1.09.0.2 UGM Addendum UICC	[67]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.2 R1.09.0.2 UGM Addendum System Management	[68]	Electronic Document (PDF via NXP Docstore)

Table 11. Delivery items for JCOP 7.3 R1.07.0.2

Type	Name	Identification	Delivery form
IC Hardware	NXP SN300 Series - Secure Element	SN300_SE B1.1 J9 (see UGM, Table 20, and SN300_SE Security Target [76])	Package WLCSP
Embedded Software	JCOP 7.3 R1.07.0.2 OS with eUICC functionalities and including Shared Code (with Cryptolib), FlashOS, CommOS, SystemOS, and SMK.	1.07.0.2(see UGM below and Table 13)	On-chip software stored into the FLASH area of the TOE.
Document	JCOP 7.3 R1.07.0.2 User Guidance Manual (UGM)	[69]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.3 R1.07.0.2 UGM Addendum	[70]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.3 R1.07.0.2 UGM Anomaly	[71]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.3 R1.07.0.2 (JCOP 7.3 21.4-2.07) UGM for JCOP eUICC	[72]	Electronic Document (PDF via NXP Docstore)

Table 11. Delivery items for JCOP 7.3 R1.07.0.2*...continued*

Type	Name	Identification	Delivery form
Document	JCOP 7.3 R1.07.0.2 UGM Addendum UICC	[73]	Electronic Document (PDF via NXP Docstore)
Document	JCOP 7.3 R1.07.0.2 UGM Addendum System Management	[74]	Electronic Document (PDF via NXP Docstore)

Table 12. Documentation Errata for JCOP7.0, JCOP7.1, JCOP7.2, JCOP 7.3 UGMs

Type	Name	Identification	Delivery form
Document	ES_JCOP7.x Documentation Errata	[75]	Electronic Document (PDF via NXP Docstore)

1.6.1 Platform Identifier

The Platform can be identified by the Platform ID and the Platform String. See Table 13.

When JCOP eUICC OS is addressed, the Platform ID and the Platform String can be obtained by using the Version Query command (GET DATA command with tag 0xDF4C). See UGM [45], [51], [57], [63], [69] chapter 1.3.5 and [48], [54], [60], [66], [72]).

The TOE is JCOP 7.x eUICC on SN300 B1.1 SE and has three configuration:

- JCOP 7.0 R1.64.0.2
- JCOP 7.0 R2.04.0.2
- JCOP 7.1 R1.04.0.2
- JCOP 7.2 R1.09.0.2
- JCOP 7.3 R1.07.0.2

Table 13. Product Identification

Product Name	Version	Platform ID (Tag 0x82)	Platform String (Tag 0x83, 0x84)
JCOP 7.x with eUICC on SN300 B1.1	JCOP 7.0 R1.64.0.2[1]	N5A2M500013D0000	7016402
JCOP 7.x with eUICC on SN300 B1.1	JCOP 7.0 R2.04.0.2[2]	N5A2M50001890000	7020402
JCOP7.x with eUICC on SN300 B1.1	JCOP 7.1 R1.04.0.2[2]	N5A2M500020A0000	7110402
JCOP7.x with eUICC on SN300 B1.1	JCOP 7.2 R1.09.0.2[3]	N5A2M500028D0000	7210902
JCOP7.x with eUICC on SN300 B1.1	JCOP 7.3 R1.07.0.2[3]	N5A2M500030B0000	7310702

[1] includes CryptoLib v1.0.0 and FlashOS v1.52.0
[2] includes CryptoLib v1.0.0 and FlashOS v1.52.2
[3] includes CryptoLib v1.3.0 and FlashOS v1.53.5

The Platform ID has the following form: **Nabcccxxxxxxyyzz**

The **"N"** is constant, the other letters are variables. For a detailed description of these variables, please see Table 14.

Table 14. Platform ID Format

Variable	Meaning	Value	Parameter Settings
a	Hardware Type	5	NFC hardware
b	JCOP OS Version	A	JCOP 7.0 or JCOP 7.1 or JCOP7.2 or JCOP 7.3
ccc	Non-Volatile Memory Size	2M5	2.5MB
xxxxxx	Build Number (hexadecimal)	00013D	svn rev. JCOP 7.0 R1.64.0.2 OS
		000189	svn rev. JCOP 7.0 R2.04.0.2 OS
		00028D	svn rev. JCOP 7.2 R1.09.0.2 OS
		00030B	svn rev. JCOP 7.3 R1.07.0.2 OS
		00020A	svn rev. JCOP 7.1 R1.04.0.2 OS
yy	Mask ID	00	Mask 00
zz	Patch ID	00	Patch 00

The Platform String has the following form **wxyzzvc** and is to be interpreted as "JCOP w.x Ry.zz.v.c"

Table 15. Platform String Format for JCOP 7.0 R1.64.0.2

Variable	Meaning	Value	Parameter Settings
w	JCOP Major version	7	JCOP 7.0
x	JCOP Minor version	0	
y	JCOP OS Major release	1	R1.64.0
zz	JCOP OS Minor release	64	
v	Variant identifier	0	
c	JCOP instance	2	eUICC

Table 16. Platform String Format for JCOP 7.0 R2.04.0.2

Variable	Meaning	Value	Parameter Settings
w	JCOP Major version	7	JCOP 7.0
x	JCOP Minor version	0	
y	JCOP OS Major release	2	R2.04.0
zz	JCOP OS Minor release	04	
v	Variant identifier	0	

Table 16. Platform String Format for JCOP 7.0 R2.04.0.2*...continued*

Variable	Meaning	Value	Parameter Settings
c	JCOP instance	2	eUICC

Table 17. Platform String Format for JCOP 7.1 R1.04.0.2

Variable	Meaning	Value	Parameter Settings
w	JCOP Major version	7	JCOP 7.1
x	JCOP Minor version	1	
y	JCOP OS Major release	1	R1.04.0
zz	JCOP OS Minor release	04	
v	Variant identifier	0	
c	JCOP instance	2	eUICC

Table 18. Platform String Format for JCOP 7.2 R1.09.0.2

Variable	Meaning	Value	Parameter Settings
w	JCOP Major version	7	JCOP 7.2
x	JCOP Minor version	2	
y	JCOP OS Major release	1	R1.09.0
zz	JCOP OS Minor release	09	
v	Variant identifier	0	
c	JCOP instance	2	eUICC

Table 19. Platform String Format for JCOP 7.3 R1.07.0.2

Variable	Meaning	Value	Parameter Settings
w	JCOP Major version	7	JCOP 7.3
x	JCOP Minor version	3	
y	JCOP OS Major release	1	R1.07.0
zz	JCOP OS Minor release	07	
v	Variant identifier	0	
c	JCOP instance	2	eUICC

1.6.1.1 Sequence Number

Additionally to the Platform Identifier the TOE can also be identified by its sequence number:

1. If SystemOS is active then the "SELECT OS Update AID" command will return the Current Sequence Number of SystemOS and the Reference Sequence Number.
2. If JCOP OS is active then the "Get OS Info" command will return the Current Sequence Number of JCOP eUICC (Final Sequence Number).

1.6.1.2 IC Identifier

When the Systme OS is addressed, the Version Query command can be used to retrieve the identifier of the different components of the SE software and hardware. The Version Query Command is a proprietary GET DATA command with tag 0xDF4C. The Data returned by the Version Query includes the Tag for Hardware ID (tag 0x8C), which is 2 bytes long.

Table 20. Hardware ID Data Format

Tag	Length	Value (MSB only)	Comment
0x8C	2	0x43	SN300 B1.1

The MSB of the Hardware ID provides physical identification of the IC (including ROM contents). Note that the Hardware ID together with the Platform ID uniquely identity the SN300 B1.1 J9 (including SN300 dedicated Flash content).

1.7 Evaluated Package Types

The TOE is delivered as a packaged device. The security of the TOE does not rely on the way the pads are connected to the package. Therefore the security functionality of JCOP 7.x with eUICC on SN300 B1.1 is not affected by the delivered package type.

The only available package type is "Wafer Level Chip Scale Package" (WLCSP). This package is a thin fine-pitch ball grid array package. All (enabled) pins of the TOE are externally accessible. Any additional security provided by the plastic package is ignored for the security of the TOE.

2 Conformance Claims (ASE_CCL)

This chapter is divided into the following sections: "CC Conformance Claim", "PP Claim", and "Conformance Claim Rationale".

2.1 CC Conformance Claim

This Security Target claims conformance to version 2022 of Common Criteria for Information Technology Security Evaluation according to

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, CC:2022 Revision 1, November 2022, CCMB-2022-11-001 [1].
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, CC:2022 Revision 1, November 2022, CCMB-2022-11-002 [2].
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, CC:2022 Revision 1, November 2022, CCMB-2022-11-003 [3].
- Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, CC:2022 Revision 1, November 2022, CCMB-2022-11-004 [4].
- Common Criteria for Information Technology Security Evaluation, Part 5: Predefined packages of security requirements, CC:2022 Revision 1, November 2022, CCMB-2022-11-005 [5].

However, CC Part 4 is not used.

The following methodology will be used for the evaluation:

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology, CEM:2022 Revision 1, November 2022, CCMB-2022-11-006 [6].

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant.

This Security Target claims conformance to the assurance package **EAL4 augmented**. The augmentations to EAL4 are

- AVA_VAN.5 “Advanced methodical vulnerability analysis”
- ALC_DVS.2 “Sufficiency of security measures”
- ALC_FLR.2 “Flaw Reporting Procedures”

Furthermore this Security Target claims conformance to Composite product package, chapter 6, CC part 5[5].

As demonstrated in Section 8, this claim includes or exceeds the minimum assurance level for the Protection Profile identified in Section 2.2.

2.2 PP Claim

This Security Target claims conformance to the following Protection Profiles.

2.2.1 Java Card - Open Configuration (BSI-CC-PP-0099-V3-2024)

This Security Target claims demonstrable conformance to the Java Card Protection Profile - Open Configuration [11], with "Sensitive Result" augmentation package. Other packages of the PP are not supported.

This ST is more restrictive than the PP [11] which Section 2.3 provides a rationale for.

2.2.2 GSMA SGP.25 - eUICC for Consumer and IoT Devices Protection Profile (BSI-CC-PP-0100-V2-2025)

The Security Target claims also strict conformance to the eUICC for Consumer and IoT Devices Protection Profile (Base-PP only) [12].

2.3 Conformance Claim Rationale

2.3.1 TOE Type

The TOE type as stated in Section 1.3 of this ST corresponds to the TOE type of the Java Card Platform PP [11] , implementing the Java Card Specification Version 3.1 [14] [15] [13] with a co-existing eUICC Application, also underpinned by the Java Card and Global Platform Technologies, but accessible via separate, independent communications channels.

2.3.2 Java Card - Open Configuration

2.3.2.1 SPD Statement for Java Card Component

The SPD statement that is presented in Section 4 includes the threats as presented in the PP [11], but also includes additional threats. The additional threats are:

- T.RND
- T.CONFID-UPDATE-IMAGE.LOAD
- T.INTEG-UPDATE-IMAGE.LOAD
- T.UNAUTH-LOAD-UPDATE-IMAGE
- T.INTERRUPT-OSU
- T.CONFIG
- T.COM_EXPLOIT
- T.LIFE_CYCLE
- T.UNAUTHORIZED_CARD_MNGT
- T.INTEG-APPLI-DATA[REFINED]
- T.RESTRICTED-MODE
- T.CONFID-CONT
- T.INTEG-CONT
- T.EXE-CONT
- T.CONT-DOS
- T.CONT-SID

The threat T.RND is taken from the Security IC PP [10]

The threats

- T.CONFID-UPDATE-IMAGE.LOAD
- T.INTEG-UPDATE-IMAGE.LOAD
- T.UNAUTH-LOAD-UPDATE-IMAGE
- T.INTERRUPT-OSU

are included for the OS Update which is additional functionality the PP allows.

The threats

- T.CONFID-CONT
- T.INTEG-CODE
- T.EXE-CONT
- T.CONT-DOS
- T.CONT-SID

are introduced for multiple Guest OSs embeded on the same product inside and outside the TOE boundaries.

The threat T.CONFIG is an additional threat to cover unauthorized modifications and read access of the configuration area in the TOE. It is an addition to the threats defined in the PP [11]. The threat T.RESTRICTED-MODE is included for the Restricted Mode which is additional functionality the PP allows. The threat T.COM_EXPLOIT is included to cover communication channels attacks and it is an addition to the threats in the PP [11].

The threat T.LIFE_CYCLE is included to cover content management attacks and it is an addition to the threats in the PP [11].

The threat T.UNAUTHORIZED_CARD_MNGT refines the threats T.INSTALL and T.DELETION from the PP [11].

The threat T.INTEG-APPLI-DATA[REFINED] refines the threat T.INTEG-APPLI-DATA in the PP [11].

The threat T.RND is to cover the weak random numbers.

Note that the threat T.EXE-CODE-REMOTE is not included, since the TOE does not support Java Card RMI. The Java Card Protection Profile [11] makes the use of Java Card RMI optional.

The SPD statement presented in Section 4, copies the OSP from the PP [11], and adds the following additional OSPs:

- OSP.PROCESS-TOE
- OSP.KEY-CHANGE
- OSP.SECURITY-DOMAINS

The OSP OSP.PROCESS-TOE is introduced for the pre-personalisation feature of the TOE and is an addition to the OSPs in PP [11]. The OSP OSP.KEY-CHANGE is introduced for the SD feature of the TOE and is an addition to the OSPs in PP [11]. The OSP OSP.SECURITY-DOMAINS is introduced for the SD feature of the TOE and is an addition to the OSPs in PP [11].

The SPD statement includes two of the three assumptions from the PP [11]. The assumption A.Deletion is excluded. The Card Manager is part of the TOE and therefore the assumption is no longer relevant. Leaving out the assumption, makes the SPD of this ST more restrictive than the SPD in the PP [11]. As the Card Manager is part of the TOE, it is ensuring that the deletion of applets through the Card Manager is secure, instead of assuming that it is handled by the Card Manager in the environment of the TOE.

Besides the assumptions from the PP [11], the following assumptions are added:

- A.PROCESS-SEC-IC
- A.USE_DIAG
- A.USE_KEYS
- A.APPS-PROVIDER
- A.VERIFICATION-AUTHORITY
- A.TRUSTED-GUESTOS

The assumption A.PROCESS-SEC-IC is taken from the underlying Security IC Platform PP [10].

The assumptions A.USE_DIAG and A.USE_KEYS are included because the Card Manager is part of the TOE and no longer part of the environment.

The assumptions A.APPS-PROVIDER and A.VERIFICATION-AUTHORITY are added because Security Domains from the GlobalPlatform Specification are introduced. All the applets and packages are signed by the APSD and the correctness is verified on the TOE by VASD before the package or applet is installed or loaded. A.APPS-PROVIDER and A.VERIFICATION-AUTHORITY are additions to PP [11] for card content management environment.

The assumptions A.TRUSTED-GUESTOS is included because the Guest Operating Systems that are hosted in external contexts are provided by a trusted actor.

2.3.2.2 Security Objectives Statement for Java Card Component

The statement of security objectives in the ST presented in Section 5 includes all security objectives as presented in the PP [11], but also includes a number of additional security objectives. The additional security objectives are:

- OT.IDENTIFICATION
- OT.CONFID-UPDATE-IMAGE.LOAD
- OT.AUTH-LOAD-UPDATE-IMAGE
- OT.SECURE_LOAD_ACODE
- OT.SECURE_AC_ACTIVATION
- OT.TOE_IDENTIFICATION
- OT.CARD-CONFIGURATION
- OT.ATTACK-COUNTER
- OT.RESTRICTED-MODE
- OT.DOMAIN-RIGHTS
- OT.APPLI-AUTH
- OT.COMM_AUTH
- OT.COMM_INTEGRITY
- OT.COMM_CONFIDENTIALITY
- OT.CONT_SEP
- OT.CONT_PRIV
- OT.CONT_DOS
- OT.RND

The security objectives OT.IDENTIFICATION is part of the security objectives of the Secure Element Hardware (see Section 1.4.1), component of this composite evaluation, but is also relevent for the pre-personalisation feature of the TOE, which is additional functionality the PP allows.

The security objectives

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

are included for the OS Update which is additional functionality the PP allows.

The security objectives

- OT.CONT_SEP
- OT.CONT_PRIV
- OT.CONT_DOS

are included for the protection and separation of the contexts inside and outside the TOE boundaries.

The security objectives OT.CARD-CONFIGURATION is included for the Config Applet which is additional functionality the PP allows.

The security objectives OT.ATTACK-COUNTER and OT.RESTRICTED-MODE are included for the restricted mode which is additional functionality the PP allows.

The security objectives

- OT.DOMAIN-RIGHTS
- OT.APPLI-AUTH
- OT.COMM_AUTH
- OT.COMM_INTEGRITY
- OT.COMM_CONFIDENTIALITY

are objectives for the TOE as the GlobalPlatform API and the definitions for Secure Channel, Security Domains and Card Content Management are used from it.

The ST contains OE.CAP_FILE, OE.VERIFICATION and OE.CODE-EVIDENCE from Security Objectives for the Operational Environment from [11]. Additionally, some of the Security Objectives for the Operational Environment from [11] are listed as TOE Security Objectives in this ST:

- OT.SCP.RECOVERY instead of OE.SCP.RECOVERY
- OT.SCP.SUPPORT instead of OE.SCP.SUPPORT
- OT.SCP.IC instead of OE.SCP.IC
- OT.CARD-MANAGEMENT instead of OE.CARD-MANAGEMENT

OT.SCP.RECOVERY, OT.SCP.SUPPORT, and OT.SCP.IC are objectives for the TOE as the Smart Card Platform belongs to the TOE for this evaluation. OT.CARD-MANAGEMENT is an objective for the TOE as the Card Manager belongs to the TOE for this evaluation. Moving objectives from the environment to the TOE, adds objectives to the TOE without changing the overall objectives. The statement of security objectives is therefore equivalent to the security objectives in the PP [11] to which conformance is claimed.

The security objectives OT.INSTALL, OT.LOAD, and OT.DELETION from the PP [11] are not included since these functionality and objectives are covered by the refined OT.CARD-MANAGEMENT.

The security objective OT.RND is included for the random number quality.

Note that the following objectives are defined as optional in the Protection Profile and are not included in the TOE, therefore are not included in the Security Target:

- O.REMOTE
- O.BIO-MNGT
- O.EXT-MEM
- O.SENSITIVE_ARRAYS_INTEG

The optional O.SENSITIVE_RESULTS_INTEG is included as OT.SENSITIVE_RESULTS_INTEG using the rationale defined in the PP.

The ST introduces eight additional security objectives for the environment. The additional objectives for the environment are:

- OE.USE_DIAG
- OE.USE_KEYS
- OE.PROCESS_SEC_IC
- OE.CONFID-UPDATE-IMAGE.CREATE
- OE.APPS-PROVIDER
- OE.VERIFICATION-AUTHORITY
- OE.KEY-CHANGE
- OE.SECURITY-DOMAINS
- OE.TRUSTED-GUESTOS

The security objective for the environment OE.PROCESS_SEC_IC is from the hardware platform (see Section 1.4.1) that is part of this composite product evaluation. Therefore the statement of security objectives for the environment is equivalent to the statement in the Security IC PP [10].

OE.USE_KEYS and OE.USE_DIAG are included because the Card Manager is part of the TOE and not a security objective for the environment as in PP [11].

The security objective for the environment OE.CONFID-UPDATE-IMAGE.CREATE is to cover the confidentiality during creation and transmission phase of D.UPDATE_IMAGE and therfore partly covers the threats introduced by the update mechanism which is additional functionality.

OE.APPS-PROVIDER and OE.VERIFICATION-AUTHORITY cover trusted actors which enable the creation, distribution and verification of secure applications.

OE.KEY-CHANGE covers the switch to trusted keys for the AP. OE.SECURITY-DOMAINS covers the management of security domains in the context of the GlobalPlatform Specification.

OE.TRUSTED-GUESTOS covers the trusted and secure development of external Guest OSs that are outside the TOE boundaries. The external Guest OSs are secured and not threatening.

The statement of security objectives for the environment is therefore considered to be equivalent to the security objectives in the PP [11] to which conformance is claimed.

2.3.2.3 SFRs Statement for Java Card Component

The Security Functional Requirements Statement copies most SFRs as defined in the PP [11], with the exception of a number of options. For the copied set of SFRs the ST is considered equivalent to the statement of SFRs in the PP [11]. Moreover as requested by the PP [11] the ST adds additional threats, objectives and SFRs to fully cover and describe additional security functionality implemented in the TOE.

The TOE restricts remote access from the CAD to the services implemented by the applets on the card to none, and as a result the SFRs concerning Java Card RMI (FDP_ACF.1[JCRMI], SFRs FDP_IFC.1/JCRMI, FDP_IFF.1/JCRMI, FMT_MSA.1/EXPORT, FMT_MSA.1/REM_REFS, FMT_MSA.3/JCRMI, FMT_SMF.1/JCRMI, FMT_REV.1/JCRMI, and FMT_SMR.1/JCRMI) are not included in the ST. In the PP [11] the use of the Java Card RMI is optional. The TOE does not implement Java Card RMI.

The TOE does not allow external memory access to the services implemented by the applets on the card, and as a result the SFRs concerning "Management of External Memory (EXT-MEM)" (FDP_ACC.1/EXT_MEM, FDP_ACF.1/EXT_MEM, FMT_MSA.1/EXT_MEM, FMT_MSA.3/EXT_MEM and FMT_SMF.1/EXT_MEM) are not included in the ST. In the PP [11] the use of the "Management of External Memory (EXT-MEM)" is optional. The TOE does not implement "Management of External Memory (EXT-MEM)".

The SFR FDP_ITC.2/INSTALLER from the PP [11] is replaced by FDP_ITC.2[CCM] which enforces the Firewall access control policy and the Secure Channel Protocol information flow policy and which is more restrictive than the PACKAGE LOADING information flow control SFP from PP [11].

The set of SFRs that define the card content management mechanism CarG are partly replaced or refined and are considered to be equivalent or more restrictive because of the newly introduced SFPs:

1. Security Domain access control policy
2. Secure Channel Protocol information flow policy

These SFPs provide a concrete and more restrictive implementation of the PACKAGE LOADING information flow control SFP from PP [11] by following the information flow policy defined by Global Platform specifications. The table below lists the SFRs from CarG of PP [11] and their corresponding refinements in this ST.

Table 21. CarG SFRs refinements

SFR from PP [11]	Refinement
FCO_NRO.2/CM	FCO_NRO.2[SC]
FDP_IFC.2/CM	FDP_IFC.2[SC]
FDP_IFF.1/CM	FDP_IFF.1[SC]
FDP_UIT.1/CM	FDP_UIT.1[CCM]
FIA_UID.1/CM	FIA_UID.1[SC]
FMT_MSA.1/CM	FMT_MSA.1[SC]
FMT_MSA.3/CM	FMT_MSA.3[SC]
FMT_SMF.1/CM	FMT_SMF.1[SC]
FMT_SMR.1/CM	FMT_SMR.1[SD]
FTP_ITC.1/CM	FTP_ITC.1[SC]

The following SFRs realize refinements of SFRs from PP [11] and add functionality to the TOE making the Security Functional Requirements Statement more restrictive than the PP [11]:

FDP_ROL.1[CCM], FPT_FLS.1[CCM] and FPT_PHP.3 realize additional security functionality for the card manager which is allowed by the PP [11].

The set of SFRs that define the security domains mechanism as specified by GlobalPlatform, realize refinements of SFRs from PP [11] (see above Table 21) and additional security functionality which is allowed by the PP [11]. This set of SFRs comprise

- FDP_ACC.1[SD]
- FDP_ACF.1[SD]
- FMT_MSA.1[SD]

- FMT_MSA.3[SD]
- FMT_SMF.1[SD]
- FMT_SMR.1[SD]

The set of SFRs that define the secure channel mechanism as specified by GlobalPlatform, realize refinements of SFRs from PP [11] (see above Table 21) and additional security functionality which is allowed by the PP [11]. This set of SFRs comprise

- FCO_NRO.2[SC]
- FDP_IFC.2[SC]
- FDP_IFF.1[SC]
- FMT_MSA.1[SC]
- FMT_MSA.3[SC]
- FMT_SMF.1[SC]
- FIA_UID.1[SC]
- FIA_UAU.1[SC]
- FIA_UAU.4[SC]
- FTP_ITC.1[SC]

The set of SFRs belonging to the CoreG group related to the Java Card API, which are refined multiple times, comprise

- FCS_CKM.1
- FCS_COP.1

The SFRs FAU_SAS.1[SCP], FIA_AFL.1[PIN] and FCS_RNG.1 realize additional security functionality which is allowed by the PP [11].

The set of SFRs that define the Config Applet realize additional security functionality, which is allowed by the PP [11]. This set of SFRs comprise FDP_IFC.2[CFG], FDP_IFF.1[CFG], FIA_UID.1[CFG], FMT_MSA.1[CFG], FMT_MSA.3[CFG], FMT_SMF.1[CFG], FMT_SMR.1[CFG].

The set of SFRs that define the OS Update realize additional security functionality, which is allowed by the PP [11]. This set of SFRs comprise 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] and FPT_FLS.1[OSU].

The set of SFRs that define the Restricted Mode realize additional security functionality, which is allowed by the PP [11]. This set of SFRs comprise FDP_ACC.2[RM], FDP_ACF.1[RM], FMT_MSA.3[RM], FMT_MSA.1[RM], FMT_SMF.1[RM], FIA_UID.1[RM] and FIA_UAU.1[RM].

The set of SFRs that define the Context Separation realize additional security functionality, which is allowed by the PP [11]. This set of SFRs comprise FDP_ACC.2[CONTSEP], FDP_ACF.1[CONTSEP], FMT_MSA.3[CONTSEP], FMT_MSA.1[CONTSEP], FMT_SMF.1[CONTSEP], FMT_SMR.1[CONTSEP] and FIA_UID.1[CONTSEP].

