NXP JCOP-eSE 8.9 R6 on SN330 Secure Element

Security Target Lite

Rev. 1.1 — 6 August 2025 **Product evaluation document**

NSCIB-2400084-01

Document information

Information	Content
Keywords	NXP, ASE, JCOP-SE 8.9 R6, Single Chip Secure Element and NFC Controller, JCOP, Common Criteria, EAL5 augmented
Abstract	This document is the Security Target Lite of NXP JCOP-eSE 8.9 R6 on SN330 Secure Element, developed and provided by NXP Semiconductors. The TOE complies with Evaluation Assurance Level 5 of the Common Criteria for Information Technology Security Evaluation CC:2022 Revision 1 with Augmentations.

Revision History

Revision history

Revision number	Date	Description
1.0	2025-06-05	Public version of Security Target
1.1	2025-08-06	Correction of Typo in Platform String

1 ST Introduction (ASE_INT)

1.1 ST Reference

"NXP JCOP-eSE 8.9 R6 on SN330 Secure Element", Security Target Lite, Version details as per the Revision History.

1.2 TOE Reference

Table 1. TOE Reference

Content	Version
Product Type	Java Card
TOE name	NXP JCOP-eSE 8.9 R6 on SN330 Secure Element
TOE version(s)	JCOP-eSE 8.9 R6.01.00.1.1

1.3 TOE Overview

1.3.1 Usage and Major Security Features of the TOE

JCOP-SE 8.9 R6 supports concurrent execution of multiple JCOP OS instances out of individual and hardware controlled execution domains, running on the Embedded Secure Element of SN330 Secure Element, a single die Secure microcontroller, which also includes an NFC controller, a dedicated Power Management Unit and IC Specific software services.

The platform core software consists of a Secure Micro-Kernel and a shared Code Subsystem as well as dedicated host OSs for System and Communication management.

- **Secure Micro-Kernel (SMK)**: Creates seperate, independently updateable secondary Guest OS instances controlling the execution domain switching and ensuring the timing requirements, based on pre-assigned priorities, of the JCOP OS are met.
- **Shared Code Subsystem**: Provides common code, such as Crypto Lib, Flash Services, Java Card and GlobalPlatform implementations.
- **SystemManagement OS** (System OS), which provides an interface to update and manage the different JCOP OS instances and the platform core itself. It also provides some Memory management and Error Handling. This is described in the System Management Addendum of the UGM
- **Communication OS**(Comm OS) which serves the communication interfaces and implements communication protocols. It is not part of the Target of Evaluation (TOE) for security certification but it is covered for functional certification.

The platform also hosts secondary Guest OSs which may be application specific. The TOE of this Security Target is such a Guest OS.

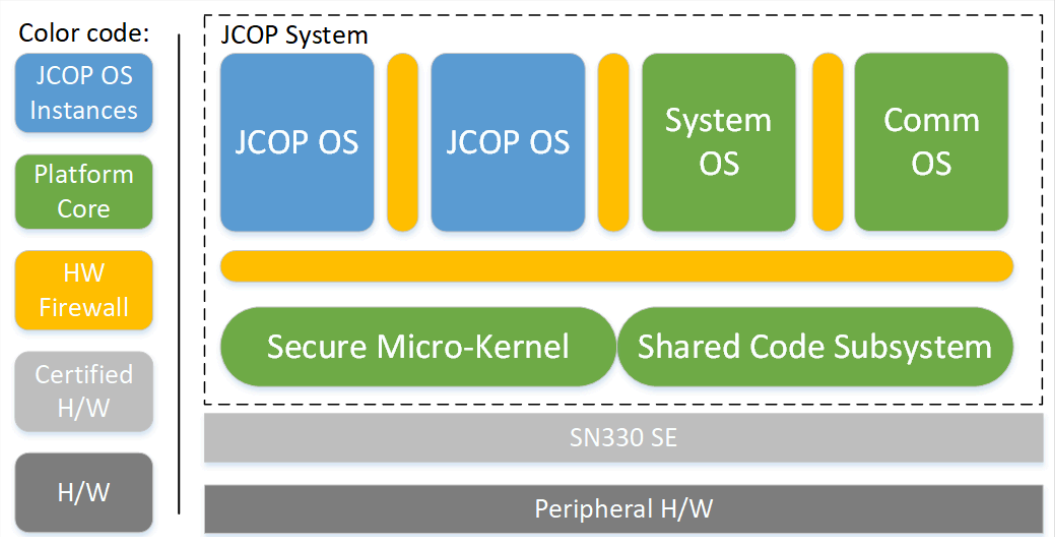

Figure 1. JCOP8.9 System Context

1.3.1.1 JCOP8.9 SE

The JCOP 8.9 Secure Micro-Kernel creates separate and independently updatable secondary Operating Systems (Guest OS).