2.3.3 Conformance Claim Rationale for eUICC component

2.3.3.1 SPD Statement for eUICC Component

The Security Problem Definition of the eUICC component is the same as in eUICC PP [12], no item have been added, removed or modified.

2.3.3.2 Security Objectives Statement for eUICC Component

The Security Objectives for the TOE and its environment of the eUICC component is the same as in the eUICC PP [12] with some exclusions due to the overlap with the JCOP objectives defined in [11]:

- OE.IC.SUPPORT from eUICC PP [12] refines the objective OE.SCP.SUPPORT from the JCOP PP [11] with its own specific needs; it is then directly met by the coverage of the JCOP objective OE.SCP.SUPPORT.
- OE.IC.RECOVERY from eUICC PP [12] refines the objective OE.SCP.RECOVERY from the JCOP PP [11] with its own specific needs; it is then directly met by the coverage of the JCOP objective OE.SCP.RECOVERY.
- OE.RE.PRE-PPI from eUICC PP [12] is a request on the Runtime Environment and is met by the JCOP component objectives related to the threats T.DELETION and T.INSTALL defined in [11].
- OE.RE.SECURE-COMM from eUICC PP [12] is a request on the Runtime Environment and is met by the JCOP component objectives OT.FIREWALL and those related to the threats T.CONFID-APPLI-DATA and T.INTEG-APPLI-DATA defined in [11].
- OE.RE.API from eUICC PP [12] is a request on the Runtime Environment and is met by the JCOP component objectives related to the threats T.CONFID-JCS-CODE, T.INTEG-JCS-CODE, T.CONFID-JCS-DATA and T.INTEG-JCS-DATA defined in [11].
- OE.RE.DATA-CONFIDENTIALITY from eUICC PP [12] is a request on the Runtime Environment and is met by the JCOP component objectives related to the threat T.CONFID-APPLI-DATA defined in [11].
- OE.RE.DATA-INTEGRITY from eUICC PP [12] is a request on the Runtime Environment and is met by the JCOP component objectives related to the threats T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD defined in [11].
- OE.RE.CODE-EXE from eUICC PP [12] is a request on the Runtime Environment and is met by the JCOP component objectives related to the threats T.EXE-CODE.1, T.EXE-CODE.2 and T.NATIVE defined in [11].
- OE.RE.IDENTITY from eUICC PP [12] is a request on the Runtime Environment and is met by the JCOP component objectives related to the threats T.SID.1 and T.SID.2 defined in [11].
- OE.IC.PROOF_OF_IDENTITY from eUICC PP [12] is a request on the IC component and is met as described in Section 1.4.1.

2.3.3.3 Security Functional Requirements Statement for eUICC Component

The Security Functional Requirements for the eUICC component are copied from the eUICC PP [12], non have been added, removed, or modified.

3 Security Aspects

This chapter describes the main security issues of the Java Card System and its environment addressed in this ST, called "security aspects", in a CC-independent way. In addition to this, the security aspects 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. The description is based on [11].

3.1 Confidentiality

SA.CONFID-UPDATE-IMAGE — **Confidentiality of Update Image**
The update image must be kept confidential. This concerns the non disclosure of the update image in transit to the card.

SA.CONFID-APPLI-DATA — **Confidentiality of Application 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. This must also consider that applets receiving an ArrayView must not be able to access beyond the boundaries and access rights defined during the creation of the ArrayView.

SA.CONFID-JCS-CODE — **Confidentiality of Java Card System 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.

SA.CONFID-JCS-DATA — **Confidentiality of Java Card System 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

SA.INTEG-UPDATE-IMAGE — **Integrity of Update Image**
The update image must be protected against unauthorized modification. This concerns the modification of the image in transit to the card.

SA.INTEG-APPLI-CODE — **Integrity of Application 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.

SA.INTEG-APPLI-DATA **Integrity of Application 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. This must also consider the Integrity of data accessed through the use of ArrayView.

SA.INTEG-JCS-CODE **Integrity of Java Card System 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.

SA.INTEG-JCS-CODE **Integrity of Java Card System 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 Execution

SA.EXE-APPLI-CODE **Execution of Application 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 ([16])
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).

SA.EXE-JCS-CODE **Execution of Java Card System 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. This concerns:

1. invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([16])
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 SA.NATIVE.

SA.FIREWALL **Firewall**

The Firewall shall ensure controlled sharing of class instances[1], 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.

SA.NATIVE	**Native Code Execution** 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.

[1] This concerns in particular the arrays, which are considered as instances of the Object class in the Java programming language.

3.4 Bytecode Verification

SA.VERIFICATION	**Bytecode 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. The latest available version of the verifier should be used.

3.5 Card Management

SA.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.
SA.INSTALL	**Installation** 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 cancelled (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.

SA.SID **Subject Identification**

1. 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). A change of identity, especially standing for an administrative role (like an applet impersonating the Java Card RE), is a severe violation of the 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.

SA.OBJ-DELETION **Object Deletion**

1. Deallocation of objects should not introduce security holes in the form of references pointing to memory zones that are not 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.

SA.DELETION **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

SA.ALARM **Alarm**

The TOE shall provide appropriate feedback upon detection of a potential security 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.

SA.OPERATE **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.

SA.RESOURCES **Resources**

The TOE controls the availability of resources for the applications and enforces quotas and limitations 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.

SA.CIPHER **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.

SA.KEY-MNGT **Key Management**

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.

SA.PIN-MNGT **PIN Management**

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. Enhanced protection of PIN’s security attributes (state, try counter ...) in confidentiality and integrity.

SA.SCP **Smart Card Platform**

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 a ”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.
7. Finally, it is required that the IC is designed in accordance with a well-defined set of policies and standards (for instance, those specified in [10]), 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.

SA.TRANSACTION **Transaction**

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

3.7 Config Applet

SA.CONFIG-APPLET **Config Applet**

The Config Applet is a JCOP functionality which allows to:

1. Read and modify configuration items in the configuration area of the TOE,
2. Disable Access to configuration item.

3.8 OS Update

SA.OSU **OS Update**

The SystemOS allows to update JCOP eUICC, FlashOS, CommonOS, Shared code, SMK, and the SystemOS itself. It ensures that only valid updates can be installed on the TOE.

3.9 Restricted Mode

SA.RM	**Restricted Mode** If the Attack Counter reaches its limit the TOE goes into Restricted Mode. In this mode it is possible to perform a limited set of functions, like authenticate against the ISD, reset the Attack Counter or read logging information. The GlobalPlatform state of the ISD is not changed.

3.10 Context Separation

SA.CONTEXT-SEPARATION	**Context Separation** The hardware enforced Context Separation ensures that all the operating systems hosted on the secure element are running in dedicated contexts. The external operating systems that are outside the boudaries of the TOE (typically eSE, CommOS, JCOP xxx) cannot interact (read/write data, fetch unshared code, impersonate) with the TOE in an uncontrolled and unauthorized way. The communications between the TOE and external Operating systems is allowed through dedicated communication channels under the control of the SMK.

4 Security Problem Definition (ASE_SPD)

The following sections list the assets, threats, organisational security policies and assumptions of the TOE.

These are listed separately for each component to allow tracing of the conformance to the corresponding Protection Profile.

4.1 SPD for Java Card System

4.1.1 Assets for Java Card System

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 concerning the threats are given in Section 4.1.2 hereafter.

Assets have to be protected, some in terms of confidentiality and some in terms of integrity or both integrity and confidentiality. These assets might get compromised by the threats that the TOE is exposed to.

The assets of the Security IC Embedded Software 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). This definition of grouping is taken from Section 5.1 of PP [11] .

4.1.1.1 User data

Table 22. User Data Assets

D.APP_CODE	The code of the applets and libraries loaded on the card. To be protected from unauthorized modification.
D.APP_C_DATA	Confidential 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, a static field, a local variable of the currently executed method, or a position of the operand stack. 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.APSD_KEYS	Refinement of D.APP_KEYS of [11]. Application Provider Security Domains cryptographic keys needed to establish secure channels with the AP. These keys can be used to load and install applications on the card if the Security Domain has the appropriate privileges. To be protected from unauthorized disclosure and modification.
D.ISD_KEYS	Refinement of D.APP_KEYS of [11]. Issuer Security Domain cryptographic keys needed to perform card management operations on the card. To be protected from unauthorized disclosure and modification.

Table 22. User Data Assets*...continued*

D.VASD_KEYS	Refinement of D.APP_KEYS of [11]. Verification Authority Security Domain cryptographic keys needed to verify applications Mandated DAP signature. 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.

4.1.1.2 TSF data

Table 23. TSF Data Assets

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 JCVM, 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 JCRE, 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.UPDATE_IMAGE	Can be an update for JCOP 7.x OS and SystemOS. It is sent to the TOE, received by the SystemOS. It includes executable code, configuration data, as well as a Sequence Number (Received Sequence Number) and Image Type. To be protected from unauthorized disclosure and modification. It is decrypted using the Package Decryption Key and its signature is verified using the Verification Key. Is also referred to as Additional Code, see [9].
D.CONFIG_ITEM	A configuration that can be changed using the Config Applet.
D.RESTRICTED_MODE_STATE	The Restricted Mode is entered when the attack counter reaches its limit (the Attack Counter is incremented when a potential attack is detected and decrements after sufficient time in a powered state without detecting any new attacks). Once the Restricted Mode is entered, it shall not be possible to exit without the approval of authorized users.
D.TOE_IDENTIFIER	Identification Data to identify the TOE.

4.1.2 Threats for Java Card System

The threats for the Security IC Embedded Software are listed below. The definition of the grouping is taken from Section 5.2 of PP [11].

4.1.2.1 Confidentiality

T.CONFID-APPLI-DATA	**Confidentiality of Application Data** The attacker executes an application to disclose data belonging to another application. See SA.CONFID-APPLI-DATA for details. Directly threatened asset(s): D.APP_C_DATA, D.PIN and D.APP_KEYS.
T.CONFID-JCS-CODE	**Confidentiality of Java Card System Code** The attacker executes an application to disclose the Java Card System code. See SA.CONFID-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.
T.CONFID-JCS-DATA	**Confidentiality of Java Card System Data** The attacker executes an application to disclose data belonging to the Java Card System. See SA.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	**Integrity of Application Code** The attacker executes an application to alter (part of) its own code or another application's code. See SA.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.
T.INTEG-APPLI-CODE.LOAD	**Integrity of Application 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 SA.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.
T.INTEG-APPLI-DATA [REFINED]	**Integrity of Application Data** The attacker executes an application to alter (part of) another application's data. See SA.INTEG-APPLI-DATA for details. Directly threatened asset(s): D.APP_I_DATA, D.PIN, D.APP_KEYS, D.ISD_KEYS, D.VASD_KEYS and D.APSD_KEYS. This threat is a refinement of the Threat T.INTEG-APPLI-DATA from [11].
T.INTEG-APPLI-DATA.LOAD	**Integrity of Application 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 SA.INTEG-APPLI-DATA for details. Directly threatened asset(s): D.APP_I_DATA and D.APP_KEYS.
T.INTEG-JCS-CODE	**Integrity of Java Card System Code** The attacker executes an application to alter (part of) the Java Card System code. See SA.INTEG-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.
T.INTEG-JCS-DATA	**Integrity of Java Card System Data** The attacker executes an application to alter (part of) Java Card System or API data. See SA.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 **Subject Identification 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 SA.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 **Subject Identification 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 SA.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 **Code Execution 1**

An applet performs an unauthorized execution of a method. See SA.EXE-JCS-CODE and SA.EXE-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

T.EXE-CODE.2 **Code Execution 2**

An applet performs an execution of a method fragment or arbitrary data. See SA.EXE-JCS-CODE and SA.EXE-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

T.NATIVE **Native Code Execution**

An applet executes a native method to bypass a TOE Security Function such as the firewall. See SA.NATIVE for details. Directly threatened asset(s): D.JCS_DATA.

4.1.2.5 Denial of Service

T.RESOURCES **Consumption of Resources**

An attacker prevents correct operation of the Java Card System through consumption of some resources of the card: RAM or NVRAM. See SA.RESOURCES for details. Directly threatened asset(s): D.JCS_DATA.

4.1.2.6 Card Management

T.UNAUTHORIZED_CARD_MNGT — **Unauthorized Card Management**

The attacker performs unauthorized card management operations (for instance impersonates one of the actor represented on the card) in order to take benefit of the privileges or services granted to this actor on the card such as fraudulent:

- load of a CAP file
- installation of a CAP file
- extradition of a CAP file or an applet
- personalization of an applet or a Security Domain
- deletion of a CAP file or an applet
- privileges update of an applet or a Security Domain

Directly threatened asset(s): D.ISD_KEYS, D.APSD_KEYS, D.APP_C_DATA, D.APP_I_DATA, D.APP_CODE, D.SEC_DATA, and D.CARD_MNGT_DATA (any other asset may be jeopardized should this attack succeed, depending on the virulence of the installed application).

This security objective is a refinement of the Threats T.INSTALL and T.DELETION from [11].

T.COM_EXPLOIT — **Communication Channel Remote Exploit**

An attacker remotely exploits the communication channels established between a third party and the TOE in order to modify or disclose confidential data. All assets are threatened.

T.LIFE_CYCLE — **Life Cycle**

An attacker accesses to an application outside of its expected availability range thus violating irreversible life cycle phases of the application (for instance, an attacker repersonalizes the application). Directly threatened asset(s): D.APP_I_DATA, D.APP_C_DATA, and D.CARD_MNGT_DATA.

4.1.2.7 Services

T.OBJ-DELETION — **Object 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 SA.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 — **Physical Tampering**

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 SA.SCP, and all aspects related to confidentiality and integrity of code and data.

4.1.2.9 Random Numbers

T.RND	**Deficiency of Random Numbers** An attacker may predict or obtain information about random numbers generated by the TOE for instance because of a lack of entropy of the random numbers provided. An attacker may gather information about the produced random numbers which might be a problem because they may be used for instance to generate cryptographic keys. Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the TOE without specific knowledge about the TOE's generator. Malfunctions or premature ageing are also considered which may assist in getting information about random numbers.

4.1.2.10 Config Applet

T.CONFIG	**Unauthorized configuration** The attacker tries to change configuration items without authorization. Directly threatened asset(s): D.CONFIG_ITEM.

4.1.2.11 OS Update

T.CONFID-UPDATE-IMAGE.LOAD	**Confidentiality of 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 SA.CONFID-UPDATE-IMAGE for details. Directly threatened asset(s): D.UPDATE_IMAGE, D.JCS_CODE, and D.JCS_DATA.
T.UNAUTH-LOAD-UPDATE-IMAGE	**Load unauthorized version of Update Image** The attacker tries to upload an unauthorized Update Image. Directly threatened asset(s): D.JCS_CODE, D.JCS_DATA, D.UPDATE_IMAGE.
T.INTEG-UPDATE-IMAGE.LOAD	**Integrity of 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 SA.INTEG-UPDATE-IMAGE for details. Directly threatened asset(s): D.UPDATE_IMAGE, D.JCS_CODE, and D.JCS_DATA.
T.INTERRUPT-OSU	**OS Update procedure interrupted** 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.

4.1.2.12 Restricted Mode

T.RESTRICTED-MODE	**UNOTHAURIZED ESCAPE FROM RESTRICTED MODE** The attacker tries to exit the Restricted Mode without authorization. Directly threatened asset: D.RESTRICTED_MODE_STATE.

4.1.2.13 Context Separation

T.CONFID-CONT	**Disclosure of Context data and code** An attacker from one context discloses data or code belonging to another context (like Application Data, Java Card System Code, Java Card System Data). This threat extends the threats T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.CONFID-JCS-CODE to multiple contexts.
T.INTEG-CONT	**Modification of Context data and code** An attacker from one context alters data or code belonging to another context (like application code, application data, transmitted application package, Java Card System code, Java Card System Data). This threat extends the threats 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 to multiple contexts.
T.EXE-CONT	**Code execution from another context** An attacker from one context performs an unauthorized execution of a method, method fragment, arbitrary data, or native code of another context. This threat extends the threats T.EXE-CODE.1, T.EXE-CODE.2, T.NATIVE to multiple contexts.
T.CONT-DOS	**Deny of service between Contexts** An attacker from one context prevents the correct execution of the SMK or another context through consumption of some critical resources of the TOE. This threat extends the threats T.RESOURCES to multiple contexts.
T.CONT-SID	**Subject Identification between Contexts** An Attacker impersonates one context with an OS running in another context. This Threat extends the threats T.SID.1, T.SID.2 to multiple contexts.

4.1.3 OSPs for Java Card System

The organizational security policies to be enforced with respect to the TOE environment that are related to the Security IC Embedded Software are listed below. The definition of the grouping is taken from Section 5.3 of PP [11].

OSP.VERIFICATION	**File 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 SA.VERIFICATION for details. If the application development guidance provided by the platform developer contains recommandations related to the isolation property of the platform, this policy shall also ensure that the verification authority checks that these recommandations are applied in the application code.
OSP.PROCESS-TOE	**Identification of the TOE** An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this identification.
OSP.KEY-CHANGE	**Security Domain Keys Change** The AP shall change its initial security domain keys (APSD) before any operation on its Security Domain.

OSP.SECURITY-DOMAINS	**Security Domains** Security domains can be dynamically created, deleted and blocked during usage phase in post-issuance mode.

4.1.4 Assumptions for Java Card System

The assumptions for the Security IC Embedded Software are listed below. The definition of the grouping is taken from Section 5.4 of PP [11].

Note that the assumption A.DELETION as defined in PP [11] is excluded. The Card Manager is part of the TOE and therefore the assumption is no longer relevant.

A.CAP_FILE	**Applets without Native Methods** CAP Files loaded post-issuance do not contain native methods. The Java Card specification explicitly ”does not include support for native methods” ([14]) outside the API.
A.VERIFICATION	**Bytecode 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. The latest available version of the verifier should be used.
A.USE_DIAG	**Usage of TOE’s Secure Communication Protocol by OE** It is assumed that the operational environment supports and uses the secure communication protocols offered by the TOE.
A.USE_KEYS	**Protected Storage of Keys Outside of TOE** It is assumed that the keys which are stored outside the TOE and which are used for secure communication and authentication between Smart Card and terminals are protected for confidentiality and integrity in their own storage environment. This is especially true for D.APSD_KEYS, D.ISD_KEYS, and D.VASD_KEYS. **Info**: This is to assume that the keys used in terminals or systems are correctly protected for confidentiality and integrity in their own environment, as the disclosure of such information which is shared with the TOE but is not under the TOE control, may compromise the security of the TOE.
A.PROCESS-SEC-IC	**Protection during Packaging, Finishing and Personalisation** It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that the Phases after TOE Delivery are assumed to be protected appropriately. The assets to be protected are: The information and material produced and/or processed by the Security IC Embedded Software Developer in Phase 1 and by the Composite Product Manufacturer can be grouped as follows: 1. the Security IC Embedded Software including specifications, implementation and related documentation, 2. pre-personalisation and personalisation data including specifications of formats and memory areas, test related data, 3. the User Data and related documentation, and 4. material for software development support as long as they are not under the control of the TOE Manufacturer.

A.APPS-PROVIDER	**Application Provider** The AP is a trusted actor that provides basic or secure applications. He is responsible for his security domain keys (D.APSD_KEYS). **Info**: An AP generally refers to the entity that issues the application. For instance it can be a financial institution for a payment application such as EMV or a transport operator for a transport application.
A.TRUSTED-GUESTOS	**Trusted Guest OS** The external Guest OS provider is a trusted actor that is reponsible for the security and trust of his OS. **Info**: This mitigates the risk of an hostile external guest OS.
A.VERIFICATION-AUTHORITY	**Verification Authority** The VA is a trusted actor who is able to verify bytecode of an application loaded on the card, guarantee and generate the digital signature attached to an application and ensure that its public key for verifying the application signature is on the TOE. **Info**: As a consequence, it guarantees the success of the application validation upon loading.

4.2 SPD for eUICC

The Security Problem Definition for the eUICC component of the TOE is strictly compliant with the Security Problem Definition described in the eUICC PP [12].

5 Security Objectives

5.1 Security Objectives for the TOE

5.1.1 Security Objectives for Java Card System

5.1.1.1 Identification

OT.SID	**Subject Identification** The TOE shall uniquely identify every subject (applet, or CAP file) before granting it access to any service.

5.1.1.2 Execution

OT.FIREWALL	**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 SA.FIREWALL for details.
OT.GLOBAL_ ARRAYS_CONFID	**Confidentiality of Global Arrays** The TOE shall ensure that the APDU buffer that is shared by all applications is always cleaned 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 cleaned after the return from the install method.
OT.GLOBAL_ ARRAYS_INTEG	**Integrity of Global Arrays** The TOE shall ensure that only the currently selected applications may have a write access to the APDU buffer and the global byte array used for the invocation of the install method of the selected applet.
OT.ARRAY_VIEW_ CONFID	**Confidentiality of Array View** 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.
OT.ARRAY_VIEW_ INTEG	**Integrity of Array View** 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.
OT.SENSITIVE_ RESULTS_INTEG	**Sensitive Result** The TOE shall ensure that the sensitive results (com.nxp.id.jcopx.security.SensitiveResultX) of sensitive operations executed by applications through the Java Card API are protected in integrity specifically against physical attacks.
OT.NATIVE	**Native Code** 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 SA.NATIVE for details.

OT.OPERATE	**Correct Operation** The TOE must ensure continued correct operation of its security functions. See SA.OPERATE for details.
OT.REALLOCATION	**Secure Re-Allocation** 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.
OT.RESOURCES	**Resources availability** The TOE shall control the availability of resources for the applications. See SA.RESOURCES for details.

5.1.1.3 Services

OT.ALARM	**Alarm** The TOE shall provide appropriate feedback information upon detection of a potential security violation. See SA.ALARM for details.
OT.CIPHER	**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 SA.CIPHER for details.
OT.KEY-MNGT	**Key Management** The TOE shall provide a means to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys. See SA.KEY-MNGT.
OT.PIN-MNGT	**PIN Management** The TOE shall provide a means to securely manage PIN objects. See SA.PIN-MNGT for details. AppNote: 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.
OT.TRANSACTION	**Transaction** The TOE must provide a means to execute a set of operations atomically. See SA.TRANSACTION for details.

5.1.1.4 Object Deletion

OT.OBJ-DELETION	**Object Deletion** The TOE shall ensure the object deletion shall not break references to objects. See SA.OBJ-DELETION for further details.

5.1.1.5 Applet Management

OT.APPLI-AUTH	**Application Authentication** The card manager shall enforce the application security policies established by the card issuer by requiring application authentication during application loading on the card. This security objective is a refinement of the Security Objective O.LOAD from [11]. AppNote: Each application loaded onto the TOE has been signed by a VA. The VA will guarantee that the security policies established by the card issuer on applications are enforced. For example this authority (DAP) or a third party (Mandated DAP) can be present on the TOE as a Security Domain whose role is to verify each signature at application loading.
OT.DOMAIN-RIGHTS	**Domain Rights** The Card issuer shall not get access or change personalized AP Security Domain keys which belong to the AP. Modification of a Security Domain keyset is restricted to the AP who owns the security domain. AppNote: APs have a set of keys that allows them to establish a secure channel between them and the platform. These keys sets are not known by the TOE issuer. The security domain initial keys are changed before any operation on the SD (OE.KEY-CHANGE).
OT.COMM_AUTH	**Communication Mutual Authentication** The TOE shall authenticate the origin of the card management requests that the card receives, and authenticate itself to the remote actor.
OT.COMM_ INTEGRITY	**Communication Request Integrity** The TOE shall verify the integrity of the card management requests that the card receives.
OT.COMM_ CONFIDENTIALITY	**Communication Request Confidentiality** The TOE shall be able to process card management requests containing encrypted data.

5.1.1.6 Card Management

OT.CARD-MANAGEMENT	**Card Management** The TOE shall provide card management functionalities (loading, installation, extradition, deletion of applications and GP registry updates) in charge of the life cycle of the whole device and installed applications (applets). The card manager, the application with specific rights responsible for the administration of the smart card, shall control the access to card management functions. It shall also implement the card issuer's policy on card management. The Security Objective from [11] for the environment OE.CARD-MANAGEMENT is listed as TOE Security Objective OT.CARD-MANAGEMENT for the TOE as the Card Manager belongs to the TOE for this evaluation. This security objective is a refinement for the Security Objectives O.INSTALL, O.LOAD, and O.DELETION from [11]. Thus, the following objectives are also covered: • The TOE shall ensure that the installation of an applet performs as expected (See SA.INSTALL for details). • The TOE shall ensure that the loading of a package into the card is secure. • The TOE shall ensure that the deletion of a package from the TOE is secure. AppNote: The card manager will be tightly connected in practice with the rest of the TOE, which in return shall very likely rely on the card manager for the effective enforcement of some of its security functions. The mechanism used to ensure authentication of the TOE issuer, that manages the TOE, or of the Service Providers owning a Security Domain with card management privileges is a secure channel. This channel will be used afterwards to protect commands exchanged with the TOE in confidentiality and integrity. The platform guarantees that only the ISD or the Service Providers owning a Security Domain with the appropriate privilege (Delegated Management) can manage the applications on the card associated with its Security Domain. This is done accordingly with the card issuer's policy on card management. The actor performing the operation must beforehand authenticate with the Security Domain. In the case of Delegated Management, the card management command will be associated with an electronic signature (GlobalPlatform token) verified by the ISD before execution. The Security Objective from [11] for the environment OE.CARD-MANAGEMENT is listed as TOE Security Objective OT.CARD-MANAGEMENT for the TOE as the Card Manager belongs to the TOE for this evaluation. This security objective is a refinement for the Security Objectives O.INSTALL, O.LOAD, and O.DELETION from [11]. Thus, the following AppNote applicable to O.DELETION applies also: • 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 packages 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.

5.1.1.7 Smart Card Platform

OT.SCP.IC	**IC Physical Protection** The IC shall provide all security features against physical attacks. This security objective for the environment refers to the point (7) of the security aspect SA.SCP. AppNote: The Security Objectives from [11] for the environment OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are listed as TOE Security Objectives (OT.SCP.RECOVERY, OT.SCP.SUPPORT, and OT.SCP.IC) in this section as the IC belongs to the TOE for this evaluation.
OT.SCP.RECOVERY	**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 environment refers to the security aspect SA.SCP AppNote: The Security Objectives from [11] for the environment OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are listed as TOE Security Objectives (OT.SCP.RECOVERY, OT.SCP.SUPPORT, and OT.SCP.IC) for the TOE in this section as the Smart Card Platform belongs to the TOE for this evaluation.
OT.SCP.SUPPORT	**SCP Support** The SCP shall support the TSFs of the TOE. This security objective refers to the security aspects 2, 3, 4 and 5 of SA.SCP AppNote: The Security Objectives from [11] for the environment OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are listed as TOE Security Objectives (OT.SCP.RECOVERY, OT.SCP.SUPPORT, and OT.SCP.IC) for the TOE in this section as the Smart Card Platform belongs to the TOE for this evaluation.
OT.IDENTIFICATION	**TOE identification** The TOE must provide means to store Initialization Data and Pre-personalization Data in its non-volatile memory. The Initialization Data (or parts of them) are used for TOE identification.

5.1.1.8 Random Numbers

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

5.1.1.9 OS Update Mechanism

OT.CONFID-UPDATE-IMAGE.LOAD	**Confidentiality of 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 decrypting the image shall be kept confidential.
OT.AUTH-LOAD-UPDATE-IMAGE	**Authorization of Update Image - Load** The TOE shall ensure that it is only possible to load an authorized image.

The following Security Objectives have been added to comply to JIL "Security requirements for post-delivery code loading" [9].

OT.SECURE_LOAD_ACODE	**Secure loading of the Additional Code** 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.
OT.SECURE_AC_ACTIVATION	**Secure activation of the Additional Code** 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.
OT.TOE_IDENTIFICATION	**Secure identification of the TOE** 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.

5.1.1.10 Config Applet

OT.CARD-CONFIGURATION	**Card Configuration** The TOE shall ensure that the customer can only configure customer configuration items and that NXP can configure customer and NXP configuration items. Additionally, the customer can only disable the customer configuration and NXP can disable customer and NXP configuration.

5.1.1.11 Restricted Mode

OT.ATTACK-COUNTER	**Attack Counter Reset** The TOE shall ensure that the Attack Counter can only be decremented in a controled way, either by reset from user with appropriate authorization, or by regular self decrementation after time elapse.
OT.RESTRICTED-MODE	**Restricted Mode** The TOE shall ensure that in Restricted Mode all operations return an error except for the limited set of commands that are allowed by the TOE when in Restricted Mode.

5.1.1.12 Context Separation

OT.CONT_SEP	**Context separation** The TOE shall prevent a software running in one context from unauthorized access (read/write/execute) to another context memory, peripherals or resources. Any exchange between contexts (like OS or Services) shall be controlled by the TOE.

OT.CONT_PRIV	**Privileges management** The TOE shall ensure that its kernel, also known as SMK, has the highest privileges with regard to any other software-parts running in contexts inside or outside the TOE boundaries.
OT.CONT_DOS	**Deny of Service** The TOE shall prevent Deny of Service by managing the scheduling of the contexts according to their priorities. Only SMK shall be able to define and modify the priorities.

5.1.2 Security Objectives for eUICC

The Security Objectives for the eUICC component of the TOE is strictly compliant with the Security Objectives for the TOE described in the eUICC PP [12].

5.2 Security Objectives for the Operational Environment

5.2.1 Security Objectives for the Operational Environment of Java Card System

OE.CAP_FILE	**Applet** No CAP file loaded post-issuance shall contain native methods.
OE.VERIFICATION	**Bytecode 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. The latest available version of the verifier should be used. See SA.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	**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 verifications 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 Protection Profile. 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.APPS-PROVIDER	**Application Provider** The AP shall be a trusted actor that provides applications. The AP is responsible for its security domain keys.
OE.TRUSTED-GUEST OS	**Trusted Guest OS** The external Guest OS provider is a trusted actor that is reponsible for the security and trust of his OS.
OE.VERIFICATION-AUTHORITY	**Verification Authority** The VA should be a trusted actor who is able to verify bytecode of an application loaded on the card, guarantee and generate the digital signature attached to an application and ensure that its public key for verifying the application signature is on the TOE.
OE.KEY-CHANGE	**Security Domain Key Change** The AP must change its security domain initial keys before any operation on it.
OE.SECURITY-DOMAINS	**Security Domains** Security domains can be dynamically created, deleted and blocked during usage phase in post-issuance mode.
OE.USE_DIAG	**Secure TOE communication protocols** Secure TOE communication protocols shall be supported and used by the environment.
OE.USE_KEYS	**Protection of OPE keys** During the TOE usage, the terminal or system in interaction with the TOE, shall ensure the protection (integrity and confidentiality) of their own keys by operational means and/or procedures.
OE.PROCESS_SEC_IC	**Protection during composite product manufacturing** Security procedures shall be used after TOE Delivery up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that Phases after TOE Delivery up to the end of Phase 6 must be protected appropriately.
OE.CONFID-UPDATE-IMAGE.CREATE	**Confidentiality of Update Image - CREATE** The off-card Update Image Creator ensures that the image is signed and transferred encrypted to the device and is not disclosed during the creation and transfer. The keys used for signing and encrypting the image are kept confidential.

5.2.2 Security Objectives for the Operational Environment of eUICC

The Security Objectives for the Environment of the eUICC component of the TOE is strictly compliant with the Security Objectives for the Environment described in the eUICC PP [12].

5.3 Security Objectives Rationale

In this section each threat, Organizational Security Policy, and assumption identified in Section 4 is traced to the security objectives with a rationale.

The security objectives for the TOE defined in Section 5.1 are traced back to the threats countered by them, and to the organisational security policies enforced by them. The security objectives for the operational environment defined in Section 5.2 are traced back to the assumptions they uphold.

5.3.1 Security Objective Rationale related to the Java Card System

5.3.1.1 Rationale for Threats

This chapter provides the Security Objectives rationale for Java Card System.

The mappings in Section 5.3.1.1.1 (Confidentiality), Section 5.3.1.1.2 (Integrity), Section 5.3.1.1.3 (Identity Usurpation), Section 5.3.1.1.4 (Unauthorized Execution) and Section 5.3.1.1.5 (Denial Of Service) are not augmented with the Context Separation Objectives in order to stay eSE context centric and to maintain modularity, clarity and alignment with the JavaCard Protection Profile [11]. The inter-context protection is then covered in Section 5.3.1.1.13 which also covers, by extension, the threats of the above mentioned chapter with regards to external contexts.

5.3.1.1.1 Confidentiality

T.CONFID-UPDATE-IMAGE.LOAD

Objective	Rationale
OT.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 transfered in plain and that the keys are kept secret.

T.CONFID-APPLI-DATA

Objective	Rationale
OT.SID	Counters this threat by providing correct identification of applets.
OT.FIREWALL	Counters this threat by providing the Java Card Virtual Machine Firewall as specified in [15].
OT.GLOBAL_ARRAYS_CONFID	Counters this threat by preventing the disclosure of the information stored in the APDU buffer.
OT.ARRAY_VIEWS_CONFID	Counters this threat by preventing the disclosure of the information shared by applets using array views.
OT.OPERATE	Counters the threat by ensuring that the firewall, which is dynamically enforced, shall never stop operating.
OT.REALLOCATION	Counters this threat by preventing any attempt to read a piece of information that was previously used by an application but has been logically deleted. It states that any information that was formerly stored in a memory block shall be cleared before the block is reused.
OT.ALARM	Counters this threat by obtaining clear warning and error messages from the firewall, which is a software tool automating critical controls, so that the appropriate countermeasure can be taken.
OT.CIPHER	Contributes to counter this threat by protecting the data shared or information communicated between applets and the CAD using cryptographic functions.
OT.KEY-MNGT	Counters this threat by providing appropriate management of keys, PIN's which are particular cases of an application's sensitive data.
OT.PIN-MNGT	Counters this threat by providing appropriate management of keys, PIN's which are particular cases of an application's sensitive data.

Objective	Rationale
OT.TRANSACTION	Counters this threat by providing appropriate management of keys, PIN's which are particular cases of an application's sensitive data.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OT.SCP.RECOVERY	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OT.SCP.SUPPORT	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.

T.CONFID-JCS-CODE

Objective	Rationale
OT.NATIVE	Counters this threat by ensuring that no native applications can be run to modify a piece of code.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.

T.CONFID-JCS-DATA

Objective	Rationale
OT.SID	Counters this threat by providing correct identification of applets.
OT.FIREWALL	Contributes to counter this threat by providing means of separating data.
OT.OPERATE	Counters the threat by ensuring that the firewall, which is dynamically enforced, shall never stop operating.
OT.ALARM	Contributes to counter this threat by obtaining clear warning and error messages from the firewall, which is a software tool automating critical controls, so that the appropriate countermeasure can be taken.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OT.SCP.RECOVERY	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OT.SCP.SUPPORT	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.

5.3.1.1.2 Integrity

T.INTEG-UPDATE-IMAGE.LOAD

Objective	Rationale
OT.SECURE_LOAD_ACODE	Counters the threat directly by ensuring the authenticity and integrity of D.UPDATE_IMAGE.

T.INTEG-APPLI-CODE

Objective	Rationale
OT.NATIVE	Counters this threat by ensuring that no native code can be run to modify a piece of code.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode. 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.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OE.CODE-EVIDENCE	The objective OE.CODE-EVIDENCE contributes to counter this threat by ensuring that integrity and authenticity evidences exist for the application code loaded into the platform.

T.INTEG-APPLI-CODE.LOAD

Objective	Rationale
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions such as the installation, update or deletion of applets.
OE.CODE-EVIDENCE	Contributes to counter 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.
OT.APPLI-AUTH	Counters this threat by ensuring that the loading of packages is done securely and thus preserves the integrity of packages code.

T.INTEG-APPLI-DATA[REFINED]

Objective	Rationale
OT.SID	Counters this threat by providing correct identification of applets.
OT.FIREWALL	Contributes to counter this threat by providing means of separating data.
OT.GLOBAL_ARRAYS_INTEG	Counters this threat by ensuring the integrity of the information stored in the APDU buffer. Application data that is sent to the applet as clear text arrives in the APDU buffer, which is a resource shared by all applications.
OT.ARRAY_VIEWS_INTEG	Counters this threat by preventing the modification of the information shared by applets using array views.
OT.OPERATE	Counters the threat by ensuring that the firewall, which is dynamically enforced, shall never stop operating.

Objective	Rationale
OT.REALLOCATION	Counters the threat by preventing any attempt to read a piece of information that was previously used by an application but has been logically deleted. It states that any information that was formerly stored in a memory block shall be cleared before the block is reused.
OT.ALARM	Contributes to counter this threat by obtaining clear warning and error messages from the firewall, which is a software tool automating critical controls, so that the appropriate countermeasure can be taken.
OT.CIPHER	Contributes to counter this threat by protecting the data shared or information communicated between applets and the CAD using cryptographic functions.
OT.KEY-MNGT	Counters this threat by providing appropriate management of keys, PINs which are particular cases of an application's sensitive data.
OT.PIN-MNGT	Counters this threat by providing appropriate management of keys, PINs which are particular cases of an application's sensitive data.
OT.TRANSACTION	Counters this threat by providing appropriate management of keys, PINs which are particular cases of an application's sensitive data.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OT.SCP.RECOVERY	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OT.SCP.SUPPORT	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OE.CODE-EVIDENCE	Contributes to counter 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.
OT.DOMAIN-RIGHTS	Contributes to counter this threat by ensuring that personalization of the application by its associated security domain is only performed by the authorized AP.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.

T.INTEG-APPLI-DATA.LOAD

Objective	Rationale
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions such as the installation, update or deletion of applets.
OE.CODE-EVIDENCE	Contributes to counter 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.
OT.APPLI-AUTH	Counters this threat by ensuring that the loading of packages is done securely and thus preserves the integrity of packages code.

T.INTEG-JCS-CODE

Objective	Rationale
OT.NATIVE	Counters this threat by ensuring that no native code can be run to modify a piece of code.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode. 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.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OE.CODE-EVIDENCE	Contributes to counter 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

Objective	Rationale
OT.SID	Counters this threat by providing correct identification of applets.
OT.FIREWALL	Contributes to counter this threat by providing means of separation.
OT.OPERATE	Counters the threat by ensuring that the firewall shall never stop operating.
OT.ALARM	Contributes to counter this threat by obtaining clear warning and error messages from the firewall so that the appropriate countermeasure can be taken.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OT.SCP.RECOVERY	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OT.SCP.SUPPORT	Intended to support the OT.OPERATE and OT.ALARM objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OE.CODE-EVIDENCE	Contributes to counter 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.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes.

5.3.1.1.3 Identity Usurpation

T.SID.1

Objective	Rationale
OT.SID	Counters this threat by providing unique subject identification.
OT.FIREWALL	Counters the threat by providing separation of application data (like PINs).

Objective	Rationale
OT.GLOBAL_ ARRAYS_CONFID	Counters this threat by preventing the disclosure of the installation parameters of an applet (like its name). These parameters are loaded into a global array that is also shared by all the applications. The disclosure of those parameters could be used to impersonate the applet.
OT.GLOBAL_ ARRAYS_INTEG	Counters this threat by preventing the disclosure of the installation parameters of an applet (like its name). These parameters are loaded into a global array that is also shared by all the applications. The disclosure of those parameters could be used to impersonate the applet.
OT.CARD-MANAGEMENT	Contributes to counter this threat by preventing usurpation of identity resulting from a malicious installation of an applet on the card.

T.SID.2

Objective	Rationale
OT.SID	Counters this threat by providing unique subject identification.
OT.FIREWALL	Contributes to counter this threat by providing means of separation.
OT.OPERATE	Counters the threat by ensuring that the firewall shall never stop operating.
OT.CARD-MANAGEMENT	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.
OT.SCP.RECOVERY	Intended to support the OT.OPERATE and objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OT.SCP.SUPPORT	Intended to support the OT.OPERATE and objectives of the TOE, thus indirectly related to the threats that these latter objectives contribute to counter.

5.3.1.1.4 Unauthorized Execution

T.EXE-CODE.1

Objective	Rationale
OT.FIREWALL	Counters the threat by preventing the execution of non-shareable methods of a class instance by any subject apart from the class instance owner.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes. 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.

T.EXE-CODE.2

Objective	Rationale
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes. 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. Especially the control flow confinement and the validity of the method references used in the bytecodes are guaranteed.

T.NATIVE

Objective	Rationale
OT.NATIVE	Counters this threat by ensuring that a Java Card applet can only access native methods indirectly that is, through an API.
OE.CAP_FILE	Contributes to counter this threat by ensuring that no native applets shall be loaded in post-issuance.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes. Bytecode verification 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.

5.3.1.1.5 Denial of Service

T.RESOURCES

Objective	Rationale
OT.OPERATE	Counters the threat by ensuring correct working order.
OT.RESOURCES	Counteres the threat directly by objectives on resource-management.
OT.CARD-MANAGEMENT	Counters this threat by controlling the consumption of resources during installation and other card management operations.
OT.SCP.RECOVERY	Intended to support the OT.OPERATE and OT.RESOURCES objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OT.SCP.SUPPORT	Intended to support the OT.OPERATE and OT.RESOURCES objectives of the TOE, thus indirectly related to the threats that these objectives contribute to counter.

5.3.1.1.6 Card Management

T.UNAUTHORIZED_CARD_MNGT

Objective	Rationale
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions such as the loading, installation, extradition or deletion of applets.
OT.DOMAIN-RIGHTS	Contributes to counter this threat by restricting the modification of an AP security domain keyset to the AP who owns it.
OT.COMM_AUTH	Contributes to counter this threat by preventing unauthorized users from initiating a malicious card management operation.
OT.COMM_INTEGRITY	Contributes to counter this threat by protecting the integrity of the card management data while it is in transit to the TOE.

Objective	Rationale
OT.APPLI-AUTH	Counters this threat by ensuring that the loading of a package is safe.

T.COM_EXPLOIT

Objective	Rationale
OT.COMM_AUTH	Contributes to counter this threat by preventing unauthorized users from initiating a malicious card management operation.
OT.COMM_ INTEGRITY	Contributes to counter this threat by protecting the integrity of the card management data while it is in transit to the TOE.
OT.COMM_ CONFIDENTIALITY	Contributes to counter this threat by preventing from disclosing encrypted data transiting to the TOE.

T.LIFE_CYCLE

Objective	Rationale
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions such as the loading, installation, extradition or deletion of applets.
OT.DOMAIN-RIGHTS	Contributes to counter this threat by restricting the use of an AP security domain keysets, and thus the management of the applications related to this SD, to the AP who owns it.

5.3.1.1.7 Services

T.OBJ-DELETION

Objective	Rationale
OT.OBJ-DELETION	Counters this threat by ensuring that object deletion shall not break references to objects.

5.3.1.1.8 Miscellaneous

T.PHYSICAL

Objective	Rationale
OT.SCP.IC	Counters phyiscal attacks. Physical protections rely on the underlying platform and are therefore an environmental issue.
OT.RESTRICTED-MODE	Contributes to counter the threat by ensuring that if the limit of the Attack Counter is reached only limited functionality is available.
OT.SENSITIVE_ RESULTS_INTEG	If the sensitive result is supported by the TOE, this threat is partially covered by the security objective OT.SENSITIVE_RESULTS_INTEG which ensures that sensitive results are protected against unauthorized modification by physical attacks.

5.3.1.1.9 Random Numbers

T.RND

Objective	Rationale
OT.RND	Counters the threat by ensuring the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have sufficient entropy. Furthermore, the TOE ensures that no information about the produced random numbers is available to an attacker.

5.3.1.1.10 Config Applet

T.CONFIG

Objective	Rationale
OT.CARD-CONFIGURATION	Counters the threat by ensuring that the customer can only read and write customer configuration items using the Customer Configuration Token and NXP can read and write configuration items using the NXP Configuration Token. If access is disabled configuration items can not be read or written.

5.3.1.1.11 OS Update

T.UNAUTH-LOAD-UPDATE-IMAGE

Objective	Rationale
OT.SECURE_LOAD_ACODE	Counters the threat directly by ensuring that only authorized (allowed version) images can be installed.
OT.AUTH-LOAD-UPDATE-IMAGE	Counters the threat directly by ensuring that only authorized (allowed version) images can be loaded.

T.INTERRUPT-OSU

Objective	Rationale
OT.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).
OT.TOE_IDENTIFICATION	Counters the threat directly by ensuring that D.TOE_IDENTIFICATION is only updated after successful OS Update procedure.
OT.SECURE_AC_ACTIVATION	Counters the threat directly by ensuring that the SystemOS is only activated after successful (atomic) OS Update procedure.

5.3.1.1.12 Restricted Mode

T.RESTRICTED-MODE

Objective	Rationale
OT.ATTACK-COUNTER	Counters the threat by ensuring that only the ISD can reset the Attack Counter.
OT.RESTRICTED-MODE	Counters the threat by ensuring that only the ISD can reset the Attack Counter.

5.3.1.1.13 Context Separation

T.CONFID-CONT

Objective	Rationale
OT.CONT_SEP	Counters the threat by preventing one context to disclose code or data belonging to another context.

T.INTEG-CONT

Objective	Rationale
OT.CONT_SEP	Counters the threat by preventing one context to alter code or data belonging to another context.

T.CONT-SID

Objective	Rationale
OT.CONT_SEP	Counters the threat by maintaining a context differentiation for each Guest OS.
OT.CONT_PRIV	Counters the threat by preventing execution of code belonging to a Guest OS with kernel privileges.

T.EXE-CONT

Objective	Rationale
OT.CONT_SEP	Counters the threat by preventing one context to execute code belonging to another context.
OT.CONT_PRIV	Counters the threat by preventing execution of code belonging to a Guest OS with kernel privileges.

T.CONT-DOS

Objective	Rationale
OT.CONT_PRIV	Counters the threat by ensuring that the kernel has always the highest privilege.
OT.CONT_DOS	Counters the threat by ensuring that all the contexts will always remain accesible according the configured context management strategy maintained by the SMK.
OT.CONT_SEP	Counters the threat by preventing a context accessing the hardware peripherals and resources of another context without authorization.

5.3.1.2 Rationale for OSPs

OSP.VERIFICATION

Objective	Rationale
OE.VERIFICATION	Enforces the OSP by guaranteeing 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.
OT.CARD-MANAGEMENT	Contributing to enforce the OSP by ensuring that the loading of a CAP file into the card is safe.

Objective	Rationale
OT.APPLI-AUTH	Contributing to enforce the OSP by ensuring that the loading of a CAP file into the card is safe.
OE.CODE-EVIDENCE	This policy is enforced 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.

OSP.PROCESS-TOE

Objective	Rationale
OT.IDENTIFICATION	Enforces this organisational security policy by ensuring that the TOE can be uniquely identified.

OSP.KEY-CHANGE

Objective	Rationale
OE.KEY-CHANGE	Enforces the OSP by ensuring that the initial keys of the security domain are changed before any operation on them are performed.

OSP.SECURITY-DOMAINS

Objective	Rationale
OE.SECURITY-DOMAINS	Enforces the OSP by dynamically create, delete, and block the security domain during usage phase in post-issuance mode.

5.3.1.3 Rationale for Assumptions

A.CAP_FILE

Objective	Rationale
OE.CAP_FILE	Upholds the assumption by ensuring that no CAP file loaded post-issuance shall contain native methods.

A.VERIFICATION

Objective	Rationale
OE.VERIFICATION	Upholds the assumption by guaranteeing 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.
OE.CODE-EVIDENCE	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.

A.USE_DIAG

Objective	Rationale
OE.USE_DIAG	Directly upholds this assumption.

A.USE_KEYS

Objective	Rationale
OE.USE_KEYS	Directly upholds this assumption.

A.PROCESS-SEC-IC

Objective	Rationale
OE.PROCESS_SEC_IC	Directly upholds this assumption.

A.APPS-PROVIDER

Objective	Rationale
OE.APPS-PROVIDER	Directly upholds this assumption.

A.TRUSTED-GUESTOS

Objective	Rationale
OE.TRUSTED-GUESTOS	Directly upholds this assumption.

A.VERIFICATION-AUTHORITY

Objective	Rationale
OE.VERIFICATION-AUTHORITY	Directly upholds this assumption.

5.3.2 Security Objective Rational related to eUICC

The rationales of Security Objectives for the eUICC component of the TOE and for its Environment are strictly the same in the eUICC PP [12].

6 Extended Components Definition (ASE_ECD)

6.1 Extended Components Definition for Java Card System

There is no extended components defined in the JCOP PP[11]. The ST defines an additional extended component FAU_SAS.1 for JCOP part.

6.1.1 Audit Data Storage (FAU_SAS)

This section has been taken over from the certified Smartcard IC Platform Protection Profile [10]. To define the security functional requirements of the TOE an additional family (”Audit Data Storage (FAU_SAS)”) of the Class ”Security audit (FAU)” is defined here. This family describes the functional requirements for the storage of audit data. It has a more general approach than FAU_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

6.1.1.1 Family behaviour

This family defines functional requirements for the storage of audit data.

Component Leveling:

Figure 4. SAS Component

FAU_SAS Requires the TOE to provide the possibility to store audit data.

Management: FAU_SAS.1

There are no management activities foreseen.

Audit: FAU_SAS.1

There are no actions defined to be auditable.

FAU_SAS.1	**Audit storage.**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FAU_SAS.1.1	The TSF shall provide [**assignment: *list of subjects***] with the capability to store [assignment: list of audit information] in the [**assignment: *type of persistent memory***].

6.2 Extended Components Definition for eUICC

There is no extended components defined in the eUICC PP[12]. The ST doesn't define any extended components for eUICC part either.

7 Security Functional Requirements (ASE_REQ)

7.1 Security Functional Requirements for Java Card System

7.1.1 Security Functional Requirements

This section states the security functional requirements for the JCOP component of the TOE. For readability requirements are arranged into groups taken from [11]. Further groups are added to cover additional security functional requirements.

In this chapter, the assignment and selection operations of the SFRs are marked within [ ] with the keywords "assignment" or "selection" printed in bold. There is no distinction between the opertions preformed in the PP and the operations performed in the ST. The iterations are marked by an identifier within [ ] appended to the name of the SFR. The refinments coming from the PP are reproduced as they are in the PP. Some additional refinements are introduced and are explicitly identified in a dedicated "Refinment" paragraph just after the SFR statement. Finally, a refinment for a group of SFRs is provided and justified in Section 2.3.2.3 around Table 21. Note that this convention only applies to the current chapter.

Table 24. Requirement Groups

Group	Description
Core with Logical Channels (CoreG)	The CoreG 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.
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.
Remote Method Invocation (RMIG)	The RMIG contains the security requirements for the remote method invocation feature, which provides a new protocol of communication between the terminal and the applets. This feature is not supported by the TOE.
Object deletion (ODELG)	The ODELG contains the security requirements for the object deletion capability. This provides a safe memory recovering mechanism.
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, in those configurations that do not support on-card static or dynamic bytecode verification, the installation of a CAP file that has not been bytecode verified, or that has been modified after bytecode verification.
External Memory (EMG)	The EMG contains the security requirements for the external memory feature. This feature is not supported by the TOE.
Further Security Functional Requirements	This group contains further security requirements not covered by the PP [11].