All Guest OS at the platform level are underpinned by the same Java Card and GlobalPlatform technology, but are dedicated to their own application.

It excludes the NFC Controller, the Power Management Unit and any other present Guest OS.

The component of the SN330 on which the TOE executes is the embedded Secure Element, abbreviated to SN330_SE. Figure 2 provides an overview of the TOE and its place in the overall system.

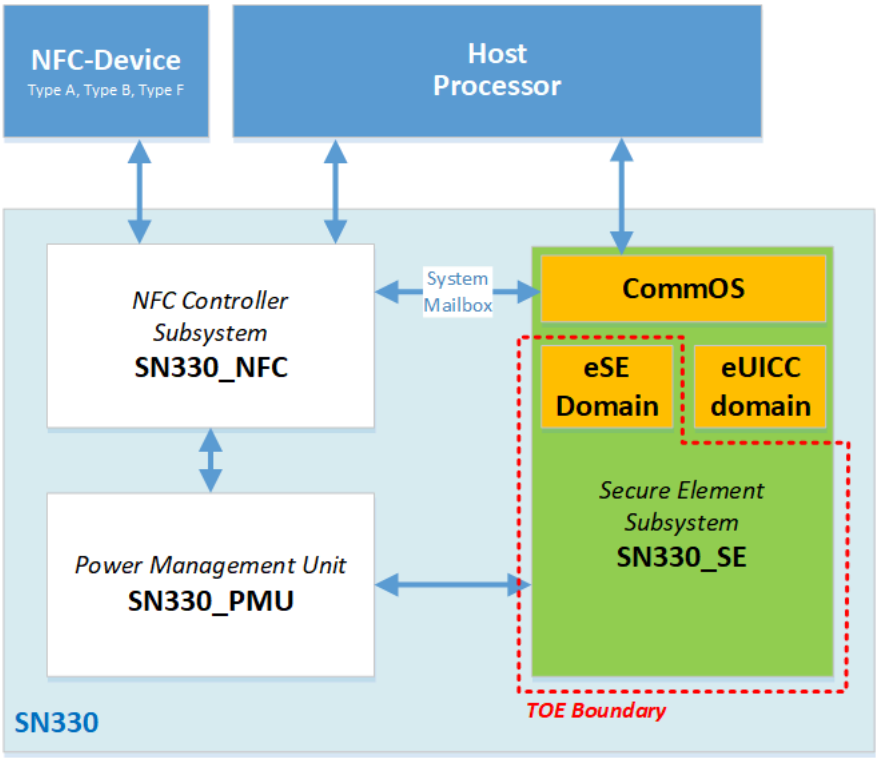

Figure 2. eSE in System context

The TOE communicates, via the CommOS, with the Host Processor through SPI, I2C, I3C and UART interfaces and with the integrated NFC controller through the System Mailbox. The integrated NFC controller is not in scope of this evaluation, however provides up to four gates for external users to communicate with the TOE, supporting Card Emulation Mode Type A, Type B and Type F as well as a wired Interface using APDUCard over UART or SPMI Gate.

The usage of the TOE is focused on security critical applications in small form factors. One main usage scenario is the use in mobile phones, which can use the TOE to enable mobile payment or mobile ticketing with the phone based on the security of the TOE.

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.

1.3.1.2 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 SN330 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

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

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, EMFI detector, 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. And the lockstep (redundant) CPU ensures protection against faults in the CPU.

1.3.1.3 Cryptographic algorithms and functionality:

- AES
- Triple-DES (3DES)
- RSA Functions
- ECDSA Functions
- ECDH Functions
- ECC Functions
- 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 Functions
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 algorithms
- HMAC algorithms

- 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 [9] and initialized (seeded) by the hardware random number generator of the TOE.

1.3.1.4 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 [17] are not implemented. A call to those APIs throw an exception of type ISO7816.SW_FUNC_NOT_SUPPORTED in this case.

1.3.1.5 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.
- Delegated Management, DAP (RSA 1024 and ECC 256).

Supported GlobalPlatform Amendments:

- Amd A : Confidential Card Management
- Amd C : Contactless Services
- Amd D : Secure Channel Protocol '03'
- Amd E : Security Upgrade for Card Management
- Amd F : Secure Channel Protocol '11'
- Amd H : Executable Load File Upgrade
- Amd I : Secure Element Management Service (SEMS)

1.3.1.6 Additional standard functionality

- Cryptographic Service Provider feature [43]

1.3.1.7 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 OS, Crypto Lib, Flash Services Software or Updater OS. 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.
- Error Detection Code (EDC) API.

1.3.1.8 Functionality without specific security claims

- eUICC features hosted in eUICC domain outside the boundaries of the TOE
- Programmable Timeout for SMB with Limitations in UGM [53] Section 6
- CPLC data made available through SystemInfo, see UGM [53] Section 1.3.3.
- NXP Proprietary ByteCode Compression – Applets installed Pre-Issuance by NXP may make use of optimised bytecodes
- Compliance to Secure Element configuration, Common Implementation Configuration, UICC Configuration, and UICC Configuration Contactless Extension.
- MIFARE is subject of separate MIFARE certification scheme
- Felica Lib is subject of separate Felica certification scheme

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 eSE Java Card Operating System and the SN330 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 eUICC or CommOS) that are considered as domains external to the TOE.

The eSE Java Card Operation System includes GP functionality. It can be used to load, install, instantiate and execute off-card verified Java Card applets. The eSE, which is externally accessible via SPI or by the System mailbox connected to the Integrated NFC controller, supports Type A,B and F contactless communications.

1.3.3 Required non-TOE Hardware/Software/Firmware

As a subsystem of a physical chip, the TOE needs the other subsystems of the chip (the Power Management Unit and the NFC controler) to behave properly and communicate with the external world.

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

1. **End-users** group, which uses the TOE 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 eSE domain of the TOE communicates via the Secure Mail Box, which is connected to the Integrated NFC controller and via SPI direct interface. 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

JCOP-SE 8.9 R6 consists of the following components that are part of the TOE:

- SN330 Secure Element excluding NFC (see Section 1.4.1)
- Secure Micro-Kernel (see Section 1.4.2)
- SystemOS (see Section 1.4.4)
- Shared Code
  - Crypto Library (see Section 1.4.3.1)
  - FlashOS (see Section 1.4.3.2)
  - Other Shared Code (see Section 1.4.3)
- JCOP-eSE 8.9 (see Section 1.4.5)

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

- NFC Controller Subsystem
- Power Management Unit
- JCOP eUICC
- JCOP-xxx (optional)
- CommOS

All components and the TOE boundaries are depicted in Figure 3. The components are described in more detail in the following sections.

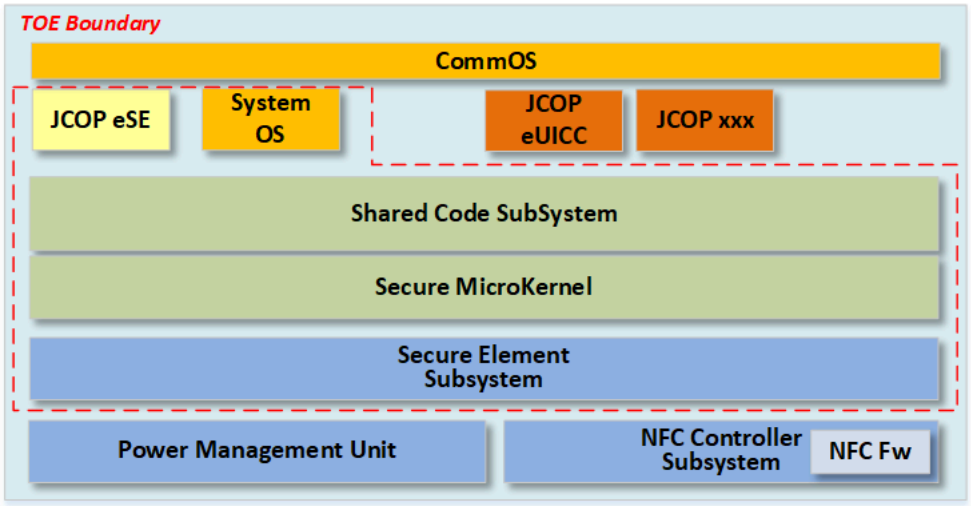

Figure 3. Components of the TOE

The certification of this TOE is a composite certification[12]. 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 3 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 SN330 Single Chip Secured (NFC) Controller is a hardware platform designed to meet the developing needs of the mobile communications market. It embeds a Secure Element Subsystem (SN330_SE), supported by an integrated NFC Controller Subsystem (SN330_NFC) and Power Management Unit (SN330_PMU).