Table 24. Requirement Groups*...continued*

Group	Description
Configuration (ConfG)	This group contains security requirements related to NXP Proprietary product configuration feature.
OS Update	This group contains security requirements related to NXP Proprietary product OS Update feature.
Restricted Mode	This group contains security requirements related to NXP Proprietary Restricted Mode feature.
Context Separation (CONTSEP)	The CONTSEP group contains the requirements for context separation between the SMK and hosted Guest OSs, and between each hosted Guest OSs themselves.

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:

Table 25. Java Card Subject Descriptions

Subjects	Descriptions
S.ADEL	The applet deletion manager which also acts on behalf of the card issuer. It may be an applet ([15], §11), but its role asks anyway for a specific treatment from the security viewpoint.
S.CAD	The CAD represents the actor that requests services by issuing commands to the card. It also plays the role of the off-card entity that communicates with the S.INSTALLER.
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.SD	A GlobalPlatform Security Domain representing on the card a off-card entity. This entity can be the Issuer, an Application Provider, the Controlling Authority or the Verification Authority.
S.MEMBER	Any object’s field, static field or array position.
S.PACKAGE	A package is a namespace within the Java programming language that may contain classes and interfaces, and in the context of Java Card technology, it defines either a user library, or one or several applets.

Table 25. Java Card Subject Descriptions*...continued*

Subjects	Descriptions
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.OSU	OSU 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 image is signed and transferred encrypted to the device and is not disclosed during the creation and transfer. The keys used for signing and encrypting the image are kept confidential.
S.Customer	The subject that has the Customer Configuration Token generation key.
S.NXP	The subject that has the NXP Configuration Token generation key.
S.ACAdmin	The subject that has the Attack Counter Token Key.
S.ConfigurationMechanism	On card entity which can read and write configuration items.
S.SMK	The SMK (secured privileged state) that is in the boundaries of the TOE.
S.GuestOS	One or several Guest Operating System (non-secured state) inside or outside the boundaries of the TOE. S.GuestOS includes the specific code of the Guest OS but also the Shared Code that is executed by the GuestOS and executed with the inherited access rights of this GuestOS.

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

Table 26. Object Groups

Objects	Descriptions
O.APPLET	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.

Objects specific to DOMAIN SEPARATION (prefixed with an "O") are described in the following table:

Table 27. Domain Separation Object Groups

Objects	Descriptions
O.GuestOS_Memory_Region	Memory region (addressable memory cells or registers) that is allocated to S.GuestOS (i.e. a context) though the Access Control Table. Some memory regions are in the boundaries of the TOE and some other not.

Table 27. Domain Separation Object Groups *...continued*

Objects	Descriptions
O.SMK_Memory_ Region	Memory region (addressable memory cells or registers) that is allocated to S.SMK and that is in the boundaries of the TOE.

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

Table 28. Information Groups

Information	Description
I.DATA	JCVM Reference Data: objectref addresses of APDU buffer, JCRE-owned instances of APDU class and byte array for install method.

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

Table 29. Security attribute description

Security attributes	Description
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.
Attack Counter	Attack Counter
CAP File AID	The AID of a CAP file.
Context	CAP file AID or "Java Card RE ".
Currently Active Context	CAP file AID or "Java Card RE".
Current Sequence Number	The current number of a valid OS installed on the TOE or current number of a OS update step during update process.
Dependent Package AID	Allows the retrieval of the Package AID and applet's version number.
Final Sequence Number	The sequence number which is reached after completing the update process. This is uniquely linked to the JCOP version of the final TOE.
Image Type	Type of D.UPDATE_IMAGE. Can be either Upgrade, Self Update or Downgrade.
LC Selection Status	Multiselectable, Non-multiselectable or "None".
LifeTime	CLEAR_ON_DESELECT or PERSISTENT. *Note: 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.*
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.
Reference Sequence Number	Is the sequence number which the TOE has before the update process is started. This is uniquely linked to the JCOP version of the initial TOE.

Table 29. Security attribute description*...continued*

Security attributes	Description
Registered Applets	The set of AID of the applet instances registered on the card.
Remote	An object is Remote if it is an instance of a class that directly or indirectly implements the interface java.rmi.Remote. It applies only if the TOE provides JCRMI functionality.
Resifdent CAP files	The set of AIDs of the CAP files already loaded on the card.
Resident Packages	The set of AIDs of the packages already loaded on the card.
Selected Applet Context	CAP file AID or ”None”.
Sharing	Standards, 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 records those references.
Address Space	Accessible memory portion.
Verification Key	Key to verify integrity of D.UPDATE_IMAGE.
Decryption Key	Key for decrypting D.UPDATE_IMAGE.
Customer Configuration Token generation key	The customer key to generate tokens for product configuration.
NXP Configuration Token generation key	The NXP key to generate tokens for product configuration.
Attack Counter Token Key	The key to generate tokens for Attack Counter Reset.
NXP Configuration Access	The NXP Configuration Access can either be enabled or disabled.
Customer Configuration Access	The Customer Configuration Access can either be enabled or disabled.
Access privilege	For each configuration item the access privilege attribute defines who (Customer and/or NXP) is allowed to read/write the item.
Key Set	Key Set for Secure Channel.
Received Sequence Number	Sequence number of the uploaded D.UPDATE_IMAGE.
Security Level	Secure Communication Security Level defined in Section 10.6 of [18].
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.
Card Life Cycle	Defined in Section 5.1.1 of [18].
Privileges	Defined in Section 6.6.1 of [18].
Loaded Applet AID	AID of O.APPLET_LOADED.
Current Instance AID	The AID of O.APPLET_CURRENT that is to be updated.

Table 29. Security attribute description*...continued*

Security attributes	Description
New Instance AID	The AID of O.APPLET_LOADED that is loaded onto the TOE and replaces O.APPLET_CURRENT.
Life-cycle Status	Defined in Section 5.3.2 of [18]
Access Control Table	Security attributes used to define the access control of S.GuestOS and S.SMK to objects O.GuestOS_Memory_Region and O.SMK_Memory_Region.

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.

Table 30. Operation Description

Operations	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)(*)[1]	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.

Table 30. Operation Description*...continued*

Operations	Description
OP.JAVA (...)	Any access in the sense of [15], §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.PUT (S1,S2,I)	Transfer a piece of information I from S1 to S2.
OP.THROW (O.JAVAOBJECT)	Throwing of an object (athrow, see [15], §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.READ_CONFIG_ ITEM	Reading a Config Item from the configuration area.
OP.MODIFY_ CONFIG_ITEM	Writing of a Config Item.
OP.USE_CONFIG_ ITEM	Operational usage of Config Items by subjects inside the TOE.
OP.TRIGGER_ UPDATE	APDU Command that initializes the OS Update procedure.
OP.CONT_ACCESS	Any Read/Write/Execute CPU or DMA (not Execute for DMA) operation performed by S.GuestOS, S.SMK
OP.Modifiction_Of_ Access_Control_Table	Modifiction of the Access Control Table

[1] 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.

7.1.1.1 COREG Security Functional Requirements

The list of SFRs of this category are taken from [11].

7.1.1.1.1 Firewall policy

FDP_ACC.2 [FIREWALL]	**Complete access control (FIREWALL)**
Hierarchical to:	FDP_ACC.1 Subset access control.
Dependencies:	FDP_ACF.1 Security attribute based 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(Sharing, LifeTime)(*), • OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...), • OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...),

- OP.JAVA(...),
- OP.THROW(O.JAVAOBJECT),
- OP.TYPE_ACCESS(O.JAVAOBJECT, class),
- OP.ARRAY_LENGTH(O.JAVAOBJECT, field),
- OP.ARRAY_T_ALOAD(O.JAVAOBJECT, field),
- OP.ARRAY_T_ASTORE(O.JAVAOBJECT, field),
- OP.ARRAY_AASTORE(O.JAVAOBJECT, field).

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_ACF.1 [FIREWALL] **Security attribute based access control (FIREWALL)**

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation.
FDP_ACF.1.1 [FIREWALL]	The TSF shall enforce the **[assignment: FIREWALL access control SFP]** to objects based on the following **[assignment:**

- S.CAP_FILE: security attributes LC Selection Status
- S.JCVM: security attributes Active Applets, Currently Active Context
- S.JCRE: security attributes Selected Applet Context
- O.JAVAOBJECT: security attributes Sharing, Context, LifeTime

].

FDP_ACF.1.2 [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 ([15], §6.2.8): S.CAP_FILE may freely perform
  - OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...)
  - OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...)
  - OP.THROW(O.JAVAOBJECT)
  - OP.TYPE_ACCESS(O.JAVAOBJECT, class)

  upon any O.JAVAOBJECT whose Sharing attribute has value "JCRE entry point" or "global array".
- R.JAVA.2 ([15], §6.2.8): S.CAP_FILE may freely perform
  - OP.ARRAY_ACCESS
  - OP.INSTANCE_FIELD
  - OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...)

– OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...)
– OP.THROW(O.JAVAOBJECT)
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 ([15], §6.2.8.10): S.CAP_FILE may perform
  – OP.TYPE_ACCESS(O.JAVAOBJECT, class)
  upon an O.JAVAOBJECT with Context attribute different from the current 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 ([15], §6.2.8.6): S.CAP_FILE may perform
  – OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...)
  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 ”CAP File AID”, only if the invoked interface method extends the Shareable interface and one of the following conditions applies:
  – The value of the attribute LC Selection Status of the CAP file whose AID is ”CAP File AID” is ”Multiselectable”,
  – The value of the attribute LC Selection Status of the CAP file whose AID is ”CAP File AID” is ”Non-multiselectable”, and either ”Package AID” is the value of the currently selected applet or otherwise ”CAP File AID” does not occur in the attribute Active Applets.
- R.JAVA.5: S.CAP_FILE may perform
  – OP.CREATE(Sharing, LifeTime)(*)
  upon O.JAVAOBJECT only if the value of the Sharing parameter is ”Standard” or ”SIO”.
- R.JAVA.6 ([15], §6.2.8.10): S.CAP_FILE may freely perform
  – OP.ARRAY_ACCESS(O.JAVAOBJECT, field)
  – OP.ARRAY_LENGTH(O.JAVAOBJECT, field)
  upon any O.JAVAOBJECT whose Sharing attribute has value ”global array”.

].

FDP_ACF.1.3 [FIREWALL]

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment:**

- The subject S.JCRE can freely perform OP.JAVA(...) and OP.CREATE(Sharing, LifeTime)(*), with the exception given in FDP_ACF.1.4 [FIREWALL], provided it is the Currently Active Context.
- 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(O.JAVAOBJECT, method, arg1, ...)
  – OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...))

].

FDP_ACF.1.4 [FIREWALL]

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

- 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.
- Any subject attempting to create an object by the means of OP.CREATE(Sharing, LifeTime)(*) and a ”CLEAR_ON_DESELECT” LifeTime parameter if the active context is not the same as the Selected Applet Context.
- S.CAP_FILE performing OP.ARRAY_AASTORE(O.JAVAOBJECT, field) of the reference of an O.JAVAOBJECT whose Sharing attribute has value ”global array” or ”Temporary”.
- 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”.
- R.JAVA.7 ([15], §6.2.8.2): S.CAP_FILE performing OP.ARRAY_T_ASTORE into an array view without ATTR_WRITABLE_VIEW access attribute.
- R.JAVA.8 ([15], §6.2.8.2): S.CAP_FILE performing OP.ARRAY_T_ALOAD into an array view without ATTR_READABLE_VIEW access attribute.

].

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.

In the case of an array type, fields are components of the array ([17], §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 ([15], §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).

([15], 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 (CAP file AID) of the selected applet. In this policy, the ”Selected Applet Context” is the AID of the selected CAP file.

([15], §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 CAP file being selected simultaneously. These applets are referred to as multiselectable applets. Applets that belong to a same CAP file are either all multiselectable or not ([14], §2.2.5). Therefore, the selection mode can be regarded as an attribute of CAP file. No selection mode is defined for a library CAP file.

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. ([15], §4).

FDP_IFC.1 [JCVM]	**Subset information flow control (JCVM)**
Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1 [JCVM]	The TSF shall enforce the **[assignment: JCVM information flow control SFPs]** on **[assignment: 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_IFF.1 [JCVM]	**Simple security attributes (JCVM)**
Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation.
FDP_IFF.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:** • S.JCVM: security attributes Currently Active Context **]**.
FDP_IFF.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 authorise 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 ([15], §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**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 [17], §2.5.1.

FMT_MSA.1 [JCRE]	**Management of security attributes (JCRE)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.
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 [15], §4 and [14], §3.4.

FMT_MSA.1 [JCVM]	**Management of security attributes (JCVM)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.

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 Selected Applet Context should be performed in accordance with the rules given in [15], §4 and [14], §3.4.

FMT_MSA.2 [FIREWALL-JCVM]	**Secure security attributes (FIREWALL-JCVM)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles.
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. • An O.JAVAOBJECT whose Sharing attribute value is a global array necessarily has ”array of primitive type” as a JavaCardClass security attribute’s value. • 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 initialisation (FIREWALL)**
Hierarchical to:	No other components.

Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.
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-EditoriallyRefined]	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 ([15], §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 initialisation (JCVM)**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.
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-EditoriallyRefined]	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	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions:**[assignment:** 1. modify the Currently Active Context, the Selected Applet Context and the Active Applets **]**.

FMT_SMR.1	**Security roles**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1	The TSF shall maintain the roles **[assignment:** • Java Card RE (JCRE), • Java Card VM (JCVM). **]**.
FMT_SMR.1.2	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. The whole set of cryptographic algorithms is generally not implemented because of limited memory resources and/or limitations due to exportation. Therefore, the following requirements only apply to the implemented subset. 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**
Hierarchical to:	No other components.
Dependencies:	FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation] [FCS_RBG.1 Random bit generation, or FCS_RNG.1

Generation of random numbers] FCS_CKM.6 Timing and event of cryptographic key destruction.

FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment:JCOP RNG]** and specified cryptographic key sizes **[assignment: DES: Key lengths - LENGTH_DES3_2KEY, LENGTH_DES3_3KEY bit, AES: Key lengths - LENGTH_AES_128, LENGTH_AES_192, LENGTH_AES_256 bit** that meet the following: **[assignment: FCS_RNG.1 or FCS_RNG.1[HDT]].**
FCS_CKM.1.1[RSA]	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: RSA and RSA-CRT key generation algorithm]** and specified cryptographic key sizes **[assignment: from 512 to 4096 bits by steps of 256 bits]** that meet the following: **[assignment: [37] and [41]]**.
FCS_CKM.1.1[ECC]	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: ECC key generation algorithm]** and specified cryptographic key sizes **[assignment: any length from 128 to 528 bits]** that meet the following: **[assignment: [38] and [41]]**.
FCS_CKM.1.1[EdDSA]	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: EdDSA key generation algorithm]** and specified cryptographic key sizes **[assignment: any length from 128 to 528 bits]** that meet the following: **[assignment: [35]]**.
FCS_CKM.1.1[Mont]	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: MontDH Key generation algorithm]** and specified cryptographic key sizes **[assignment: any length from 128 to 528 bits]** that meet the following: **[assignment: [36]]**.
Application Note	• The keys can be generated and diversified in accordance with [13] specification in class KeyBuilder. • This component shall be instantiated according to the version of the Java Card API applying to the security target and the implemented algorithms ([13]).
Application Note	• The keys can be generated and diversified in accordance with [45], [51], [57], [63], [69] specifications in class KeyBuilderX. • This component shall be instantiated according to the version of the Java Card API applying to the security target and the implemented algorithms ([45], [51], [57], [63], [69]).

FCS_CKM.6 **Timing and event of cryptographic key destruction**

Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation].
FCS_CKM.6.1	The TSF shall destroy **[assignment: all cryptographic keys]** when **[assignment: 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: physically overwriting the keys in a randomized manner]** that meets the following: **[assignment: none]**.
Application Note	• The keys are reset as specified in [13] Key class, with the method clearKey(). Any access to a cleared key for ciphering or signing shall throw an exception. • This component shall be instantiated according to the version of the Java Card API applicable to the security target and the implemented algorithms ([13]).

FCS_COP.1	**Cryptographic Operation**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation], FCS_CKM.6 Timing and event of cryptographic key destruction.
FCS_COP.1.1 [GCM]	The TSF shall perform **[assignment: encryption and decryption]** in accordance with a specified cryptographic algorithm **[assignment: AES in GCM mode]** and cryptographic key sizes **[assignment: 128 bits, 192 and 256 bits]** that meet the following: **[assignment: FIPS 197 [42], NIST Special Publication 800-38D Recommendation for BlockCipher [40]]**.
FCS_COP.1.1 [CCM]	The TSF shall perform **[assignment: encryption and decryption]** in accordance with a specified cryptographic algorithm **[assignment: AES in CCM mode]** and cryptographic key sizes **[assignment: 128 bits, 192 and 256 bits]** that meet the following: **[assignment: FIPS 197 [42], NIST Special Publication 800-38C Recommendation for BlockCipher [39]]**.
FCS_COP.1.1 [TripleDES]	The TSF shall perform **[assignment: encryption and decryption]** in accordance with a specified cryptographic algorithm in **[assignment:**

- ALG_DES_CBC_ISO9797_M1
- ALG_DES_CBC_ISO9797_M2
- ALG_DES_CBC_NOPAD
- ALG_DES_ECB_ISO9797_M1
- ALG_DES_ECB_ISO9797_M2
- ALG_DES_ECB_NOPAD
- ALG_DES_CBC_PKCS5
- ALG_DES_ECB_PKCS5
- ALG_DES_CBC_PKCS7
- ALG_DES_ECB_PKCS7

] and cryptographic key sizes **[assignment: LENGTH_DES3_2KEY, LENGTH_DES3_3KEY]** that meet the following: **[assignment: for ALG_DES_ECB_ISO9797_M2 see Java Card API Spec [13], for the rest see both [13] and JCOPX API [45], [51], [57], [69]]**.

FCS_COP.1.1 [AES] The TSF shall perform **[assignment: encryption and decryption]** in accordance with a specified cryptographic algorithm **[assignment:**

- ALG_AES_BLOCK_128_CBC_NOPAD
- ALG_AES_BLOCK_128_CBC_NOPAD_STANDARD
- ALG_AES_BLOCK_128_ECB_NOPAD
- ALG_AES_CBC_ISO9797_M1
- ALG_AES_CBC_ISO9797_M2
- ALG_AES_CBC_ISO9797_M2_STANDARD
- ALG_AES_ECB_ISO9797_M1
- ALG_AES_ECB_ISO9797_M2
- ALG_AES_CBC_PKCS5
- ALG_AES_ECB_PKCS5
- ALG_AES_CBC_PKCS7
- ALG_AES_ECB_PKCS7
- AES CTR
- AES CFB

] and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_AES_256 bit]** that meet the following: **[assignment: for ALG_AES_BLOCK_128_CBC_NOPAD_STANDARD, ALG_AES_CBC_ISO9797_STANDARD, ALG_AES_CBC_PKCS7, ALG_AES_ECB_PKCS7, ALG_AES_CFB see API specified in JCOPX [45], [51], [57], [63], [69], for the rest see Java Card API Spec [13]]**.

FCS_COP.1.1 [RSACipher] The TSF shall perform **[assignment: encryption and decryption]** in accordance with a specified cryptographic algorithm **[assignment: ALG_RSA_NOPAD, ALG_RSA_PKCS1, ALG_RSA_PKCS1_OAEP]** and cryptographic key sizes **[assignment: any key length that is amultiple of 32 between 512 and 4096 bits]** that meet the following: **[assignment: Java Card API Spec [13] and for the**

32 bit step range see API specified in JCOPX [45], [51], [57], [63], [69]].

FCS_COP.1.1 [ECDH_P1363]	The TSF shall perform **[assignment: Diffie-Hellman Key Agreement]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_EC_SVDP_DH • ALG_EC_SVDP_DH_KDF • ALG_EC_SVDP_DH_PLAIN • ALG_EC_SVDP_DHC • ALG_EC_SVDP_DHC_KDF • ALG_EC_SVDP_DHC_PLAIN • ALG_EC_SVDP_DH_PLAIN_XY **]** and cryptographic key sizes **[assignment: LENGTH_EC_FP_128, LENGTH_EC_FP_160, LENGTH_EC_FP_192, LENGTH_EC_FP_224, LENGTH_EC_FP_256,LENGTH_EC_FP_384, LENGTH_EC_FP_528 and from 128 bit to 528 bit in 1 bit steps]** that meet the following: **[assignment: Java Card API Spec [13] and for ALG_EC_SVDP_DH_PLAIN_XY 1 bit step range key size see API specified in JCOPX [45], [51], [57], [63], [69]]**.
FCS_COP.1.1 [ECDH_25519]	The TSF shall perform **[assignment: Diffie-Hellman Key Agreement]** in accordance with a specified cryptographic algorithm **[assignment: ALG_XDH]** and cryptographic key sizes **[assignment: 255 bits]** that meet the following: **[assignment: Java Card API Spec [13]]**.
FCS_COP.1.1 [DESMAC]	The TSF shall perform **[assignment: MAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: Triple-DES in outer CBC for Mode:** • 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_MAC8_NOPAD • ALG_DES_MAC4_PKCS5 • ALG_DES_MAC8_PKCS5 **]** and cryptographic key sizes **[assignment: LENGTH_DES3_2KEY, LENGTH_DES3_3KEY]** that meet the following: **[assignment: Java Card API Spec [13] and JCOPX API [45], [51], [57], [63], [69]]**.

FCS_COP.1.1 [AESMAC]	The TSF shall perform **[assignment: 16 byte MAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: AES in CBC Mode ALG_AES_MAC_128_NOPAD]** and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_AES_256 bit]** that meet the following: **[assignment: Java Card API Spec [13]]**.
FCS_COP.1.1 [RSASignaturePKCS1]	The TSF shall perform **[assignment: digital signature generation and verification]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_RSA_SHA_224_PKCS1 • ALG_RSA_SHA_224_PKCS1_PSS • ALG_RSA_SHA_256_PKCS1 • ALG_RSA_SHA_256_PKCS1_PSS • ALG_RSA_SHA_384_PKCS1 • ALG_RSA_SHA_384_PKCS1_PSS • ALG_RSA_SHA_512_PKCS1 • ALG_RSA_SHA_512_PKCS1_PSS • ALG_RSA_SHA_ISO9796 • ALG_RSA_SHA_ISO9796_MR • SIG_CIPHER_RSA in combination with MessageDigest.ALG_SHA_256 or MessageDigest.ALG_SHA_384 or MessageDigest.ALG_SHA_512 and in combination with Cipher.PAD_PKCS1_OAEP **]** and cryptographic key sizes **[assignment: any key length that is a multiple of 32 between 512 and 4096 bits]** that meet the following: **[assignment: Java Card API Spec [13] and for the 32 bit step range see API specified in JCOPX [45], [51], [57], [63], [69]]**.
FCS_COP.1.1 [ECSignature]	The TSF shall perform **[assignment: digital signature generation and verification]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_ECDSA_SHA_224 • ALG_ECDSA_SHA_256 • ALG_ECDSA_SHA_384 • ALG_ECDSA_SHA_512 • SIG_CIPHER_ECDSA in combination with MessageDigest.ALG_SHA_256 or MessageDigest.ALG_SHA_384 or MessageDigest.ALG_SHA_512] **]** and cryptographic key sizes **[assignment: LENGTH_EC_FP_128,LENGTH_EC_FP_160, LENGTH_EC_FP_192, LENGTH_EC_FP_224, LENGTH_EC_FP_256, LENGTH_EC_FP_384, LENGTH_EC_FP_528 and from 128 bit to 528 bit in 1 bit**

	steps] that meet the following: **[assignment: Java Card API Spec [13] and for 1 bit step range key size see API specified in JCOPX [45], [51], [57], [63], [69]]**.
FCS_COP.1.1 [EdDSA]	The TSF shall perform **[assignment: digital signature generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: ALG_ED25519PH_SHA_512]** and cryptographic key sizes **[assignment: 256 bit for private key, 256 bit for public key]** that meet the following: **[assignment: API specified in JCOPX [45], [51], [57], [63], [69]]**.
FCS_COP.1.1 [SHA]	The TSF shall perform **[assignment: secure hash computation]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_SHA [1] • ALG_SHA_224 • ALG_SHA_256 • ALG_SHA_384 • ALG_SHA_512 **]** and cryptographic key sizes **[assignment: LENGTH_SHA, LENGTH_SHA_224, LENGTH_SHA_256, LENGTH_SHA_384, LENGTH_SHA_512** that meet the following: **[assignment: Java Card API Spec [13] and JCOPX API specified in [45], [51], [57], [63], [69]]**.
FCS_COP.1.1 [AES_CMAC]	The TSF shall perform **[assignment: CMAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_AES_CMAC8 • ALG_AES_CMAC16 • SIG_CIPHER_AES_CMAC8 • SIG_CIPHER_AES_CMAC16 • SIG_CIPHER_AES_CMAC128 • ALG_AES_CMAC16_STANDARD • ALG_AES_CMAC_128 **]** and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_AES_256 bit]** that meet the following: **[assignment: see Java Card API Spec [13] and the JCOPX API specified in [45], [51], [57], [63], [69]]**.
FCS_COP.1.1 [HMAC]	The TSF shall perform **[assignment: HMAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_HMAC_SHA_256 • ALG_HMAC_SHA_384

1 Due to mathematical weakness only resistant against AVA_VAN.5 for temporary data (e.g. as used for generating session keys), but not if repeatedly applied to the same input data.

- ALG_HMAC_SHA_512

] and cryptographic key sizes **[assignment: LENGTH_SHA_256,LENGTH_SHA_384 and LENGTH_SHA_512 bit]** that meet the following: **[assignment: Java Card specification [13] and JCOPX API [45], [51], [57], [63], [69]]**.