The hardware part of the SN330_SE is referred to as Secure Element Hardware in the following.

The SN330 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	SN330 Series
Certified HW version	SN330_SE A0
Certification ID	NSCIB-CC-2400072-01-MA
Security Target Reference	NXP SN330 Series - Secure Element Security Target, v1.3 10-Jan-2025 [60]

1.4.1.1

1.4.1.2 Hardware Description

The separation of operating systems is based on the ARM Trustzone-M concept. One Main OS 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 SN330 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 buses, 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 SN330 Secure Element implements a wide range of hardware components. It embeds the Fast Accelerator for Modular Exponentiation(FAMEv3.5), 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[1].

SN330 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.

1 Note: FIPS 140-3 compliance is not in scope of this Common Criteria evaluation

The SN330 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 with EMFI detector integrated 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 ("SN330_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.

1.4.1.3 IC Dedicated Support Software

The IC Dedicated Support Software of the SN330_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 SN330 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 SN330 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 Secure Micro-Kernel

The Secure Micro-Kernel (SMK) is responsible for creating and scheduling multiple Guest OS instances. The arrangement of operating systems is determined by a static system configuration passed to the SMK which has the following responsibilities:

- 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,...

1.4.3 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 OS comprises the following components:

- Common Native Code (JCOP and Crypto Library Section 1.4.3.1)
- Flash OS Section 1.4.3.2
- Common Java Card implementation
- Common GlobalPlatform implementation
- Common JCOPX implementation

1.4.3.1 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, 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 and EMSA-PKCS1-v1_5.
- 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 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 ECC over GF(p) from 128 to 640 bits.

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 [10], 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.

1.4.3.2 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 key component providing System update and configuration functionality. Besides the capability to update JCOP OS, System OS is also capable of updating itself.

System OS features are:

- OS Update (Section 1.4.4.1)
- Error Handling (Section 1.4.4.2)
- Restricted Mode Management (Section 1.4.4.3)
- Configuration Interface (Section 1.4.4.4)

1.4.4.1 OS Update

SystemOS contains two sub-components dedicated to supporting OS updates:

- OS Selector (no security claimed): After a hardware reset it provides the functionality to either boot Updater OS or JCOP OS. OS Selector also ensures that
  - only one OS is active (running) at a time.
  - at any time, at least one OS can be booted.
  - an invalid OS (e.g. partly flashed) can never be booted.
- OS updater
  - it handles APDUs to write a new OS (either JCOP OS or Updater OS) 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 Error Handling

The platform implements advanced Error Handling features

1.4.4.3 Restricted Mode

System OS supports the Attack Counter management

1.4.4.4 Configuration Interface

System OS provides an interface to Runtime Configuration features.

1.4.5 JCOP-eSE 8.9

JCOP-eSE presents a familar Java Card Secure Element System with it's own dedicated GlobalPlatform card compliant content management system. The JCOP-eSE OS consists of Shared Code, JCVM, JCRE, JCAPI, Extension API, GP framework, Config Applet and UAI (Update Authorized Image). JCVM, JCRE, JCAPI and GP framework are implemented according to the Java Card Specification listed in Table 3 and the applicable GlobalPlatform Specification and Amendments are listed in Table 4.