FCS_COP.1.1 [TDES_CMAC]	The TSF shall perform **[assignment: message authentication and verification]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_DES_CMAC8 • SIG_CIPHER_DES_CMAC8 **]** and cryptographic key sizes **[assignment: LENGTH_DES3_2KEY and LENGTH_DES3_3KEY bit]** that meet the following: **[assignment: see API specified in JCOPX [45], [51], [57], [63], [69]]**.
FCS_COP.1.1 [DAP]	The TSF shall perform **[assignment: verification of the DAP signature attached to Executable Load Applications]** in accordance with a specified cryptographic algorithm **[assignment:** • ALG_RSA_SHA_PKCS1 • ALG_ECDSA_SHA_256 **]** and cryptographic key sizes **[assignment: LENGTH_RSA_1024, LENGTH_EC_FP_256]** that meet the following: **[assignment: GP Spec [22] and JCOPX API [45], [51], [57], [63], [69]]**.

FCS_RNG.1	**Random Number Generation.**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	The TSF shall provide a **[selection: deterministic]** random number generator **[DRG.3][AIS20]**that implements: **[assignment:** • (DRG.3.1) If initialized with a random seed using a PTRNG of class PTG.2 (as defined in [7]) as random source, the internal state of the RNG shall have at least 256 bit of entropy. • (DRG.3.2) The RNG provides forward secrecy (as defined in [7]). • (DRG.3.3) The RNG provides enhanced backward secrecy even if the current internal state is known (as defined in [7]) **]**.
FCS_RNG.1.2	The TSF shall provide **[selection: bits]** that meet **[assignment:**

- (DRG.3.4) The RNG, initialized with a random seed using a PTRNG of class PTG.2 (as defined in [7]) as random source, generates output for which 2^{48} strings of bit length 128 are mutually different with probability at least $1-2^{24}$
- (DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [7]).

].

Application Note
- This functionality is provided by the Crypto Library.
- FCS_RNG.1 and FCS_RNG.1[HDT] both apply to the same Random Number Generator.

FCS_RNG.1 [HDT] **Random Number Generation.**

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 [HDT] The TSF shall provide a **[selection: hybrid deterministic]** random number generator that implements: **[assignment:**

- (DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 (as defined in [7]) as random source
- (DRG.4.2) The RNG provides forward secrecy (as defined in [7]).
- (DRG.4.3) The RNG provides backward secrecy even if the current internal state is known (as defined in [7])
- (DRG.4.4) The RNG provides enhanced forward secrecy on demand (as defined in [7])
- (DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2 (as defined in [7])

].

FCS_RNG.1.2 [HDT] The TSF shall provide **[selection: bits]** that meet **[assignment:**

- (DRG.4.6) The RNG generates output for which 2^{48} strings of bit length 128 are mutually different with probability at least $1-2^{24}$
- (DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [7]).

].

Application Note
- This functionality is provided by the Crypto Library.
- FCS_RNG.1 and FCS_RNG.1[HDT] both apply to the same Random Number Generator. The 'enhanced forward secrecy'

giving access to DRG.4 is an option that can only be activated by NXP on specific customer request.
- For DRG.4.5: The Hardware PTRNG of class PTG.2 generates random data used as an input for the derivation function. The result of the derivation function is used as the seed.

FDP_RIP.1 [ABORT]	**Subset residual information protection (ABORT)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1 [ABORT]	The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: de-allocation 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 [15], §5.1.

FDP_RIP.1 [APDU]	**Subset residual information protection (APDU)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1 [APDU]	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: 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 (GlobalArray)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1 [GlobalArray-*Refined*]	The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: deallocation of the resource from]** *the applet as a result of returning from*

	the process method the following objects: **[assignment: a user Global Array]**
Application Note	An array resource is allocated when a call to the API method JCSystem.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 (bArray)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 (KEYS)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 and javacardx.crypto packages do provide secure interfaces to the cryptographic buffer in a transparent way. See javacard.security.KeyBuilder and Key interface of [13].

FDP_RIP.1 [TRANSIENT]	**Subset residual information protection (TRANSIENT)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1 [TRANSIENT]	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 transient object]**.
Application Note	• The events that provoke the de-allocation of any transient object are described in [15], §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 CAP file must share the transient memory segment if they are concurrently active ([15], §4.2.).

FDP_ROL.1 [FIREWALL]	**Basic rollback (FIREWALL)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
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(Sharing, LifeTime)(*)]** 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 [15], §7.7, within the bounds of the Commit Capacity ([15], §7.8), and those described in [13]]**.
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 [13] (see for instance, PIN-blocking, PIN-checking, update of Transient objects).

7.1.1.1.3 Card Security Management

FAU_ARP.1	**Security alarms**
Hierarchical to:	No other components.
Dependencies:	FAU_SAA.1 Potential violation analysis.
FAU_ARP.1.1	The TSF shall take **[assignment: one of the following actions:** • throw an exception, • lock the card session (after a predefined number of resetted sessions the card might switch to Restricted Mode), • reinitialize the Java Card System and its data, • response with error code to S.CAD **]** upon detection of a potential security violation.

FDP_SDI.2 [DATA]	**Stored data integrity monitoring and action (Data)**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring.
Dependencies:	No dependencies.
FDP_SDI.2.1 [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 protected data]**.
FDP_SDI.2.2 [DATA]	Upon detection of a data integrity error, the TSF shall **[assignment: reset the card session for integrity errors]**.
Refinement	The following data elements have the user data attribute ”integrity protected data”: • D.APP_KEYs • D.PIN • D.TOE_IDENTIFIER
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 is implemented and made available to developers to achieve better security for specific objects, following the same pattern that already exists in cryptographic APIs.

FPR_UNO.1	**Unobservability**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPR_UNO.1.1	The TSF shall ensure that **[assignment: all users]** are unable to observe the operation **[assignment: all operations]** on **[assignment: D.APP_KEYs, D.PIN]** by **[assignment: another user]**.

FPT_FLS.1	**Failure with preservation of secure state**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: **[assignment: 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 ([15], §6.2.3) or after a proximity card (PICC) activation sequence ([15]). Behavior of the TOE on power loss and reset is described in [15], §3.6 and §7.1. Behavior of the TOE on RF signal loss is described in [15], §3.6.1.
Refinement:	The term "failure" above also covers "circumstances" for assignments taken from [10]. The TOE prevents failures for the "circumstances" defined above.

FPT_TDC.1	**Inter-TSF basic TSF data consistency**

Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 [14] specification • the API tokens defined in the export files of reference implementation **]** 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 (AID)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_ATD.1.1 [AID]	The TSF shall maintain the following list of security attributes belonging to individual users: **[assignment:** • CAP File AID, • Package AID, • Applet's Version Number, • Registered Applet AID, • Applet Selection Status ([15], §4.6) **]**.
Refinement	"Individual users" stands for applets.

FIA_UID.2 [AID]	**User identification before any action (AID)**
Hierarchical to:	FIA_UID.1 Timing of identification.
Dependencies:	No dependencies.

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 CAP files (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 CAP file 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 is attached to an IT security function rather than to a "use" 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 (AID)

Hierarchical to:	No other components.
Dependencies:	FIA_ATD.1 User attribute definition.
FIA_USB.1.1 [AID]	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **[assignment: CAP file 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: Each uploaded package is associated with an unique Package AID]**.
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: The initially assigned Package AID is unchangeable]**.
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 "CAP file AID".

FMT_MTD.1 [JCRE] Management of TSF data (JCRE)

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.

FMT_MTD.1.1 [JCRE]	The TSF shall restrict the ability to **[selection: modify]** the **[assignment: list of registered applets' AIDs]** to **[assignment: S.JCRE]**.
Application Note	• The installer and the Java Card RE manage other TSF data such as the applet life cycle or CAP files, 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 (JCRE)**
Hierarchical to:	No other components.
Dependencies:	FMT_MTD.1 Management of TSF data.
FMT_MTD.3.1 [JCRE]	The TSF shall ensure that only secure values are accepted for **[assignment: the registered applet AIDs]**.

7.1.1.2 INSTG Security Functional Requirements

The list of SFRs of this category are taken from [11]. The SFR FDP_ITC.2[INSTALLER] has been refined and is now part of the card management SFRs (FDP_ITC.2[CCM]) in Section 7.1.1.6).

FMT_SMR.1 [INSTALLER]	**Security roles (INSTALLER)**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
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 (INSTALLER)**
Hierarchical to:	No other components.

Dependencies:	No dependencies.
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 [15], §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 (INSTALLER)**
Hierarchical to:	FPT_RCV.2 Automated recovery.
Dependencies:	AGD_OPE.1 Operational user guidance.
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: a failure 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 [2], p298: In this maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only authorised 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 [15], §11.1.5 for possible scenarios. Precise behavior is left to implementers. This component shall include among the listed failures the deletion of a CAP file/applet. See ([15], §11.3.4) for possible scenarios. Precise behavior is left to implementers.
- Other events such as the unexpected tearing of the card, power loss, and so on, are partially handled by the underlying hardware platform (see [10]) and, from the TOE’s side, by events ”that clear transient objects” and transactional features. See FPT_FLS.1.1, 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 Security Functional Requirements

The list of SFRs of this category are taken from [11].

FDP_ACC.2 [ADEL]	**Complete access control (ADEL)**
Hierarchical to:	FDP_ACC.1 Subset access control.
Dependencies:	FDP_ACF.1 Security attribute based access control.
FDP_ACC.2.1 [ADEL]	The TSF shall enforce the **[assignment: ADEL access control SFP]** on **[assignment: 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.
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.
Refinement	The operations involved in the policy are: • OP.DELETE_APPLET, • OP.DELETE_CAP_FILE,

- OP.DELETE_CAP_FILE_APPLET.

FDP_ACF.1 [ADEL]	**Security attribute based access control (ADEL)**
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation.
FDP_ACF.1.1 [ADEL]	The TSF shall enforce the **[assignment: ADEL access control SFP]** to objects based on the following **[assignment:** • S.JCVM: security attributes Active Applets • S.JCRE: security attributes Selected Applet Context, Registered Applets, Resident CAP files • O.CODE_CAP_FILE: security attributes CAP file AID, AIDs of packages within a CAP file, Dependent Package AID, Static References • O.APPLET: security attributes Applet Selection Status • O.JAVAOBJECT: security attributes 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 if one of the following conditions hold: 1. the owner of O is a registered applet instance A (O is reachable from A), 2. a static field of a resident package P contains a reference to O (O is reachable from P), 3. there exists a valid remote reference to O (O is remote reachable), 4. 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 ([15], §11.3.4.2, 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 ([15], §8.5) O.JAVAOBJECT is remote reachable.

- R.JAVA.15 ([15], §11.3.4.2.1, 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 CAP file P, or ([15], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.16 ([15], §11.3.4.3, 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 ([15], §11.3.4.4, 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 package 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 ([15], §8.5) O.JAVAOBJECT is remote reachable.

].

FDP_ACF.1.3 [ADEL]	The TSF shall explicitly authorise 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 protection profile.

FDP_RIP.1 [ADEL]	**Subset residual information protection (ADEL)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1 [ADEL]	The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: de-allocation 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 [15], §11.3.4.1, §11.3.4.2 and §11.3.4.3.

FMT_MSA.1 [ADEL]	**Management of security attributes (ADEL)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.
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: S.JCRE]**.

FMT_MSA.3 [ADEL]	**Static attribute initialisation (ADEL)**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.

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 (ADEL)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 (ADEL)

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
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 (ADEL)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 [15], §11.3.4]**.
Application Note	• The TOE may provide additional feedback information to the card manager in case of 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 ([15], §11.3.4.).

7.1.1.4 RMIG Security Functional Requirements

Not used in this ST because RMI is optional in PP [11] and the TOE does not support RMI.

7.1.1.5 ODELG Security Functional Requirements

The list of SFRs of this category are taken from [11].

FDP_RIP.1 [ODEL]	**Subset residual information protection (ODEL)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_RIP.1.1 [ODEL]	The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: de-allocation 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 [13]. • 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 (ODEL)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 a potential security violation (see FAU_ARP.1). • The Package/applet instance deletion must be atomic. The ”secure state” referred to in the requirement must comply with Java Card specification ([15], §11.3.4.).

7.1.1.6 CarG Security Functional Requirements

The card management SFRs from the PP [11] are refined and replaced by the following SFRs.

FDP_UIT.1 [CCM]	**Data exchange integrity (CCM)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path].
FDP_UIT.1.1 [CCM]	The TSF shall enforce the **[assignment: Secure Channel Protocol information flow control policy and the Security Domain access control policy]** to **[selection:receive]** user data in a manner protected from **[selection:modification, deletion, insertion and replay]** errors.
FDP_UIT.1.2 [CCM]	The TSF shall be able to determine on receipt of user data, whether **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.

FDP_ROL.1 [CCM]	**Basic rollback (CCM)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control].
FDP_ROL.1.1 [CCM]	The TSF shall enforce **[assignment: Security Domain access control policy]** to permit the rollback of the **[assignment: installation operation]** on the **[assignment: executable files and application instances]**.
FDP_ROL.1.2 [CCM]	The TSF shall permit operations to be rolled back within the **[assignment: boundaries of available memory before the card content management function started]**.

FDP_ITC.2 [CCM]	**Import of user data with security attributes (CCM)**
Hierarchical to:	No other components.

Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency.
FDP_ITC.2.1 [CCM]	The TSF shall enforce the **[assignment: Security Domain access control policy and the Secure Channel Protocol information flow policy]** when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2 [CCM]	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3 [CCM]	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 [CCM]	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 [CCM]	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 ([14], §4.5.2).]**.
Application Note	This SFR also covers security functionality required by Amendment A of the GP specification [19], i.e. personalizing SDs and loading ciphered load files.

FPT_FLS.1 [CCM]	**Failure with preservation of secure state (CCM)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1 [CCM]	The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the Security Domain fails to load/install an Executable File/application instance as described in [15], Section 11.1.5]**.

FDP_ACC.1 [SD]	**Subset access control (SD)**

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1 [SD]	The TSF shall enforce the **[assignment: Security Domain access control policy]** on **[assignment:** • Subjects: S.INSTALLER, S.ADEL, S.CAD (from [11]) and S.SD • Objects: Delegation Token, DAP Block and Load File • Operations: GlobalPlatform's card content management APDU commands and API methods **]**.

FDP_ACF.1 [SD]	**Security attribute based access control (SD)**
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1 [SD]	The TSF shall enforce the **[assignment: Security Domain access control policy]** to objects based on the following **[assignment:** • Subjects: – S.INSTALLER, defined in [11] and represented by the GlobalPlatform Environment (OPEN) on the card, the Card Life Cycle attributes (defined in Section 5.1.1 of [18]) – S.ADEL, also defined in [11] and represented by the GlobalPlatform Environment (OPEN) on the card – S.SD receiving the Card Content Management commands (through APDUs or APIs) with a set of Privileges (defined in Section 6.6.1 of [18]), a Life-cycle Status (defined in Section 5.3.2 of [18]) and a Secure Communication Security Level (defined in Section 10.6 of [18]) – S.CAD, defined in [11], the off-card entity that communicates with the S.INSTALLER and S.ADEL through S.SD • Objects: – The Delegation Token, in case of Delegated Management operations, with the attributes Present or Not Present – The DAP Block, in case of application loading, with the attributes Present or Not Present – The Load File or Executable File, in case of application loading, installation, extradition or registry update, with a set of intended privileges and its targeted associated SD AID. • Mapping subjects/objects to security attributes: – S.INSTALLER: Security Level, Card Life Cycle, Life-cycle Status, Privileges, Resident Packages, Registered Applets

– S.ADEL: Active Applets, Static References, Card Life Cycle, Life-cycle Status, Privileges, Applet Selection Status, Security Level
– S.SD: Privileges, Life-cycle Status, Security Level
– S.CAD: Security Level

].

FDP_ACF.1.2 [SD] The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[assignment: Runtime behavior rules defined by GlobalPlatform for:**

- loading (Section 9.3.5 of [18])
- installation (Section 9.3.6 of [18])
- extradition (Section 9.4.1 of [18])
- registry update (Section 9.4.2 of [18])
- content removal (Section 9.5 of [18])

].

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

FDP_ACF.1.4 [SD] The TSF shall explicitly deny access of subjects to objects based on the following additional rules:**[assignment: when at least one of the rules defined by GlobalPlatform does not hold]**.

FMT_MSA.1 [SD] **Management of security attributes (SD)**

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.

FMT_MSA.1.1 [SD] The TSF shall enforce the **[assignment: Security Domain access control policy]** to restrict the ability to **[selection: modify]** the security attributes **[assignment:**

- Card Life Cycle,
- Privileges,
- Life-cycle Status,
- Security Level.

] to **[assignment: the Security Domain and the application instance itself]**.

FMT_MSA.3 [SD] **Static attribute initialisation (SD)**

Hierarchical to: No other components.

Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.
FMT_MSA.3.1 [SD]	The TSF shall enforce the **[assignment: Security Domain access control policy]** to provide **[restrictive]** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 [SD]	The TSF shall allow the **[assignment: Card Issuer or the Application Provider]** to specify alternative initial values to override the default values when an object or information is created.
Refinement	Alternative initial values shall be at least as restrictive as the default values defined in FMT_MSA.3.1[SD].

FMT_SMF.1 [SD] Specification of Management Functions (SD)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1 [SD]	The TSF shall be capable of performing the following management functions: **[assignment:** • Management functions specified in GlobalPlatform specifications [GP]: – card locking (Section 9.6.3 of [18]) – application locking and unlocking (Section 9.6.2 of [18]) – card termination (Section 9.6.4 of [18]) – card status interrogation (Section 9.6.6 of [18]) – application status interrogation (Section 9.6.5 of [18]) **]**.

FMT_SMR.1 [SD] Security roles (SD)

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1 [SD]	The TSF shall maintain the roles **[assignment: ISD, SSD]**.
FMT_SMR.1.2 [SD]	The TSF shall be able to associate users with roles.

FCO_NRO.2 [SC] Enforced proof of origin (SC)

Hierarchical to:	FCO_NRO.1 Selective proof of origin.

Dependencies:	FIA_UID.1 Timing of identification.
FCO_NRO.2.1 [SC]	The TSF shall enforce the generation of evidence of origin for transmitted **[assignment: Executable load files]** at all times.
FCO_NRO.2.2 [SC]	The TSF shall be able to relate the **[assignment: DAP Block]** of the originator of the information, and the **[assignment: identity]** of the information to which the evidence applies.
FCO_NRO.2.3 [SC]	The TSF shall provide a capability to verify the evidence of origin of information to **[selection: originator]** given **[assignment: at the time the Executable load files are received as no evidence is kept on the card for future verification]**.
Application Note	FCO_NRO.2.1[SC]: • Upon reception of a new application CAP file for installation, the card manager shall first check that it actually comes from the verification authority. The verification authority is the entity responsible for bytecode verification. FCO_NRO.2.3[SC]: • 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 verifications.

FDP_IFC.2 [SC]	**Complete information flow control (SC)**
Hierarchical to:	FDP_IFC.1 Subset information flow control.
Dependencies:	FDP_IFF.1 Simple security attributes.
FDP_IFC.2.1 [SC]	The TSF shall enforce the **[assignment: Secure Channel Protocol information flow control policy]** on **[assignment:** • the subjects S.CAD and S.SD, involved in the exchange of messages between the TOE and the CAD through a potentially unsafe communication channel, • the information controlled by this policy are the card content management commands, including personalization commands, in the APDUs sent to the card and their associated responses returned to the CAD • **[assignment: none]** **]** and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 [SC]	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 [SC] Simple security attributes (SC)

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation.

FDP_IFF.1.1 [SC] The TSF shall enforce the **[assignment: Secure Channel Protocol information flow control policy]** based on the following types of subject and information security attributes **[assignment: :**

- Subjects:
  - S.SD receiving the Card Content Management commands (through APDUs or APIs).
  - S.CAD the off-card entity that communicates with the S.SD.
- Information:
  - executable load file, in case of application loading;
  - applications or SD privileges, in case of application installation or registry update;
  - personalization keys and/or certificates, in case of application or SD personalization.
- **[assignment: none]**

].

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

- Runtime behavior rules defined by GlobalPlatform for:
  - loading (Section 9.3.5 of [18]);
  - installation (Section 9.3.6 of [18]);
  - extradition (Section 9.4.1 of [18]);
  - registry update (Section 9.4.2 of [18]);
  - content removal (Section 9.5 of [18])

].

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

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

FDP_IFF.1.5 [SC] The TSF shall explicitly deny an information flow based on the following rules: **[assignment:]**.

- When none of the conditions listed in the element FDP_IFF.1.4 of this component hold and at least one of those listed in the element FDP_IFF.1.2 does not hold

].

Application note — The subject S.SD can be the ISD or APSD.

Application note — The on-card and the off-card subjects have security attributes such as MAC, Cryptogram, Challenge, Key Set, Static Keys, etc.

FMT_MSA.1 [SC] **Management of security attributes (SC)**

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.

FMT_MSA.1.1 [SC] The TSF shall enforce the **[assignment: Secure Channel Protocol information flow control policy]** to restrict the ability to **[selection: modify]** the security attributes **[assignment:**

- Key Set,
- Security Level,
- Secure Channel Protocol,
- Session Keys,
- Sequence Counter,
- ICV.

] to **[assignment: the actor associated with the according security domain:**

- The Card Issuer for ISD,
- The Application Provider for APSD

].

Application note — The key data used for setting up a secure channel is according to GP spec [18], Amendment D [21] and Amendmend F [23].

FMT_MSA.3 [SC] **Static attribute initialisation (SC)**

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.

FMT_MSA.3.1 [SC] The TSF shall enforce the **[assignment: Secure Channel Protocol information flow control policy]** to provide

	[restrictive] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 [SC]	The TSF shall allow the **[assignment: Card Issuer, Application Provider]** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1 [SC] Specification of Management Functions (SC)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1 [SC]	The TSF shall be capable of performing the following management functions: **[assignment:** • Management functions specified in GlobalPlatform specifications [GP]: – loading (Section 9.3.5 of [18]) – installation (Section 9.3.6 of [18]) – extradition (Section 9.4.1 of [18]) – registry update (Section 9.4.2 of [18]) – content removal (Section 9.5 of [18]) **]**.
Application note	All management functions related to secure channel protocols shall be relevant.

FIA_UID.1 [SC] Timing of Identification (SC)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1 [SC]	The TSF shall allow **[assignment:** • application selection • initializing a secure channel with the card • requesting data that identifies the card or the Card Issuer **]** on behalf of the user to be performed before the user is identified.
FIA_UID.1.2 [SC]	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 GlobalPlatform TSF mediated actions listed in [GP] such as selecting an application, requesting data, initializing, etc.

FIA_UAU.1 [SC]	**Timing of authentication (SC)**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.1.1 [SC]	The TSF shall allow **[assignment: the TSF mediated actions listed in FIA_UID.1[SC]]** on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2 [SC]	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 [SC]	**Single-use authentication mechanisms**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.4.1 [SC]	The TSF shall prevent reuse of authentication data related to **[assignment: the authentication mechanism used to open a secure communication channel with the card]**.

FTP_ITC.1 [SC]	**Inter-TSF trusted channel(SC)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1.1 [SC]	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 [SC]	The TSF shall permit **[selection: another trusted IT product]** to initiate communication via the trusted channel.
FTP_ITC.1.3 [SC]	The TSF shall initiate communication via the trusted channel for **[assignment: all card management functions including:** • loading; • installation; • extradition; • registry update; • content removal; • changing the Application Life Cycle or Card Life Cycle;

].

7.1.1.7 EMG Security Functional Requirements

Not used in this ST because EMG is optional in PP [11] and the TOE does not support EMG.

7.1.1.8 Further Security Functional Requirements

The SFRs in this section provide additional proprietary features.

FAU_SAS.1 [SCP]	**Audit Data Storage (SCP)**
Hierarchical to:	No other components.
Dependencies:	No other components.
FAU_SAS.1.1 [SCP]	The TSF shall provide **[assignment: the test process before TOE Delivery]** with the capability to store **[selection: the Initialisation Data, Prepersonalisation Data, [assignment: supplements of the Smartcard Embedded Software]]** in the **[assignment: audit records]**.
Application Note	This SFR performs selection and assignment operations on FAU_SAS.1 as defined in the Security IC Platform Protection Profile [10]. The test process is running under control of the test-personnel.

FIA_AFL.1 [PIN]	**Basic Authentication Failure Handling (PIN)**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FIA_AFL.1.1 [PIN]	The TSF shall detect when **[selection: an administrator configurable positive integer within [1 and 127]]** unsuccessful authentication attempts occur related to **[assignment: any user authentication using D.PIN]**.
FIA_AFL.1.2 [PIN]	When the defined number of unsuccessful authentication attempts has been **[selection: surpassed]**, the TSF shall **[assignment: block the authentication with D.PIN]**.
Application Note	The dependency with FIA_UAU.1 is not applicable. The TOE implements the firewall access control SFP, based on which access to the object implementing FIA_AFL.1[PIN] is organized.

FPT_EMS.1	**Emanation of TSF and User Data**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 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 31

Table 31. FPT_EMS.1.1 Table

Emissions	attack surface	TSF data	User data
variations in power consumption or timing during command execution	electrical contacts or RF field	TSF data D.CRYPTO	User data D.PIN, D.APP_KEYS

FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

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

Refinement The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, ”automatic response” means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.

Application Note This SFR is taken from the certified Security IC Platform Protection Profile [10].