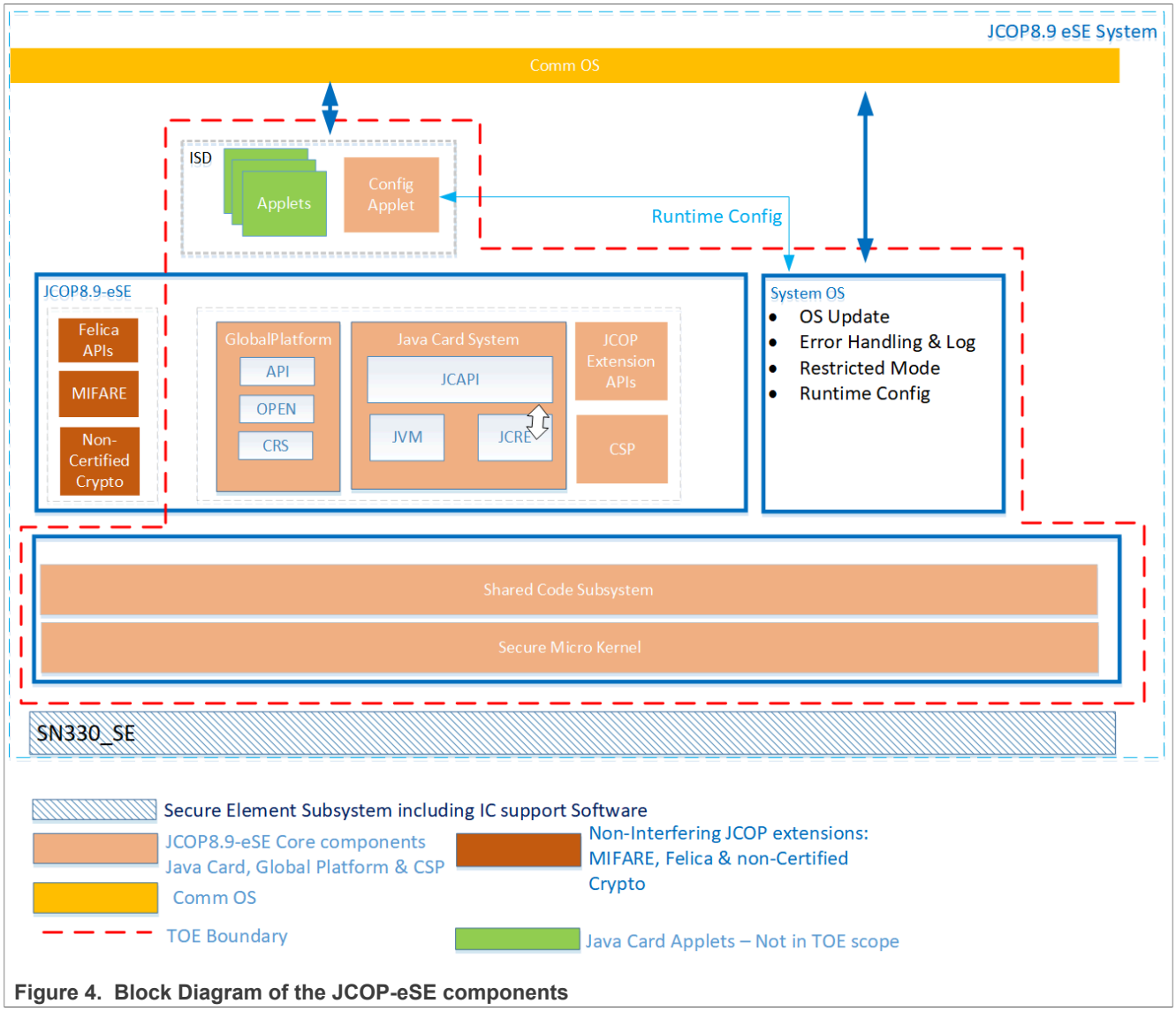

Figure 4. Block Diagram of the JCOP-eSE components

Table 3. Java Card Specification Version

Name	Version
JCVM and JCRE version	Version 3.1 Classic Edition [18] [19]
JC API version	Version 3.1 Classic Edition [17]

Table 4. GlobalPlatform and Amendments

Name	Version
GP Framework	Version 2.3.1 [21]
Amendment A, Confidential Card Content Management	Version 1.2 [22]
Amendment C, Contactless Services	Version 1.3 [23]
Amendment D, Secure Channel Protocol 03	Version 1.2 [24]

Table 4. GlobalPlatform and Amendments*...continued*

Name	Version
Amendment E, Security Upgrade for Card Content Management	Version 1.1 [25]
Amendment F, Secure Channel Protocol 11	Version 1.2.1 [26]
Amendment H, Executable Load File Upgrade	Version 1.1 [27]
Amendment I, Secure Element Management Service	Version 1.0 [28]
Secure Element Configuration	Version 2.0 [31]
Common Implementation Configuration	Version 2.1 [30]
GP Card API	Version 1.7 [32]

JCOP8.9 eSE 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 [53]). The Platform Identification data, which includes the Hardware Type, JCOP Version, Build Number, Mask ID and Non-Volatile Memory Size, identifies the JCOP8.9 platform (combination of HW and SW). The Platform Release is a data string that identifies the eSE OS component. Table 3 in Section 1.6 lists all possible values for the Platform String that are valid for this TOE.

1.4.5.1 Card Content Management

Applet migration can be performed using GlobalPlatform Amd-H.

Card Content Management and Applet Migration can be combined in a sequence of commands which are distributed to the secure elements, this is called Distributed Card Content Management.

1.4.5.2 Update Authorized Image (UAI)

Update Authorized Image (UAI) is supported by the eSE domain and enables an OS Update to be controlled from the eSE domain.

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

1.4.5.3 MIFARE

JCOP8.9 provides Java Card APIs providing specific support for MIFARE standards with access to MIFARE dedicated accelerators. The complete implementation of the MIFARE in JCOP8.9 relies upon applets using these accelerated APIs. JCOP8.9 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.

The MIFARE version is directly identifiable via the version Query command, and may be updated without impacting the Platform Identification.

1.4.5.4 FELICA

JCOP8.9 provides Java Card APIs providing specific support for FELICA standards with access to FELICA specific Cryptography. The complete implementation of the Felica standards in JCOP8.9 relies upon applets using the provided Crypto. JCOP8.9 receives, processes and routes commands from the NFC controller according to the pipe used, with FELICA coming through as raw Type-F commands requiring JCOP to decode, process and route correctly.

The Felica Cryptographic algorithms are embedded in the Crypto Library and therefore any updates to the Felica algorithms will result in updated Crypto Library - as reported in the SystemOS Version Query command, see System OS UGM [54].

Table 5. Felica Lib Versions

JCOP-eSE 8.9	Felica Lib Version
R6.01.00.1.1	6.1.5

1.4.5.5 CSP component details

The CSP component is a Java Card package extension exposing a Java Card CSP API to other Java Card applications.

It implements a platform architecture defined in the CSP PP i.e. users are other applications running on top of the JCOP platform. The JCOP platform provides the required secure execution environment while the CSP JavaCard package provides the secure services implementation.

Table 6. CSP Application Identification

Registered AID	E804007F00070308
Version	0.2

1.4.5.6 Non Certified Crypto

JCOP8.9 provides Java Card APIs for non certified Crypto Algorithms as well as those claimed in the Security Target.

The Algorithms related to SM2, SM3 and SM4 are grouped in an Crypto Library which is independently identifiable and may be updated without affecting the Platform Identifier.

1.4.6 Interfaces of the TOE

Electrical interface

The electrical interface of the TOE are the lines between the I/O interface of the SN330_SE and the communication pads, that are exclusively used by the SN330_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 SN330_PMU subsystem, which connects power supply voltage input and ground as reference voltage, and an interface to the Power-Clock-Reset Module of the SN330_NFC subsystem. Communication between SN330_SE and SN330_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 SN330_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 [14], see Figure 5. Authentic delivery of the TOE is supported by the NXP Trust provisioning Service.