FCS_CKM.2 Cryptographic key distribution

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation]

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[assignment:**

	methods: set keys and components of DES, AES, RSA, RSA-CRT, ECC, ECDH, HMAC, XEC, GenericSecret keys] that meets the following: **[assignment: [13], [45], [51], [57], [63], [69]]**.
Application Note	• The keys can be accessed as specified in [13] Key class and [45], [51], [57], [63], [69] for proprietary classes. • This component shall be instantiated according to the version of the Java Card API applying to the security target and the implementated algorithms [13] and [45], [51], [57], [63], [69] for proprietary classes.
FCS_CKM.3	**Cryptographic key access**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation].
FCS_CKM.3.1	The TSF shall perform **[assignment: management of DES, AES, RSA, RSA-CRT, ECC, ECDH, HMAC, XEC, and GenericSecret Keys]** in accordance with a specified cryptographic key access method **[assignment: methods/ commands defined in packages javacard.security of [13] and [45], [51], [57], [63], [69] for proprietary classes]** that meets the following: **[assignment: [13] and [45], [51], [57], [63], [69]]**.
Application Note	• The keys can be accessed as specified in [13] and [45], [51], [57], [63], [69] for proprietary classes. • This component shall be instantiated according to the version of the Java Card API applying to the security target and the implemented algorithms [13] and [45], [51], [57], [63], [69] for proprietary classes.
FDP_SDI.2 [SENSITIVE_ RESULT]	**Stored data integrity monitoring and action (Sensitive Result)**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring.
Dependencies:	No dependencies.
FDP_SDI.2.1 [SENSITIVE_ RESULT]	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: sensitive API result stored in the javacardx.security.SensitiveResult class]**.

FDP_SDI.2.2 [SENSITIVE_ RESULT]	Upon detection of a data integrity error, the TSF shall **[assignment: throw an exception]**.

7.1.1.9 Configuration Security Functional Requirements

FDP_IFC.2 [CFG] **Complete information flow control (CFG)**

Hierarchical to: FDP_IFC.1 Subset information flow control.

Dependencies: FDP_IFF.1 Simple security attributes.

FDP_IFC.2.1 [CFG] The TSF shall enforce the **[assignment: CONFIGURATION information flow control SFP]** on **[assignment: S.Customer, S.NXP, S.ConfigurationMechanism, and D.CONFIG_ITEM]** and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 [CFG] 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 [CFG] **Simple security attributes**

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation.

FDP_IFF.1.1 [CFG] The TSF shall enforce the **[assignment: CONFIGURATION information flow control SFP]** based on the following types of subject and information security attributes **[assignment:**

- S.Customer: security attributes Customer Configuration Token generation key
- S.NXP: security attributes NXP Configuration Token generation key
- S.ConfigurationMechanism: security attributes NXP Configuration Access, Customer Configuration Access
- D.CONFIG_ITEM: security attributes access privilege

].

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

- Read and write operations of D.CONFIG_ITEM between S.ConfigurationMechanism and S.NXP shall only be possible when S.NXP is authenticated with its token using the NXP Configuration Token generation key.

- Read and write operations of D.CONFIG_ITEM between S.ConfigurationMechanism and S.Customer shall only be possible when S.Customer is authenticated with its token using the Customer Configuration Token generation key and if access privilege allows it.
- Enabling or disabling of NXP Configuration Access between S.ConfigurationMechanism and S.NXP shall only be possible when S.NXP is authenticated with its token using the NXP Configuration Token generation key.

].

FDP_IFF.1.3 [CFG]	The TSF shall enforce the additional information flow control SFP rules **[assignment: none]**.
FDP_IFF.1.4 [CFG]	The TSF shall explicitly authorise an information flow based on the following rules: **[assignment: none]**.
FDP_IFF.1.5 [CFG]	The TSF shall explicitly deny an information flow based on the following rules: **[assignment:** • If the NXP Configuration Access is disabled then nobody can read or write D.CONFIG_ITEM. • If the Customer Configuration Access is disabled then S.Customer can not read or write D.CONFIG_ITEM. **]**.
Application note	GlobalPlatform Framework authentication mechnism is used to authenticate the tokens.

FIA_UID.1 [CFG] Timing of Identification (CFG)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1 [CFG]	The TSF shall allow **[assignment: to select the Runtime Configuration Interface]** on behalf of the user to be performed before the user is identified.
FIA_UID.1.2 [CFG]	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FMT_MSA.1 [CFG] Management of security attributes (CFG)

Hierarchical to:	No other components.

Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.
FMT_MSA.1.1 [CFG]	The TSF shall enforce the **[assignment: CONFIGURATION information flow control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: NXP Configuration Access and Customer Configuration Access]** to **[assignment: S.NXP and S.Customer]** respectively.

FMT_MSA.3 [CFG] Static attribute initialisation (CFG)

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.
FMT_MSA.3.1 [CFG]	The TSF shall enforce the **[assignment: CONFIGURATION information flow control SFP]** to provide **[restrictive]** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 [CFG]	The TSF shall allow the **[assignment: nobody]** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1 [CFG] Specification of Management Functions (CFG)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1 [CFG]	The TSF shall be capable of performing the following management functions: **[assignment: disable the NXP Configuration Access, disable the Customer Configuration Access]**.

FMT_SMR.1 [CFG] Security roles (CFG)

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1 [CFG]	The TSF shall maintain the roles **[assignment: S.NXP and S.Customer]**.
FMT_SMR.1.2 [CFG]	The TSF shall be able to associate users with roles.

Application note	The roles of the CONFIGURATION information flow control SFP are defined by the NXP Configuration Token generation key and the Customer Configuration Token generation key.

7.1.1.10 OS Update Security Functional Requirements

The SFRs in this section provide JCOP proprietary features.

FDP_IFC.2 [OSU]	**Complete information flow control (OSU)**
Hierarchical to:	FDP_IFC.1 Subset information flow control.
Dependencies:	FDP_IFF.1 Simple security attributes.
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**
Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation.
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:** • S.OSU: security attributes Current Sequence Number, Verification Key, Package Decryption Key • D.UPDATE_IMAGE: security attributes Received Sequence Number, Image Type **]**.
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:** • S.OSU shall only accept D.UPDATE_IMAGE which signature can be verified with Verification Key. • S.OSU shall only accept D.UPDATE_IMAGE for the update process that can be decrypted with Package Decryption Key. **]**.

FDP_IFF.1.3 [OSU]	The TSF shall enforce the additional information flow control SFP rules **[assignment: S.OSU shall only authorize D.UPDATE_IMAGE for the update process if the following rules apply:** • If Image Type equals Reset then Received Sequence Number shall equal Current Sequence Number. • If Image Type equals Upgrade then Received Sequence Number shall be higher than Current Sequence Number. • If Image Type equals Downgrade then Received Sequence Number shall be lower than Current Sequence Number. **]**.
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: D.UPDATE_IMAGE which is not included in the pre-loaded OS Update plan]**.
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.
Application note	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 (OSU)**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.
FMT_MSA.3.1 [OSU]	The TSF shall enforce the **[assignment: OS Update information flow control SFP]** to provide **[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 (OSU)**
Hierarchical to:	No other components.

Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.
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 Sequence Number]** to **[assignment: S.OSU]**.

FMT_SMR.1 [OSU] Security roles (OSU)

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
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 of the CONFIGURATION information flow control SFP are defined by the NXP Configuration Token generation key and the Customer Configuration Token generation key.

FMT_SMF.1 [OSU] Specification of Management Functions (OSU)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1 [OSU]	The TSF shall be capable of performing the following management functions: **[assignment:** • query Current Sequence Number • query Reference Sequence Number **]**.
Application note	After the atomic activation of the additional code the Final Sequence Number is returned on querying the Current Sequence Number.

FIA_UID.1 [OSU] Timing of Identification (OSU)

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_UID.1.1 [OSU]	The TSF shall allow **[assignment: OP.TRIGGER_UPDATE]** 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 (OSU)

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.1.1 [OSU]	The TSF shall allow **[assignment: OP.TRIGGER_UPDATE]** 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 (OSU)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 (OSU)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
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. **]**.

7.1.1.11 Restricted Mode Security Functional Requirements

The SFRs in this section provide JCOP proprietary features.

FDP_ACC.2 [RM]	**Complete access control (RM)**
Hierarchical to:	FDP_ACC.1 Subset access control.
Dependencies:	FDP_ACF.1 Security attribute based access control.
FDP_ACC.2.1 [RM]	The TSF shall enforce the **[assignment: Restricted Mode access control SFP]** on **[assignment: S.ACAdmin]** and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2 [RM]	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 [RM]	**Security attribute based access control (RM)**
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation.
FDP_ACF.1.1 [RM]	The TSF shall enforce the **[assignment: Restricted Mode access control SFP]** to objects based on the following **[assignment:** • S.ACAdmin: security attribute Attack Counter **]**.
FDP_ACF.1.2 [RM]	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[assignment:The Attack Counter can be reset by S.ACAdmin].**
FDP_ACF.1.3 [RM]	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment: none]**.
FDP_ACF.1.4 [RM]	The TSF shall explicitly deny access of subjects to objects based on the following additional rules:**[assignment: Deny all operations on all objects when the TOE is in restricted mode, except for operations listed in FMT_SMF.1[RM]]**.

FMT_MSA.3 [RM]	**Static attribute initialisation (RM)**
Hierarchical to:	No other components.

Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.
FMT_MSA.3.1 [RM]	The TSF shall enforce the **[assignment: Restricted Mode access control SFP]** to provide **[restrictive]** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 [RM]	The TSF shall allow the **[assignment: nobody]** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1 [RM] Management of security attributes (RM)

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.
FMT_MSA.1.1 [RM]	The TSF shall enforce the **[assignment: Restricted Mode access control]** to restrict the ability to **[selection: change_default, [assignment: reset]]** the security attributes **[assignment: Attack Counter]** to **[assignment: S.ACAdmin]**.

FMT_SMF.1 [RM] Specification of Management Functions (RM)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1 [RM]	The TSF shall be capable of performing the following management functions: **[assignment:** • reset Attack Counter. • select ISD. • authentication against the ISD. • initialize a Secure Channel with the card. • query the Serial Number (Unique ID for chip). • read Platform Identifier. • query the logging information. • read Secure Channel Sequence Counter. • read Current Sequence Number. **]**.

FIA_UID.1 [RM] Timing of Identification (RM)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1 [RM]	The TSF shall allow **[assignment:** • select ISD • identify the card • query the debug logging information • send Restricted Mode Unlock Request **]** on behalf of the user to be performed before the user is identified.
FIA_UID.1.2 [RM]	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 [RM]	**Timing of authentication (RM)**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.1.1 [RM]	The TSF shall allow **[assignment:** • OP.TRIGGER_UPDATE • identify the card • query the debug logging information • send Restricted Mode Unlock Request **]** on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2 [RM]	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

7.1.1.12 Context Separation Security Functional Requirements

The SFRs in this section provide JCOP proprietary features.

FDP_ACC.2 [CONTSEP]	**Complete access control (CONTSEP)**
Hierarchical to:	FDP_ACC.1 Subset access control.
Dependencies:	FDP_ACF.1 Security attribute based access control.

FDP_ACC.2.1 [CONTSEP]	The TSF shall enforce the **[assignment: Context Separation SFP]** on **[assignment: subject S.SMK, S.GuestOS and object O.SMK_Memory_region, O.GuestOS_Memory_Region]** and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2 [CONTSEP]	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 [CONTSEP] **Security attribute based access control (CONTSEP)**

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation.
FDP_ACF.1.1 [CONTSEP]	The TSF shall enforce the **[assignment: Context Separation SFP]** to objects based on the following **[assignment:** • Subjects: S.SMK, S.GuestOS • Objects: O.SMK_Memory_Region, O.GuestOS_Memory_Region • Security Attributes: Access Control Table **]**.
FDP_ACF.1.2 [CONTSEP]	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[assignment:** • One S.GuestOS can perform operation OP.CONT_ACCESS over its own O.GuestOS_Memory_Region. • One S.GuestOS can perform operation OP.CONT_ACCESS over O.GuestOS_Memory_Region of another S.GuestOS only if so authorized by S.SMK according to the Access Control Table. • S.GuestOS cannot perform operation OP.CONT_ACCESS over O.SMK_Memory_Region. • S.SMK can perform operation OP.CONT_ACCESS over its own O.SMK_Memory_Region. • S.SMK can perform operation OP.CONT_ACCESS over O.GuestOS_Memory_Region only if so authorized in Access Control Table. **]**.
FDP_ACF.1.3 [CONTSEP]	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment:** • S.GuestOS can perform operation OP.CONT_ACCESS over O.SMK_Memory_Region only through dedicated call gates mechanism.

].

FDP_ACF.1.4 [CONTSEP]	The TSF shall explicitly deny access of subjects to objects based on the following additional rules:**[assignment: None]**.

FMT_MSA.3 [CONTSEP] **Static attribute initialisation (CONTSEP)**

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles.
FMT_MSA.3.1 [CONTSEP]	The TSF shall enforce the **[assignment: Context Separation SFP]** to provide **[assignment: restrictive]** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 [CONTSEP]	The TSF shall allow the **[assignment: S.SMK]** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1 [CONTSEP] **Management of security attributes (CONTSEP)**

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions.
FMT_MSA.1.1 [CONTSEP]	The TSF shall enforce the **[assignment: Context Separation SPF]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: Access Control Table]** to **[assignment: S.SMK]**.

FMT_SMF.1 [CONTSEP] **Specification of Management Functions (CONTSEP)**

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1 [CONTSEP]	The TSF shall be capable of performing the following management functions: **[assignment: OP.Modifiction_Of_Access_Control_Table].**

FMT_SMR.1 [CONTSEP]	**Security roles (CONTSEP)**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FMT_SMR.1.1 [CONTSEP]	The TSF shall maintain the roles **[assignment:** • One S.SMK running in O.SMK_Memory_Region • Several S.GuestOS running in there own O.GuestOS_Memory_Region **]**.
FMT_SMR.1.2 [CONTSEP]	The TSF shall be able to associate users with roles.

FIA_UID.1 [CONTSEP]	**Timing of Identification (CONTSEP)**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1 [CONTSEP]	The TSF shall allow **[assignment: no action]** on behalf of the user to be performed before the user is identified.
FIA_UID.1.2 [CONTSEP]	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

7.1.2 Security Requirements Rationale

7.1.2.1 Identification

OT.SID

SFR	Rationale
FIA_UID.2[AID]	Subjects' identity is AID-based (applets, packages and CAP files) and is met by the SFR. Installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities and is met by the SFR.
FIA_USB.1[AID]	Subjects' identity is AID-based (applets, packages) and is met by the SFR. Installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities and is met by the SFR.
FMT_MSA.1[JCRE]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.

SFR	Rationale
FMT_MSA.1[JCVM]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.1[ADEL]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_ MSA.3[FIREWALL]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.3[JCVM]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.3[ADEL]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MTD.1[JCRE]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MTD.3[JCRE]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_SMF.1[ADEL]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FIA_ATD.1[AID]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FDP_ITC.2[CCM]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.1[SC]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.3[SC]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_SMF.1[SC]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.

7.1.2.2 Execution

OT.FIREWALL

SFR	Rationale
FDP_ ACC.2[FIREWALL]	The FIREWALL access control policy contributes to meet this objective.
FDP_ ACF.1[FIREWALL]	The FIREWALL access control policy contributes to meet this objective.
FDP_IFC.1[JCVM]	The JCVM information flow control policy contributes to meet this objective.
FDP_IFF.1[JCVM]	The JCVM information flow control policy contributes to meet this objective.
FMT_MSA.1[JCRE]	Contributes indirectly to meet this objective.
FMT_MSA.1[JCVM]	Contributes indirectly to meet this objective.
FMT_MSA.1[ADEL]	Contributes indirectly to meet this objective.

SFR	Rationale
FMT_ MSA.2[FIREWALL-JCVM]	Contributes indirectly to meet this objective.
FMT_ MSA.3[FIREWALL]	Contributes indirectly to meet this objective.
FMT_MSA.3[JCVM]	Contributes indirectly to meet this objective.
FMT_MSA.3[ADEL]	Contributes indirectly to meet this objective.
FMT_MTD.1[JCRE]	Contributes indirectly to meet this objective.
FMT_MTD.3[JCRE]	Contributes indirectly to meet this objective.
FMT_SMF.1	Contributes indirectly to meet this objective.
FMT_SMF.1[ADEL]	Contributes indirectly to meet this objective.
FMT_SMR.1	Contributes indirectly to meet this objective.
FMT_ SMR.1[INSTALLER]	Contributes indirectly to meet this objective.
FMT_SMR.1[ADEL]	Contributes indirectly to meet this objective.
FDP_ITC.2[CCM]	Contributes indirectly to meet this objective.
FMT_SMR.1[SD]	Contributes indirectly to meet this objective.
FMT_MSA.1[SC]	Contributes indirectly to meet this objective.
FMT_MSA.3[SC]	Contributes indirectly to meet this objective.
FMT_SMF.1[SC]	Contributes indirectly to meet this objective.

OT.GLOBAL_ARRAYS_CONFID

SFR	Rationale
FDP_IFC.1[JCVM]	The JCVM information flow control policy meets the objective by preventing 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.
FDP_IFF.1[JCVM]	The JCVM information flow control policy meets this objective by preventing 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.
FDP_RIP.1[OBJECTS]	Contributes to meet the objective by protecting the array parameters of remotely invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[ABORT]	Contributes to meet the objective by protecting the array parameters of remotely invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[APDU]	Contributes to meet this objective by fulfilling the clearing requirement of these arrays.
FDP_ RIP.1[GlobalArray]	Contributes to meet this objective by fulfilling the clearing requirement of these arrays.
FDP_RIP.1[bArray]	Contributes to meet this objective by fulfilling the clearing requirement of these arrays.

SFR	Rationale
FDP_RIP.1[KEYS]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.
FDP_ RIP.1[TRANSIENT]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[ADEL]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[ODEL]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.

OT.GLOBAL_ARRAYS_INTEG

SFR	Rationale
FDP_IFC.1[JCVM]	Contributes to meet the objective by preventing an application from keeping a pointer to the APDU buffer of the card or to the global byte array of the applet's install method. Such a pointer could be used to access and modify it when the buffer is being used by another application.
FDP_IFF.1[JCVM]	Contributes to meet the objective by preventing an application from keeping a pointer to the APDU buffer of the card or to the global byte array of the applet's install method. Such a pointer could be used to access and modify it when the buffer is being used by another application.

OT.ARRAY_VIEWS_CONFID

SFR	Rationale
FDP_IFC.1[JCVM]	The JCVM information flow control policy meets the objective by preventing an application from storing a reference to the array view or reading the content of an array view that don't have ATTR_READABLE_ VIEW security attribute.
FDP_IFF.1[JCVM]	The JCVM information flow control policy meets the objective by preventing an application from storing a reference to the array view or reading the content of an array view that don't have ATTR_READABLE_ VIEW security attribute.
FDP_ ACC.2[FIREWALL]	The FIREWALL access control SFP meets the objective by enforcing access control to array views without ATTR_READABLE_VIEW access attributes.
FDP_ ACF.1[FIREWALL]	The FIREWALL access control SFP meets the objective by enforcing access control to array views without ATTR_READABLE_VIEW access attributes.

OT.ARRAY_VIEWS_INTEG

SFR	Rationale
FDP_IFC.1[JCVM]	The JCVM information flow control policy meets the objective by preventing an application from storing a reference to the array view or altering the content of an array view that don't have ATTR_WRITABLE_VIEW security attribute.
FDP_IFF.1[JCVM]	The JCVM information flow control policy meets the objective by preventing an application from storing a reference to the array view or altering the content of an array view that don't have ATTR_WRITABLE_VIEW security attribute.
FDP_ACC.2[FIREWALL]	The FIREWALL access control SFP meets the objective by enforcing access control to array views without ATTR_WRITABLE_VIEW access attributes.
FDP_ACF.1[FIREWALL]	The FIREWALL access control SFP meets the objective by enforcing access control to array views without ATTR_WRITABLE_VIEW access attributes.

OT.NATIVE

SFR	Rationale
FDP_ACF.1[FIREWALL]	Covers this objective by ensuring that 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.

OT.OPERATE

SFR	Rationale
FAU_ARP.1	Contributes to meet this objective by detecting and blocking various failures or security violations during usual working.
FDP_ACC.2[FIREWALL]	Contributes to meet this objective by protecting the TOE through the FIREWALL access control policy.
FDP_ACF.1[FIREWALL]	Contributes to meet this objective by protecting the TOE through the FIREWALL access control policy.
FDP_ROL.1[FIREWALL]	Contributes to meet this objective by providing support for cleanly abort applets' installation, which belongs to the category security-critical parts and procedures protection.
FIA_AFL.1[PIN]	Contributes to meet the objective by protecting the authentication.
FIA_USB.1[AID]	Contributes to meet this objective by controlling the communication with external users and their internal subjects to prevent alteration of TSF data.
FPT_TDC.1	Contributes to meet this objective by protection in various ways against applets' actions.
FPT_RCV.3[INSTALLER]	Contributes to meet this objective by providing safe recovery from failure, which belongs to the category of security-critical parts and procedures protection.
FIA_ATD.1[AID]	Contributes to meet this objective by controlling the communication with external users and their internal subjects to prevent alteration of TSF data.

SFR	Rationale
FPT_FLS.1	Contributes to meet this objective by detecting and blocking various failures or security violations during usual working.
FPT_ FLS.1[INSTALLER]	Contributes to meet this objective by detecting and blocking various failures or security violations during usual working.
FPT_FLS.1[ADEL]	Contributes to meet this objective by detecting and blocking various failures or security violations during usual working.
FPT_FLS.1[ODEL]	Contributes to meet this objective by detecting and blocking various failures or security violations during usual working.
FDP_ITC.2[CCM]	Contributes to meet this objective by detecting and blocking various failures or security violations during usual working.

OT.REALLOCATION

SFR	Rationale
FDP_RIP.1[OBJECTS]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[ABORT]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[APDU]	Only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer and the global byte array input parameter (bArray) to an applet's install method. Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_ RIP.1[GlobalArray]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[bArray]	Only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer and the global byte array input parameter (bArray) to an applet's install method. Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[KEYS]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_ RIP.1[TRANSIENT]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[ADEL]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[ODEL]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.

OT.RESOURCES

SFR	Rationale
FAU_ARP.1	Contributes to meet this objective by detecting stack/memory overflows during execution of applications.
FDP_ ROL.1[FIREWALL]	Contributes to meet this objective by preventing that failed installations create memory leaks.

SFR	Rationale
FMT_MTD.1[JCRE]	Contributes to meet this objective since the TSF controls the memory management.
FMT_MTD.3[JCRE]	Contributes to meet this objective since the TSF controls the memory management.
FMT_SMF.1	Contributes to meet this objective since the TSF controls the memory management.
FMT_SMF.1[ADEL]	Contributes to meet this objective since the TSF controls the memory management.
FMT_SMR.1	Contributes to meet this objective since the TSF controls the memory management.
FMT_ SMR.1[INSTALLER]	Contributes to meet this objective since the TSF controls the memory management.
FMT_SMR.1[ADEL]	Contributes to meet this objective since the TSF controls the memory management.
FPT_ RCV.3[INSTALLER]	Contributes to meet this objective by preventing that failed installations create memory leaks.
FPT_FLS.1	Contributes to meet this objective by detecting stack/memory overflows during execution of applications.
FPT_ FLS.1[INSTALLER]	Contributes to meet this objective by detecting stack/memory overflows during execution of applications.
FPT_FLS.1[ADEL]	Contributes to meet this objective by detecting stack/memory overflows during execution of applications.
FPT_FLS.1[ODEL]	Contributes to meet this objective by detecting stack/memory overflows during execution of applications.
FMT_SMR.1[SD]	Contributes to meet this objective since the TSF controls the memory management.
FMT_SMF.1[SC]	Contributes to meet this objective since the TSF controls the memory management.

7.1.2.3 Services

OT.ALARM

SFR	Rationale
FAU_ARP.1	Contributes to meet this objective by defining TSF reaction upon detection of a potential security violation.
FPT_FLS.1	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur.
FPT_ FLS.1[INSTALLER]	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur.
FPT_FLS.1[ADEL]	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur.
FPT_FLS.1[ODEL]	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur.

OT.CIPHER

SFR	Rationale
FCS_CKM.1	Covers the objective directly.
FCS_CKM.2	Covers the objective directly.
FCS_CKM.3	Covers the objective directly.
FCS_CKM.6	Covers the objective directly.
FCS_COP.1	Covers the objective directly.
FPR_UNO.1	Contributes to meet the objective by controlling the observation of the cryptographic operations which may be used to disclose the keys.

OT.KEY-MNGT

SFR	Rationale
FCS_CKM.1	Covers the objective directly.
FCS_CKM.2	Covers the objective directly.
FCS_CKM.3	Covers the objective directly.
FCS_CKM.6	Covers the objective directly.
FCS_COP.1	Covers the objective directly.
FDP_RIP.1[OBJECTS]	Covers the objective directly.
FDP_RIP.1[ABORT]	Covers the objective directly.
FDP_RIP.1[APDU]	Covers the objective directly.
FDP_ RIP.1[GlobalArray]	Covers the objective directly.
FDP_RIP.1[bArray]	Covers the objective directly.
FDP_RIP.1[KEYS]	Covers the objective directly.
FDP_ RIP.1[TRANSIENT]	Covers the objective directly.
FDP_RIP.1[ADEL]	Covers the objective directly.
FDP_RIP.1[ODEL]	Covers the objective directly.
FDP_SDI.2[DATA]	Covers the objective directly.
FPR_UNO.1	Contributes to meet objective by controlling the observation of the cryptographic operations which may be used to disclose the keys.

OT.PIN-MNGT

SFR	Rationale
FDP_ ACC.2[FIREWALL]	Contributes to meet the objective by protecting the access to private and internal data of the objects.
FDP_ ACF.1[FIREWALL]	Contributes to meet the objective by protecting the access to private and internal data of the objects.
FDP_RIP.1[OBJECTS]	Contributes to meet the objective.
FDP_RIP.1[ABORT]	Contributes to meet the objective.
FDP_RIP.1[APDU]	Contributes to meet the objective.

SFR	Rationale
FDP_ RIP.1[GlobalArray]	Contributes to meet the objective.
FDP_RIP.1[bArray]	Contributes to meet the objective.
FDP_RIP.1[KEYS]	Contributes to meet the objective.
FDP_ RIP.1[TRANSIENT]	Contributes to meet the objective.
FDP_RIP.1[ADEL]	Contributes to meet the objective.
FDP_RIP.1[ODEL]	Contributes to meet the objective.
FDP_ ROL.1[FIREWALL]	Contributes to meet the objective.
FDP_SDI.2[DATA]	Contributes to meet the objective.
FPR_UNO.1	Contributes to meet the objective.
FIA_AFL.1[PIN]	Directly contributes to meet the objective.

OT.TRANSACTION

SFR	Rationale
FDP_RIP.1[OBJECTS]	Covers the objective directly.
FDP_RIP.1[ABORT]	Covers the objective directly.
FDP_RIP.1[APDU]	Covers the objective directly.
FDP_ RIP.1[GlobalArray]	Covers the objective directly.
FDP_RIP.1[bArray]	Covers the objective directly.
FDP_RIP.1[KEYS]	Covers the objective directly.
FDP_ RIP.1[TRANSIENT]	Covers the objective directly.
FDP_RIP.1[ADEL]	Covers the objective directly.
FDP_RIP.1[ODEL]	Covers the objective directly.
FDP_ ROL.1[FIREWALL]	Covers the objective directly.

7.1.2.4 Object Deletion

OT.OBJ-DELETION

SFR	Rationale
FDP_RIP.1[ODEL]	Contributes to meet the objective.
FPT_FLS.1[ODEL]	Contributes to meet the objective.

7.1.2.5 Applet Management

OT.APPLI-AUTH

SFR	Rationale
FCS_COP.1	Refinement: applies to FCS_COP.1[DAP]. Contributes to meet the security objective by ensuring that the loaded Executable Application is legitimate by specifying the algorithm to be used in order to verify the DAP signature of the Verification Authority.
FDP_ROL.1[CCM]	Contributes to meet this security objective by ensures that card management operations may be cleanly aborted.
FPT_FLS.1[CCM]	Contributes to meet the security objective by preserving a secure state when failures occur.

OT.DOMAIN-RIGHTS

SFR	Rationale
FDP_ACC.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FDP_ACF.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_MSA.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_MSA.3[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_SMF.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_SMR.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FTP_ITC.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FCO_NRO.2[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FDP_IFC.2[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FDP_IFF.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_MSA.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_MSA.3[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.

SFR	Rationale
FMT_SMF.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UID.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UAU.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UAU.4[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.

OT.COMM_AUTH

SFR	Rationale
FCS_COP.1	Contributes to meet the security objective by specifying secure cryptographic algorithm that shall be used to determine the origin of the card management commands.
FMT_SMR.1[SD]	Contributes to meet the security objective by specifying the authorized identified roles enabling to send and authenticate card management commands.
FTP_ITC.1[SC]	Contributes to meet the security objective by ensuring the origin of card administration commands.
FDP_IFC.2[SC]	Contributes to meet the security objective by specifying the authorized identified roles enabling to send and authenticate card management commands.
FDP_IFF.1[SC]	Contributes to meet the security objective by specifying the authorized identified roles enabling to send and authenticate card management commands.
FMT_MSA.1[SC]	Contributes to meet the security objective by specifying security attributes enabling to authenticate card management requests.
FMT_MSA.3[SC]	Contributes to meet the security objective by specifying security attributes enabling to authenticate card management requests.
FIA_UID.1[SC]	Contributes to meet the security objective by specifying the actions that can be performed before authenticating the origin of the APDU commands that the TOE receives.
FIA_UAU.1[SC]	Contributes to meet the security objective by specifying the actions that can be performed before authenticating the origin of the APDU commands that the TOE receives.

OT.COMM_INTEGRITY

SFR	Rationale
FCS_COP.1	Contributes to meet the security objective by by specifying secure cryptographic algorithm that shall be used to ensure the integrity of the card management commands.

SFR	Rationale
FMT_SMR.1[SD]	Contributes to cover this security objective by defining the roles enabling to send and authenticate the card management requests for which the integrity has to be ensured.
FTP_ITC.1[SC]	Contributes to meet the security objective by ensuring the integrity of card management commands.
FDP_IFC.2[SC]	Contributes to cover the security objective by enforcing the Secure Channel Protocol information flow control policy to guarantee the integrity of administration requests.
FDP_IFF.1[SC]	Contributes to cover the security objective by enforcing the Secure Channel Protocol information flow control policy to guarantee the integrity of administration requests.
FMT_MSA.1[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the integrity of card management requests.
FMT_MSA.3[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the integrity of card management requests.
FMT_SMF.1[SC]	Contributes to meet the security objective by specifying the actions activating the integrity check on the card management commands.

OT.COMM_CONFIDENTIALITY

SFR	Rationale
FCS_COP.1	Contributes to meet this objective by specifying secure cryptographic algorithm that shall be used to ensure the confidentiality of the card management commands.
FMT_SMR.1[SD]	Contributes to cover the security objective by defining the roles enabling to send and authenticate the card management requests for which the confidentiality has to be ensured.
FTP_ITC.1[SC]	Contributes to cover the security objective by ensuring the confidentiality of card management commands.
FDP_IFC.2[SC]	Contributes to cover the security objective by enforcing the Secure Channel Protocol information flow control policy to guarantee the confidentiality of administration requests.
FDP_IFF.1[SC]	Contributes to cover the security objective by enforcing the Secure Channel Protocol information flow control policy to guarantee the confidentiality of administration requests.
FMT_MSA.1[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the confidentiality of card management requests by decrypting those requests and imposing management conditions on that attributes.
FMT_MSA.3[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the confidentiality of card management requests by decrypting those requests and imposing management conditions on that attributes.
FMT_SMF.1[SC]	Contributes to cover the security objective by specifying the actions ensuring the confidentiality of the card management commands.

7.1.2.6 Card Management

OT.CARD-MANAGEMENT

SFR	Rationale
FDP_ACC.2[ADEL]	Contributes to meet the objective by the ADEL access control policy which ensures the non-introduction of security holes. The integrity and confidentiality of data that does not belong to the deleted applet or package is a by-product of this policy as well.
FDP_ACF.1[ADEL]	Contributes to meet the objective by the ADEL access control policy which ensures the non-introduction of security holes. The integrity and confidentiality of data that does not belong to the deleted applet or package is a by-product of this policy as well.
FDP_RIP.1[ADEL]	Contributes to meet the objective by ensuring the non-accessibility of deleted data.
FMT_MSA.1[ADEL]	Contributes to meet the objective by enforcing the ADEL access control SFP.
FMT_MSA.3[ADEL]	Contributes to meet the objective by enforcing the ADEL access control SFP.
FMT_SMR.1[ADEL]	Contributes to meet the objective by maintaing the role applet deletion manager.
FPT_ RCV.3[INSTALLER]	Contributes to meet the objective by protecting the TSFs against possible failures of the deletion procedures.
FPT_ FLS.1[INSTALLER]	Contributes to meet the objective by protecting the TSFs against possible failures of the installer.
FPT_FLS.1[ADEL]	Contributes to meet the objective by protecting the TSFs against possible failures of the deletion procedures.
FDP_UIT.1[CCM]	Contributes to meet the objective by enforcing the Secure Channel Protocol information flow control policy and the Security Domain access control policy which controls the integrity of the corresponding data.
FDP_ROL.1[CCM]	Contributes to meet this security objective by ensures that card management operations may be cleanly aborted.
FDP_ITC.2[CCM]	Contributes to meet the security objective by enforcing the Firewall access control policy and the Secure Channel Protocol information flow policy when importing card management data.
FPT_FLS.1[CCM]	Contributes to meet the security objective by preserving a secure state when failures occur.
FDP_ACC.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FDP_ACF.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_MSA.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_MSA.3[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.

SFR	Rationale
FMT_SMF.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_SMR.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FTP_ITC.1[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FCO_NRO.2[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FDP_IFC.2[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FDP_IFF.1[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_MSA.1[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_MSA.3[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_SMF.1[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UID.1[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UAU.1[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UAU.4[SC]	Contributes to meet this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.

7.1.2.7 Smart Card Platform

OT.SCP.IC

SFR	Rationale
FAU_ARP.1	Contributes to the coverage of the objective by resetting the card session or terminating the card in case of physical tampering.
FPR_UNO.1	Contributes to the coverage of the objective by ensuring leakage resistant implementations of the unobservable operations.
FPT_EMS.1	Contributes to meet the objective.

SFR	Rationale
FPT_PHP.3	Contributes to the coverage of the objective by preventing bypassing, deactivation or changing of other security features.

OT.SCP.RECOVERY

SFR	Rationale
FAU_ARP.1	Contributes to the coverage of the objective by ensuring reinitialization of the Java Card System and its data after card tearing and power failure.
FPT_FLS.1	Contributes to the coverage of the objective by preserving a secure state after failure.

OT.SCP.SUPPORT

SFR	Rationale
FCS_CKM.1	Contributes to meet the objective.
FCS_CKM.6	Contributes to meet the objective.
FCS_COP.1	Contributes to meet the objective.
FDP_ ROL.1[FIREWALL]	Contributes to meet the objective.

OT.IDENTIFICATION

SFR	Rationale
FAU_SAS.1[SCP]	Covers the objective.The Initialisation Data (or parts of them) are used for TOE identification

7.1.2.8 Random Numbers

OT.RND

SFR	Rationale
FCS_RNG.1	Covers the objective by providing random numbers of good quality by specifying class DRG.3 of AIS 20. It was chosen to define FCS_RNG.1 explicitly, because Part 2 of the Common Criteria does not contain generic security functional requirements for Random Number generation. (Note that there are security functional requirements in Part 2 of the Common Criteria, which refer to random numbers. However, they define requirements only for the authentication context, which is only one of the possible applications of random numbers).
FCS_RNG.1[HDT]	Covers the objective by providing random numbers of good quality by specifying class DRG.4 of AIS 20

7.1.2.9 Config Applet

OT.CARD-CONFIGURATION

SFR	Rationale
FDP_IFC.2[CFG]	Contributes to meet the objective by controlling the ability to modify configuration items.

SFR	Rationale
FDP_IFF.1[CFG]	Contributes to meet the objective by controlling the ability to modify configuration items.
FMT_MSA.3[CFG]	Contributes to meet the objective by controlling the ability to modify configuration items.
FMT_MSA.1[CFG]	Contributes to meet the objective by controlling the ability to modify configuration items.
FMT_SMR.1[CFG]	Contributes to meet the objective by controlling the ability to modify configuration items.
FMT_SMF.1[CFG]	Contributes to meet the objective by controlling the ability to modify configuration items.
FIA_UID.1[CFG]	Contributes to meet the objective by requiring identification before modifying configuration items.

7.1.2.10 OS Update Mechanism

OT.CONFID-UPDATE-IMAGE.LOAD

SFR	Rationale
FPR_UNO.1	Contributes to the coverage of the objective by ensuring the unobservability of the S.OSU decryption key.
FIA_UID.1[OSU]	Contributes to the coverage of the objective by requiring identification.
FIA_UAU.1[OSU]	Contributes to the coverage of the objective by requiring authentication.

OT.AUTH-LOAD-UPDATE-IMAGE

SFR	Rationale
FDP_IFC.2[OSU]	Contributes to the coverage of the objective by applying the rules of the Information Flow Control policy.
FDP_IFF.1[OSU]	Contributes to the coverage of the objective by applying the rules of the Information Flow Control policy.
FMT_MSA.3[OSU]	Contributes to the coverage of the objective by enforcing restrictive default values for the attributes of the OS Update information flow control SFP.
FMT_SMR.1[OSU]	Contributes to the coverage of the objective by letting S.OSU handle the OS Update procedure.
FIA_UID.1[OSU]	Contributes to the objective by requiring identification of the authorized images.
FIA_UAU.1[OSU]	Contributes to the objective by requiring authentication of the authorized images.

OT.SECURE_LOAD_ACODE

SFR	Rationale
FDP_IFC.2[OSU]	Contributes to the coverage of the objective by ensuring that only allowed versions of the D.UPDATE_IMAGE are accepted and by checking the evidence data of authenticity and integrity.

SFR	Rationale
FMT_SMR.1[OSU]	Contributes to the coverage of the objective by letting S.OSU handle the OS Update procedure.
FPT_FLS.1[OSU]	Contributes to the coverage of the objective by ensuring a secure state after interruption of the OS Update procedure (Load Phase).
FIA_UAU.4[OSU]	Contributes to meet the objective by enforcing authenticity and integrity of D.UPDATE_IMAGE (i.e. Additional Code).

OT.SECURE_AC_ACTIVATION

SFR	Rationale
FMT_MSA.1[OSU]	Contributes to the coverage of the objective by allowing to modify the Current Sequence Number only after successful OS Update procedure.
FMT_SMR.1[OSU]	Contributes to the coverage of the objective by letting S.OSU handle the OS Update procedure.
FMT_SMF.1[OSU]	Contributes to the objective by providing information on the currently activated software (Current Sequence Number).
FPT_FLS.1[OSU]	Contributes to the coverage of the objective by ensuring atomicity of the OS Update procedure (Load Phase).

OT.TOE_IDENTIFICATION

SFR	Rationale
FDP_SDI.2	Contributes to cover the objective by storing the identification data (D.TOE_IDENTIFICATION) in an integrity protected store.
FMT_SMF.1[OSU]	Contributes to cover the objective by providing the ability to query the identification data (Current Sequence Number, Reference Sequence Number, Final Sequence Number) of the TOE.

7.1.2.11 Restricted Mode

OT.ATTACK-COUNTER

SFR	Rationale
FMT_MSA.3[RM]	Contributes to cover the objective by restricting the initial value of the Attack Counter and allowing nobody to change the initial value.
FMT_MSA.1[RM]	Contributes to cover the objective by only allowing the S.ACAdmin to modify the Attack Counter.
FIA_UAU.1[RM]	Contributes to cover the objective by requiring authentication before resetting the Attack Counter.
FIA_UID.1[RM]	Contributes to cover the objective by requiring identification before resetting the Attack Counter.

OT.RESTRICTED-MODE

SFR	Rationale
FDP_ACC.2[RM]	Contributes to the coverage of the objective by defining the subject of the Restricted Mode access control SFP.

SFR	Rationale
FDP_ACF.1[RM]	Contributes to cover the objective by controlling access to objects for all operations.
FMT_SMF.1[RM]	Contributes to cover the objective by defining the management functions of the restricted mode.
FIA_UAU.1[RM]	Contributes to cover the objective by requiring authentication before resetting the Attack Counter.
FIA_UID.1[RM]	Contributes to cover the objective by requiring identification before resetting the Attack Counter.

7.1.2.12 Package Sensitive Result

OT.SENSITIVE_RESULTS_INTEG

SFR	Rationale
FDP_ SDI.2[SENSITIVE_ RESULT]	The security objective is covered directly by the SFR FDP_ SDI.2[SENSITIVE_RESULT] which ensures that integrity errors related to the senstive API result are detected by the TOE.

7.1.2.13 Context Separation

OT.CONT-SEP

SFR	Rationale
FDP_ ACC.2[CONTSEP]	Contributes to cover the objective by defining the context separation SFP.
FDP_ ACF.1[CONTSEP]	Contributes to cover the objective by defining the rules of the context separation SFP.
FMT_ MSA.3[CONTSEP]	Contributes to cover the objective by providing restrictive default values for the Access Control Table and by allowing only the SMK to create new entries.
FMT_ MSA.1[CONTSEP]	Contributes to cover the objective by allowing only SMK to modify the Memory Region Access Control Table.
FMT_ SMR.1[CONTSEP]	Contributes to cover the objective by maintaining the roles S.SMK, S.GuestOS.
FMT_ SMF.1[CONTSEP]	Contributes to cover the objective by defining a management function for the Memory Region Access Control Table.
FIA_UID.1[CONTSEP]	Contributes to cover the objective by ensuring that no user can access the TOE before the context separation SFP has been set up

OT.CONT-PRIV

SFR	Rationale
FDP_ ACC.2[CONTSEP]	Contributes to cover the objective by defining the context separation SFP.
FDP_ ACF.1[CONTSEP]	Contributes to cover the objective by defining the rules that makes SMK the most privileged.

SFR	Rationale
FMT_ MSA.3[CONTSEP]	Contributes to cover the objective by providing restrictive default values for the Access Control Table and by allowing only the SMK to create new entries.
FMT_ MSA.1[CONTSEP]	Contributes to cover the objective by allowing only SMK to modify the Memory Region Access Control Table.
FMT_ SMR.1[CONTSEP]	Contributes to cover the objective by maintaining the roles S.SMK, S.GuestOS.
FMT_ SMF.1[CONTSEP]	Contributes to cover the objective by defining a management function for the Memory Region Access Control Table.
FIA_UID.1[CONTSEP]	Contributes to cover the objective by ensuring that no user can access the TOE before the context separation SFP has been set up

OT.CONT-DOS

SFR	Rationale
FDP_ ACC.2[CONTSEP]	Contributes to cover the objective by defining the context separation SFP.
FDP_ ACF.1[CONTSEP]	Contributes to cover the objective by defining the rules that ensure that SMK stays the leader and manages context switching.
FMT_ MSA.3[CONTSEP]	Contributes to cover the objective by providing restrictive default values for the Access Control Table and by allowing only the SMK to create new entries.
FMT_ MSA.1[CONTSEP]	Contributes to cover the objective by allowing only SMK to modify the Memory Region Access Control Table.
FMT_ SMR.1[CONTSEP]	Contributes to cover the objective by maintaining the roles S.SMK, S.GuestOS.
FMT_ SMF.1[CONTSEP]	Contributes to cover the objective by defining a management function for the Memory Region Access Control Table.
FIA_UID.1[CONTSEP]	Contributes to cover the objective by ensuring that no user can access the TOE before the context separation SFP has been set up

7.1.3 Security Requirements Dependencies

Table 32. SFRs Dependencies.

Requirements	CC Dependencies	Satisfied dependencies
FAU_ARP.1	FAU_SAA.1 Potential violation analysis	see §7.3.3.1 of [11]
FAU_SAS.1[SCP]	No other components.	
FCO_NRO.2[SC]	FIA_UID.1 Timing of identification	FIA_UID.1[SC]

Table 32. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation], [FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers], FCS_ CKM.6 Timing and event of cryptographic key destruction	see §7.3.3.1 of [11]
FCS_CKM.2	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation]	FCS_CKM.1
FCS_CKM.3	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation]	FCS_CKM.1
FCS_CKM.6	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	see §7.3.3.1 of [11]
FCS_COP.1	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.6 Cryptographic key destruction.	see §7.3.3.1 of [11]
FCS_RNG.1	No dependencies	
FCS_RNG.2[HDT]	No dependencies	
FDP_ACC.1[SD]	FDP_ACF.1 Security attribute based access control	FDP_ACF.1[SD]
FDP_ACC.2[FIREWALL]	FDP_ACF.1 Security attribute based access control	see §7.3.3.1 of [11]
FDP_ACC.2[ADEL]	FDP_ACF.1 Security attribute based access control	see §7.3.3.1 of [11]
FDP_ACC.2[RM]	FDP_ACF.1 Security attribute based access control	FDP_ACF.1[RM]

Table 32. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
FDP_ACC.2[CONTSEP]	FDP_ACF.1 Security attribute based access control	FDP_ACF.1[CONTSEP]
FDP_ACF.1[FIREWALL]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	see §7.3.3.1 of [11]
FDP_ACF.1[ADEL]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	see §7.4.3.1 of [11]
FDP_ACF.1[SD]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1[SD] FMT_MSA.3[SD]
FDP_ACF.1[RM]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.2[RM] FMT_MSA.3[RM]
FDP_ACF.1[CONTSEP]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.2[CONTSEP] FMT_MSA.3[CONTSEP]
FDP_IFC.1[JCVM]	FDP_IFF.1 Simple security attributes	see §7.3.3.1 of [11]
FDP_IFC.2[SC]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[SC]
FDP_IFC.2[OSU]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[OSU]
FDP_IFC.2[CFG]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[CFG]
FDP_IFC.2[CFG]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[CFG]
FDP_IFF.1[JCVM]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	see §7.3.3.1 of [11]
FDP_IFF.1[SC]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.2[SC] FMT_MSA.3[SC]
FDP_IFF.1[OSU]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.2[OSU] FMT_MSA.3[OSU]
FDP_IFF.1[CFG]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.2[CFG] FMT_MSA.3[CFG]
FDP_ITC.2[CCM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1[SD] FTP_ITC.1[SC] FPT_TDC.1
FDP_RIP.1[OBJECTS]	No dependencies	

Table 32. SFRs Dependencies*...continued*

Requirements	CC Dependencies	Satisfied dependencies
FDP_RIP.1[ABORT]	No dependencies	
FDP_RIP.1[APDU]	No dependencies	
FDP_RIP.1[bArray]	No dependencies	
FDP_RIP.1[KEYS]	No dependencies	
FDP_RIP.1[TRANSIENT]	No dependencies	
FDP_RIP.1[ADEL]	No dependencies	
FDP_RIP.1[ODEL]	No dependencies	
FDP_ROL.1[FIREWALL]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	see §7.3.3.1 of [11]
FDP_ROL.1[CCM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FDP_ACC.1[SD]
FDP_SDI.2[DATA]	No dependencies	
FDP_UIT.1[CCM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	FDP_ACC.1[SD] FTP_ITC.1[SC]
FIA_AFL.1[PIN]	FIA_UAU.1 Timing of authentication	see AppNote in FIA_ AFL.1[PIN]
FIA_ATD.1[AID]	No dependencies	
FIA_UID.1[SC]	No dependencies	
FIA_UID.1[OSU]	No dependencies	
FIA_UID.1[CFG]	No dependencies	
FIA_UID.1[RM]	No dependencies	
FIA_UID.1[CONTSEP]	No dependencies	
FIA_UID.2[AID]	No dependencies	
FIA_USB.1[AID]	FIA_ATD.1 User attribute definition	see §7.3.3.1 of [11]
FIA_UAU.1[SC]	A_UID.1 Timing of identification	FIA_UID.1[SC]
FIA_UAU.1[RM]	FIA_UID.1 Timing of identification	FIA_UID.1[RM]
FIA_UAU.1[OSU]	FIA_UID.1 Timing of identification	FIA_UID.1[OSU]
FIA_UAU.4[SC]	No dependencies	
FIA_UAU.4[OSU]	No dependencies	

Table 32. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
FMT_MSA.1[JCRE]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [11]
FMT_MSA.1[JCVM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [11]
FMT_MSA.1[ADEL]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [11]
FMT_MSA.1[SC]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1[SD] FMT_SMR.1[SD] FMT_SMF.1[SC]
FMT_MSA.1[OSU]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_IFC.2[OSU] FMT_SMR.1[OSU] FMT_SMF.1[OSU]
FMT_MSA.1[CFG]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_IFC.2[CFG] FMT_SMR.1[CFG] FMT_SMF.1[CFG]
FMT_MSA.1[SD]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1[SD] FMT_SMR.1[SD] FMT_SMF.1[SD]
FMT_MSA.1[RM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.2[RM] FMT_SMR.1[SD] FMT_SMF.1[RM]

Table 32. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
FMT_MSA.1[CONTSEP]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.2[CONTSEP] FMT_SMR.1[CONTSEP] FMT_SMF.1[CONTSEP]
FMT_MSA.2[FIREWALL-JCVM]	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [11]
FMT_MSA.3[FIREWALL]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [11]
FMT_MSA.3[JCVM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [11]
FMT_MSA.3[ADEL]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [11]
FMT_MSA.3[OSU]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[OSU] FMT_SMR.1[OSU]
FMT_MSA.3[CFG]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[CFG] FMT_SMR.1[CFG]
FMT_MSA.3[SD]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SD] FMT_SMR.1[SD]
FMT_MSA.3[SC]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SC] FMT_SMR.1[SD]
FMT_MSA.3[RM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[RM] FMT_SMR.1[SD]
FMT_MSA.3[CONTSEP]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[CONTSEP] FMT_SMR.1[CONTSEP]
FMT_MTD.1[JCRE]	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [11]
FMT_MTD.3[JCRE]	FMT_MTD.1 Management of TSF data	see §7.3.3.1 of [11]
FMT_SMF.1	No dependencies	
FMT_SMF.1[ADEL]	No dependencies	
FMT_SMF.1[OSU]	No dependencies	

Table 32. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
FMT_SMF.1[CFG]	No dependencies	
FMT_SMF.1[SD]	No dependencies	
FMT_SMF.1[SC]	No dependencies	
FMT_SMF.1[RM]	No dependencies	
FMT_SMF.1[CONTSEP]	No dependencies	
FMT_SMR.1	FIA_UID.1 Timing of identification	see §7.3.3.1 of [11]
FMT_SMR.1[INSTALLER]	FIA_UID.1 Timing of identification	see §7.3.3.1 of [11]
FMT_SMR.1[ADEL]	FIA_UID.1 Timing of identification	see §7.3.3.1 of [11]
FMT_SMR.1[OSU]	FIA_UID.1 Timing of identification	FIA_UID.1[OSU]
FMT_SMR.1[CFG]	FIA_UID.1 Timing of identification	FIA_UID.1[CFG]
FMT_SMR.1[SD]	FIA_UID.1 Timing of identification	FIA_UID.1[SC]
FMT_SMR.1[CONTSEP]	FIA_UID.1 Timing of identification	FIA_UID.1[CONTSEP]
FPR_UNO.1	No dependencies	
FPT_EMS.1	No dependencies	
FPT_FLS.1	No dependencies	
FPT_FLS.1[INSTALLER]	No dependencies	
FPT_FLS.1[ADEL]	No dependencies	
FPT_FLS.1[ODEL]	No dependencies	
FPT_FLS.1[OSU]	No dependencies	
FPT_FLS.1[CCM]	No dependencies	
FPT_TDC.1	No dependencies	
FPT_RCV.3[INSTALLER]	AGD_OPE.1 Operational user guidance	see §7.3.3.1 of [11]
FPT_PHP.3	No dependencies	
FTP_ITC.1[SC]	No dependencies	

7.1.4 Rationale for Exclusion of Dependencies

The dependency FIA_UID.1 of FMT_SMR.1[INSTALLER] is unsupported. This ST 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 unsupported. This ST 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 unsupported. 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 unsupported. 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 this ST.

The dependency FIA_UAU.1 of FIA_AFL.1[PIN] is unsupported. The TOE implements the firewall access control SFP, based on which access to the object Implementing FIA_AFL.1[PIN] is organized.

7.2 Security Functional Requirements for eUICC

7.2.1 Security Functional 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 eUICC PP [12]. Note that the cryptographic SFRs are provided in the JavaCard section Section 7.1.1

The following table provides the selection and assignments for SFRs:

Table 33.

SFR ID	Selection / Assignment text	Selection / Assignment value
FIA_UID.1.1 [EXT]	[assignment: list of additional TSF mediated actions]	no additional TSF mediated actions
FIA_UAU.1.1 [EXT]	[assignment: list of additional TSF mediated actions]	no additional TSF mediated actions
FIA_USB.1.1 [EXT]	[selection: eIM ID is associated to S.ISD-R, acting on behalf of U.EIM, no other associations]	no other associations
FIA_USB.1.2 [EXT]	[selection: Initial association of eIM ID requires U.EIM to be authenticated via CERT.EIM.ECDSA (SGP.32), no other initial associations]	no other initial associations
FIA_USB.1.3 [EXT]	[selection: change of eIM ID requires U.EIM to be authenticated via CERT.EIM.ECDSA (SGP.32), no other changes]	no other changes
FIA_UAU.4.1 [EXT]	[selection: none, U.EIM]	none
FIA_UID.1.1 [MNO-SD]	[assignment: list of TSF-mediated actions]	• application selection • requesting data that identifies the eUICC

Table 33. *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FIA_ATD.1.1 [Base]	[selection: CERT.EIM.ECDSA and eIM ID belonging to U.EIM, no additional attributes]	no additional attributes
FDP_IFF.1.3 [SCP]	[assignment: additional information flow control SFP rules]	no additional information flow control SFP rules
FDP_IFF.1.4 [SCP]	[assignment: rules, based on security attributes, that explicitly authorise information flows]	none
FTP_ITC.1.3 [SCP]	[assignment: list of functions for which a trusted channel is required]	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 remote OTA Platform to open a SCP80 or SCP81 secure channel to transmit the following operation: ES6.UpdateMetadata.
FDP_ITC.2.5 [SCP]	[assignment: additional importation control rules]	none
FPT_TDC.1.2 [SCP]	[assignment: list of interpretation rules to be applied by the TSF]	the rules defined in GSMA SGP.22 Specification [31]
FCS_ CKM.1.1[SCP-SM]	[assignment: at least one elliptic curve referenced in SGP.22 or SGP.32[32]]	ECKA-EG as described in [31] (annex G) using one of the following standardized curves: o NIST P-256 (FIPS PUB 186-3 Digital Signature Standard) o brainpoolP256r1 (BSI TR-03111, Version 1.11, RFC 5639) o FRP256V1 (ANSSI ECC FRP256V1)
FCS_CKM.2 [SCP-MNO]	[assignment: cryptographic key distribution method]	set keys and components of DES, AES, RSA, RSA-CRT, EC, secure messaging and Network Authentication Algorithms EC
	[assignment: list of standards]	[13], [45], [51], [57], [46], [52], [58]
FCS_CKM.6.1 [SCP-SM]	[selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]	no longer needed
FCS_CKM.6.2 [SCP-SM]	[assignment: cryptographic key destruction method]	physically overwriting the keys in a randomized manner
	[assignment: list of standards]	none

Table 33. *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FCS_CKM.6.1 [SCP-MNO]	[selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]	no longer needed
FCS_CKM.6.2 [SCP-MNO]	[assignment: cryptographic key destruction method]	physically overwriting the keys in a randomized manner
	[assignment: list of standards]	none
FDP_ACF.1.1 [ISDR]	[Selection: “Reference Enterprise Rule”, no additional attributes]	no additional attributes
FDP_ACF.1.2 [ISDR]	[Selection: the Reference Enterprise Rule allows enabling S.ISD-P, no additional conditions]	no additional conditions
FDP_ACF.1.3 [ISDR]	[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]	ISDR shall perform the following operations: • 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) based on Profile ”state” and profile policy rules ”PPR”
FDP_ACF.1.4 [ISDR]	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	when any of the defined rules by SGP.22 Specification [31] related to Profile ”state” and profile policy rules ”PPR” do not hold
FDP_ACC.1.1 [ECASD]	[assignment: additional list of subjects, objects, and operations between subjects and objects covered by the SFP]	• additional operations defined by the interfaces ES8+ (SM-DP+ – eUICC), and ES10x (LPA – eUICC) • creation of an eUICC signature on material provided by an ISD-R
FDP_ACF.1.1 [ECASD]	[assignment: additional list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]	OT.SECURE-CHANNELS, OT.INTERNALSECURE-CHANNELS
FDP_ACF.1.2 [ECASD]	[assignment: additional rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	Rules defined in GSMA SGP.22 Specification [31]

Table 33. *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_ACF.1.3 [ECASD]	[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]	none
FDP_ACF.1.4 [ECASD]	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	none
FDP_IFC.1.1 [Platform-Services]	[selection: Reference Enterprise Rule enforcement, no additional operations]	no additional operations
FDP_IFF.1.1 [Platform-Services]	[selection: Reference Enterprise Rule enforcement, no additional operations]	no additional operations
FDP_IFF.1.2 [Platform-Services]	[selection: by S.ISD-R to S.PRE in order to execute the Reference Enterprise Rule enforcement function, no additional information flows]	no additional information flows
FDP_IFF.1.3 [Platform-Services]	[assignment: additional information flow control SFP rules]	no additional information flow control SFP rules
FDP_IFF.1.4 [Platform-Services]	[assignment: rules, based on security attributes, that explicitly authorise information flows]	none
FDP_IFF.1.5 [Platform-Services]	[assignment: rules, based on security attributes, that explicitly deny information flows]	When none of the conditions listed in the element FDP_IFF.1.4 of this component hold and at least one of those listed in the element FDP_IFF.1.2 does not hold.
FPT_FLS.1.1 [Platform-Services]	[assignment: selection: Reference Enterprise Rule enforcement, no additional functions]	no additional functions
	[assignment: other type of failure]	none
FPT_EMS.1.1 [Base]	[assignment: types of emissions]	variations in power consumption or timing during command execution

Table 33. ...continued

SFR ID	Selection / Assignment text	Selection / Assignment value
FMT_MSA.1.1 [RULES]	[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" (SGP.22) • S.ISD-R to delete, via function "ESep.Delete" (SGP.32)].	S.ISD-R to delete, via function "ES10c.DeleteProfile" (SGP.22)
FMT_MSA.1.1 [CERT_KEY]	[selection: ES10c.DeleteProfile (SGP.22), ESep.Delete (SGP.32)]	ES10c.DeleteProfile (SGP.22)
FMT_SMF.1 [Base]	[assignment: list of management functions to be provided by the TSF]	Profile Management functions specified in GSMA SGP.22 [31]
FMT_SMR.1 [Base]	[selection: U.EIM (SGP.32), none]	none
FCS_CKM.2.1 [Mobile_ network]	[assignment: cryptographic key distribution method]	Network Authentication keys as part of the D.PROFILE_NAA_PARAMS loaded during profile installation
	[assignment: list of standards]	[29], [30], [31]
FCS_COP.1.1 [Mobile_ network]	[selection: [assignment: cryptographic algorithms], no other algorithm]	other algorithms (CAVE)
	[selection: [assignment: list of standards], no additional standards]	[44] & [43] for CDMA_CAVE
FCS_CKM.6.1 [Mobile_ network]	[selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]	no longer needed
FCS_CKM.6.2 [Mobile_ network]	[assignment: cryptographic key destruction method]	physically overwriting the keys in a randomized manner
	[assignment: list of standards]	none

7.2.2 Security Requirements Rationale for eUICC

The rationale of Security Functional Requirements for the the eUICC component is strictly the same as in the eUICC PP [12].

7.2.3 Security Requirements Dependencies for eUICC

The Security Functional Requirements dependencies for the eUICC component are strictly the same as in the eUICC PP [12].

8 Security Assurance Requirements (ASE_REQ)

8.1 Security Assurance Requirements

The assurance requirements of this evaluation are EAL4, augmented by AVA_VAN.5, ALC_DVS.2, ALC_FLR.2. The assurance requirements ensure, among others, the security of the TOE during its development and production.

8.2 Rationale for the Security Assurance Requirements

All the Protection Profiles referenced in PP claim Section 2.2 target EAL4 augmented with ALC_DVS.2, and AVA_VAN.5 and also give a rationale for this choice, which is entirely applicable to this Security Target. ALC_FLR.2 rationale is provided in [11] that is also applicable to this Security Target.

8.3 Dependencies of Security Assurance Requirements

All the security assurance requirements selected in Section 8.1 are copied from the Protection Profiles referenced in PP claim Section 2.2 so no additional rationale is needed.

9 TOE summary specification (ASE_TSS)

9.1 Introduction

The Security Functions (SF) and Security Services (SS) introduced in this section realize the SFRs of the TOE. Each SF/SS consists of components spread over several TOE modules to provide a security functionality and fulfill SFRs.

9.2 Security Functionality of Java Card System

The Security Functions (SF) introduced in this section realize the SFRs of the TOE. See Table 34 for list of all Security Functions. Each SF consists of components spread over several TOE modules to provide a security functionality and fulfill SFRs.

Table 34. Overview of Security Functionality

Name	Title
SF.JCVM	Java Card Virtual Machine
SF.CONFIG	Configuration Management
SF.OPEN	Card Content Management
SF.CRYPTO	Cryptographic Functionality
SF.RNG	Random Number Generator
SF.DATA_STORAGE	Secure Data Storage
SF.OSU	Operating System Update
SF.OM	Java Object Management
SF.MM	Memory Management
SF.PIN	PIN Management
SF.PERS_MEM	Persistent Memory Management
SF.EDC	Error Detection Code API
SF.HW_EXC	Hardware Exception Handling
SF.RM	Restricted Mode
SF.PID	Platform Identification
SF.SMG_NSC	No Side-Channel
SF.SENS_RES	Sensitive Result
SF.CONT_SEP	Context Separation

SF.JCVM	**Java Card Virtual Machine** SF.JCVM provides the Java Card Virtual Machine including byte code interpretation and the Java Card Firewall according to the specifications [15], [14].

SF.CONFIG	**Configuration Management** SF.CONFIG provides means to store Initialization Data and Pre-personalization Data before TOE delivery. SF.CONFIG provides means to change configurations of the card. Some configurations can be changed by the customer and some can only be changed by NXP. Additionally, SF.CONFIG provides proprietary commands to select the OS update mechanism SF.OSU and to reset the OS to an initial state.
SF.OPEN	**Card Content Management** SF.OPEN provides the card content management functionality according the GlobalPlatform Specification [18] and GlobalPlatform Amendments A [19], D [21], E [22] and F [23]. In addition to the GP specification, the Java Card Runtime Environment specification [15] is followed for application loading, installation, and deletion. AID management is provided by SF.OPEN according to the GlobalPlatform Specification [18], the Java Card Runtime Environment Specification [15], and the Java Card API Specification [13]. SF.OPEN is part of the TOE runtime environment and thus separated from other applications.
SF.CRYPTO	**Cryptographic Functionality** SF.CRYPTO provides key creation, key management, key deletion and cryptographic functionality. It provides the API in accordance to the Java Card API Specification [13]. Proprietary solutions (e.g., key lengths not supported by the Java Card API) are supported following the Java Card API. SF.CRYPTO uses SF.DATA_STORAGE. This TSF enforces protection of Key material during cryptographic functions processing and Key Generation, against state-of-the-art attacks, including IC power consumption analysis.
SF.RNG	**Random Number Generator** SF.RNG provides secure random number generation. Random numbers are generated by the Security Software certified with the TOE hardware. SF.RNG provides an API according to the Java Card API Specification [13] to generate random numbers.
SF.DATA_STORAGE	**Secure Data Storage** SF.DATA_STORAGE provides a secure data storage for confidential data. It is used to store cryptographic keys and to store PINs. All data stored by SF.DATA_STORAGE is CRC32 integrity protected. The stored data is AES encrypted.
SF.OSU	**Operating System Update** SF.OSU provides secure functionality to update the JCOP 7.x OS or SystemOS itself with an image created by a trusted off-card entity. SF.OSU allows an authenticated OSU command to upload an integrity and confidentiality protected update image to update to another operating system version. User authentication is based on the verification of signed OSU commands. Integrity protection of OSU commands uses ECDSA, SHA-256 and CRC verification. Confidentiality of the update image is ensured by ECDH and AES encryption. SF.OSU ensures that the system stays in a secure state in case of invalid or aborted update procedures and ensures that the information identifying the currently running OS is modified and the updated code is activated only after successfull OS Update procedure.

SF.OM	**Java Object Management** SF.OM provides the object management for Java objects which are processed by SF.JCVM. It provides object creation and garbage collection according to the Java Card Runtime Environment Specification [15]. SF.OM throws an Java Exception in case an object cannot be created as requested due to too less available memory.
SF.MM	**Memory Management** SF.MM provides deletion of memory for transient arrays, global arrays, and logical channels according to the Java Card Runtime Environment Specification [15] by granting access to and erasing of CLEAR_ON_RESET and CLEAR_ON_DESELECT transient arrays, by clearing the APDU buffers for new incoming data, by clearing the bArray during application installation, or by any Global Array after usage.
SF.PIN	**PIN Management** SF.PIN provides secure PIN management by using SF.DATA_STORAGE for PIN objects specified in the Java Card API Specification [13] and the GlobalPlatform Specification [18].
SF.PERS_MEM	**Persistent Memory Management** SF.PERS_MEM provides atomic write operations and transaction management according to the Java Card Runtime Environment Specification [15]. SF.PERS_MEM supports SF.JCVM by halting the system in case of object creation in aborted transactions. Low level write routines to persistent memory in SF.PERS_MEM perform checks for defect memory cells.
SF.EDC	**Error Detection Code API** SF.EDC provides an Java API for user applications to perform integrity checks based on a checksum on Java arrays [45], [51], [57], [63], [69]. The API throws a Java Exception in case the checksum in invalid.
SF.HW_EXC	**Hardware Exception Handling** SF.HW_EXC provides software exception handler to react on unforeseen events captured by the hardware (hardware exceptions). SF.HW_EXC catches the hardware exceptions, to ensure the system goes to a secure state, as well as to increase the attack counter in order to resist physical manipulation and probing.
SF.RM	**Restricted Mode** SF.RM provides a restricted mode that limits the functionality of the TOE. Only the S.ACAdmin is able to reset the Attack Counter to leave the restricted mode. SF.RM only allows a limited set of operations to not identified and not authenticated users when in restricted mode. All other operations require identification and authentication
SF.PID	**Platform Identification** SF.PID provides a platform identifier. For elements that can be identified see Section 1.6.
SF.SMG_NSC	**No Side-Channel** The TSF ensures that during command execution there are no usable variations in power consumption (measurable at e.g. electrical contacts) or timing (measurable at e.g. electrical contacts) that might disclose cryptographic keys or PINs. All functions of SF.CRYPTO except for SHA are resistant to side-channel attacks (e.g. timing attack, SPA, DPA, DFA, EMA, DEMA).

SF.SENS_RES	**Sensitive Result** SF.SENS_RES ensures that sensitive methods of the Java Card API store their results so that callers of these methods can assert their return values. If such a method returns abnormally with an exception then the stored result is tagged as Unassigned and any subsequent assertion of the result will fail.
SF.CONT_SEP	**Context Separation** The product supports several contexts of operation that guaranty code execution, data storage, and hardware virtualization in a completely isolated way from other contexts. For that, the whole address space is organized in regions. Each region is assigned access rights based on Context and Access Type (read/write/execute) thus allowing a complete control of the MPU over the entire memory space . The default context separation will be applied at boot by the SMK and only the SMK can change the context separation setting after that. One specific Context is reserved for The SMK whereas other Contexts are reserved for Guest Operating Systems. The State information and the Context number are applied to all bus transactions for checking access control by the MPU. Any tentative to perform a forbiden access will be prevented and will trigger a security alarm.

9.3 Security Functionality of eUICC

SF.CRYPTO_eUICC	**eUICC specific cryptographic algorithms** This TSF provides key creation, key management, key deletion and cryptographic functionality specific to the eUICC component. It provides the API in accordance to eUICC specification [31]. This TSF also enforces protection of key material during cryptographic functions processing and key Generation, against state-of-the-art attacks, including IC power consumption analysis.
SF.ACCESS_eUICC	**eUICC features access protection** This TSF handles the access to eUICC features by external or local users. It is based on JavaCard and GlobalPlatform features to implement the different flow controls, as well as the conditions realization granting the access: identification/authentication of users and the different trusted channels establishment.
SF.SELF-PROTECTION_eUICC	**eUICC specific self-protections** This TSF extends the scope of self-protections features provided by the JavaCard platform to the eUICC component needs.

10 Bibliography

10 . 1 Evaluation documents

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, CC:2022 Revision 1, November 2022, CCMB-2022-11-001

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, CC:2022 Revision 1, November 2022, CCMB-2022-11-002

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, CC:2022 Revision 1, November 2022, CCMB-2022-11-003

[4] Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, CC:2022 Revision 1, November 2022, CCMB-2022-11-004

[5] Common Criteria for Information Technology Security Evaluation, Part 5: Predefined packages of security requirements, CC:2022 Revision 1, November 2022, CCMB-2022-11-005

[6] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, CEM:2022 Revision 1, November 2022, CCMB-2022-11-006

[7] AIS20: Anwendungshinweise und Interpretationen zum Schema (AIS), Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, Version 3, 15.05.2013, Bundesamt für Sicherheit in der Informationstechnik (BSI),

[8] JIL-ATT-SC: Attack Methods for Smartcards and Similar Devices, Joint Interpretation Library, Version 2.4, January 2020

[9] JIL: Security requirements for post-delivery code loading, Joint Interpretation Library, Version 1.0, February 2016.

[10] 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.

[11] Java card protection profile - open configuration, published by oracle, inc., Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0099-V3-2024, Version 3.2.

[12] GSMA SGP.25 - eUICC for Consumer and IoT Devices Protection Profile, BSI-CC-PP-0100-V2-2025, Version 2.1, 03-February-2025.

10 . 2 Standards

[13] Published by Oracle. Java Card Platform, Application Programming Interface, Classic Edition, Version 3.1, November 2019.

[14] Published by Oracle. Java Card Platform, Virtual Machine Specification, Classic Edition, Version 3.1, November 2019.

[15] Published by Oracle. Java Card Platform, Runtime Environment Specification, Classic Edition, Version 3.1, November 2019.

[16] Gosling, Joy, Steele and Bracha. The Java Language Specification. Third Edition, May 2005. ISBN 0-321-24678-0.

[17] Tim Lindholm, Frank Yellin. The Java Virtual Machine Specification. Lindholm, Yellin. ISBN 0-201-43294-3.

[18] GlobalPlatform Card Specification 2.3.1, GPC_SPE_034, March 2018.

[19] GlobalPlatform Confidential Card Content Management - Amendment A v1.1.1, GPC_SPE_007, September 2018.

[20] GlobalPlatform Remote Application Management over HTTP - Amendment B v1.1.3, GPC_SPE_011, May 2015.

[21] GlobalPlatform Card Technology Secure Channel Protocol ’03’ - Amendment D v1.1.2, GPC_SPE_014, March 2019.

[22] GlobalPlatform Security Upgrade for Card Content Management - Amendment E v1.1, GPC_SPE_042, October 2016.

[23] GlobalPlatform Card Secure Channel Protocol ’11’ - Amendment F v1.2.1, GPC_SPE_093, March 2019.

[24] GlobalPlatform common Implementation Configuration - v2.1, GPC_GUI_080, August 2018.

[25] GlobalPlatform Card Composition Model Security Guidelines for Basic Applications, December 2014.

[26] GlobalPlatform Card UICC Configuration - v2.0, GPC_GUI_010, December 2015.

[27] GlobalPlatform Card Contactless Extension - v2.0, GPC_GUI_035, February 2017.

[28] GlobalPlatform Card API (org.globalplatform), v1.7, July 2019.

[29] GSMA Remote Provisionning Architecture for Embedded UICC Technical Specification - v4.0, SGP.02, February 2019.

[30] GSMA Remote SIM Provisionning (RSP) Architecture - v2.2, SGP.21, September 2017.

[31] GSMA Remote SIM Provisionning (RSP) Technical Specification - v2.3.0, SGP.22, 30 June 2021.

[32] eSIM IoT Technical Specification - v1.0, GSM Association, May 2023.

[33] ETSI. ETSI TS 102 622 v11.0.0 Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI), 9 2011.

[34] ETSI. ETSI TS 102 622 v12.1.0 Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI), 10 2014.

[35] IETF RFC 8032. Edwards-Curve Digital Signature Algorithm (EdDSA).

[36] IETF RFC 7748. Elliptic Curves for Security.

[37] PKCS #1: RSA Cryptography Standard, Version 2.2, October 27, 2012, RSA Laboratories

[38] ANSI X9.62-2005: Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA), American National Standards Institute (ANSI), 2005.

[39] NIST SP 800-38C: Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, July 2007, Morris Dworkin, National Institute of Standards and Technology

[40] NIST SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/ Counter Mode (GCM) and GMAC, November 2007, Morris Dworkin, National Institute of Standards and Technology

[41] FIPS PUB 186-4-2013: Digital Signature Standard, Federal Information Processing Standards Publication, 2013, July, National Institute of Standards and Technology

[42] FIPS PUB 197-2001: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/National Institute of Standards and Technology, 26 November 2001.

[43] 3GPP2 S.S0053-0: Common Cryptographic Algorithms, v2.0, May 2009

[44] 3GPP2 C.S0065-B: cdma2000 Application on UICC for Spread Spectrum Systems, v2.0, Jan 2011

10 . 3 Developer documents

[45] NXP. JCOP 7.0 User Guidance Manual, Rev. 1.24.7, date 2025-02-24.
[46] NXP. JCOP 7.0 UGM Addendum, Rev. 1.24.1, date 2024-04-09.
[47] NXP. JCOP 7.0 UGM Anomaly, Rev. 1.24.1, date 2024-04-09.
[48] NXP. JCOP 7.0 R1.64.0.2 (17.4-2.64) UGM for JCOP eUICC, Rev. 1.24.5, date 2025-03-03.
[49] NXP. JCOP 7.0 UGM Addendum UICC, Rev. 1.28.1, date 2024-04-09.
[50] NXP. JCOP 7.0 UGM Addendum System Management, Rev. 1.24.2, date 2024-04-29.
[51] NXP. JCOP 7.0 User Guidance Manual, Rev. 2.04.2, date 2025-02-18.
[52] NXP. JCOP 7.0 UGM Addendum, Rev. 2.04.0, date 2024-04-02.
[53] NXP. JCOP 7.0 UGM Anomaly, Rev. 2.04.0, date 2024-04-02.
[54] NXP. JCOP 7.0 R2.04.0.2 (18.4-2.04) UGM for JCOP eUICC, Rev. 2.04.2, date 2025-02-27.
[55] NXP. JCOP 7.0 UGM Addendum UICC, Rev. 2.04.0, date 2024-04-02.
[56] NXP. JCOP 7.0 UGM Addendum System Management, Rev. 2.04.1, date 2024-04-29.
[57] NXP. JCOP 7.1 User Guidance Manual, Rev. 3.05.4, date 2025-02-17.
[58] NXP. JCOP 7.1 UGM Addendum, Rev. 3.04.1, date 2024-04-02.
[59] NXP. JCOP 7.1 UGM Anomaly, Rev. 3.04.1, date 2024-04-02.
[60] NXP. JCOP 7.1 R1.04.0.2 (19.4-2.04) UGM for JCOP eUICC, Rev. 3.05.3, date 2025-02-17.
[61] NXP. JCOP 7.1 UGM Addendum UICC, Rev. 3.04.1, date 2024-04-02.
[62] NXP. JCOP 7.1 UGM Addendum System Management, Rev. 3.04.2, date 2024-04-29.
[63] NXP. JCOP 7.2 User Guidance Manual, Rev. 4.05.3, date 2025-02-18.
[64] NXP. JCOP 7.2 UGM Addendum, Rev. 4.05.0, date 2024-02-28.
[65] NXP. JCOP 7.2 UGM Anomaly, Rev. 4.05.0, date 2024-02-28.
[66] NXP. JCOP 7.2 R1.09.0.2 (20.4-2.06) UGM for JCOP eUICC, Rev. 4.05.3, date 2025-02-18.
[67] NXP. JCOP 7.2 UGM Addendum UICC, Rev. 4.05.0, date 2024-02-28.
[68] NXP. JCOP 7.2 UGM Addendum System Management, Rev. 4.05.0, date 2024-02-28.
[69] NXP. JCOP 7.3 User Guidance Manual, Rev. 5.6.2, date 2025-08-07.
[70] NXP. JCOP 7.3 UGM Addendum, Rev. 5.6.0, date 2025-03-07.
[71] NXP. JCOP 7.3 UGM Anomaly, Rev. 5.6.1, date 2025-04-09.
[72] NXP. JCOP 7.3 R1.07.0.2 (21.4-2.07) UGM for JCOP eUICC, Rev. 5.6.2, date 2025-08-07.
[73] NXP. JCOP 7.3 UGM Addendum UICC, Rev. 5.6.0, date 2025-03-07.
[74] NXP. JCOP 7.3 UGM Addendum System Management, Rev. 5.6.0, date 2025-03-07.
[75] ES_JCOP7.x Documentation Errata, Rev. 1.1, date 2025-08-07 .
[76] NXP SN300 Series - Secure Element Security Target, rev. 1.0.6, 19 February 2025.

11 Legal information

11.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

11.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

11.3 Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 18 August 2025
Document identifier: NXP-ST01-SN300-Jxxxx