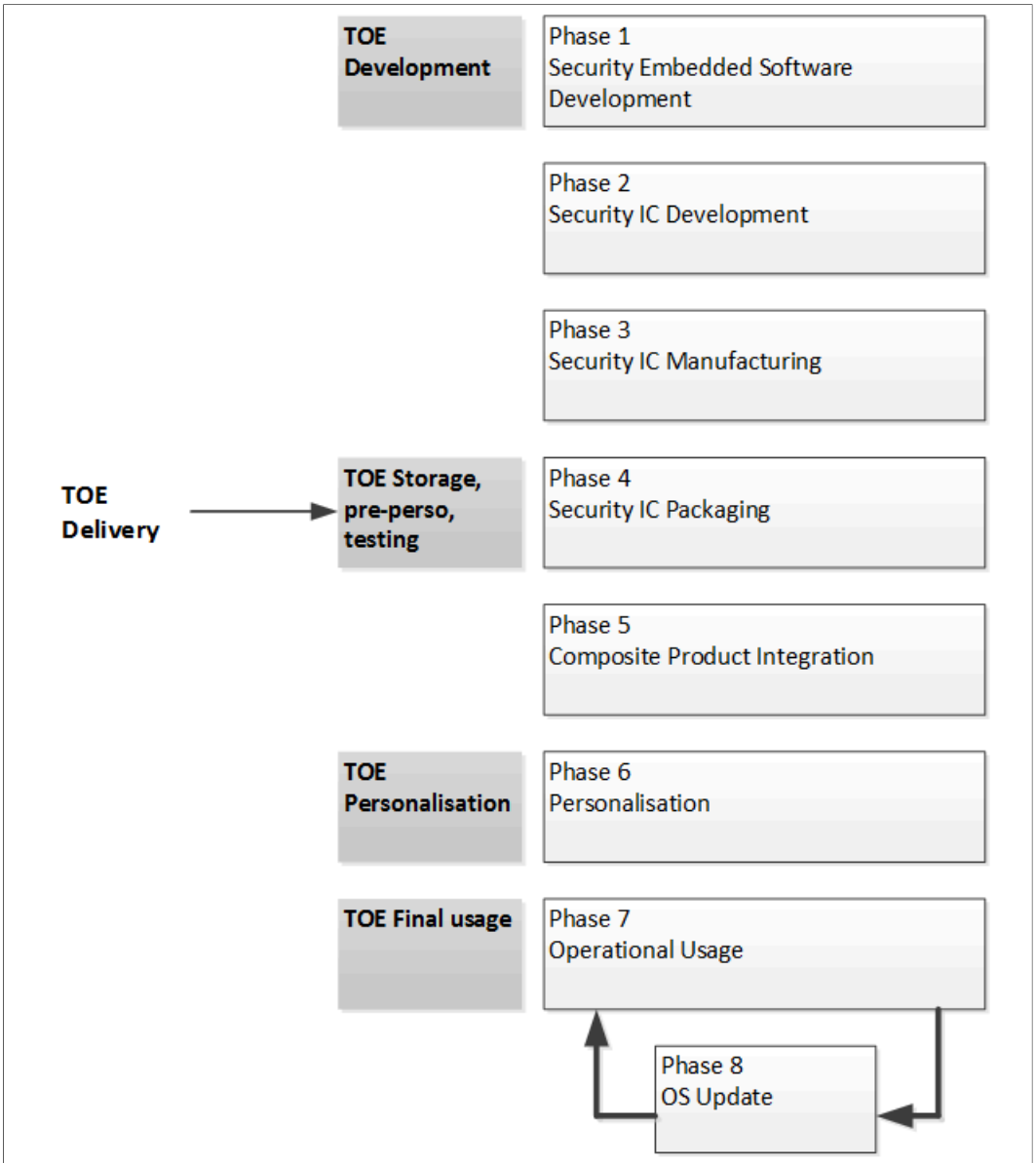

Figure 5. TOE Life Cycle within Product Life Cycle

Table 7.

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 7. *...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.
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 GlobalPlatform 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 1. 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 CSP specific life-cycle

The CSP life-cycle follows the JCOP life cycle in compliance with the CSP PP [15]

1.6 TOE Identification

The TOE, JCOP-eSE 8.9 R6.01.00.1.1, is integrated in a Secure Element product,JCOP-SE 8.9. The TOE may appear on more than one variant of JCOP-SE Product. Current JCOP-SE [2] products integrating the TOE are described in Table 8.

Table 8. JCOP-SE 8.9 Platform Identifier

JCOP-SE	Platform Identifier	Integrated TOE:
JCOP-SE 8.9	N5F0000000030600	**JCOP-eSE 8.9 R6.01**

The Platform ID (PID) Section 1.6.1 and the Platform Release String Section 1.6.2 can be obtained by using the Version Query command (GET DATA command with tag 0xDF4C). See Section 1.4.5 of UGM [53], .

1.6.1 Platform Identifier

JCOP-SE Platform is composed on a specific Hardware Secure Element and is identified by the Platform ID Table 8.

JCOP-SE 8.9 is a multi-OS system, individual JCOP instances are uniquely identified by their Platform String.Table 10. The TOE of this ST is such a JCOP instance.

The Platform Identifier provides an SE family identifier, i.e. JCOP-SE 8.9, and is unaffected by the application of patches to any of the components of that JCOP-SE.

The Platform ID (PID) has the format: **Nabcccxxxxxxyyzz**

2 It is important to note that this certified TOE configuration could appear on future JCOP-SE variants where other components are changed.

The **"N"** is constant, the other letters are variables. For a detailed description of these variables, please see Table 9.

Table 9. Platform ID Format

Variable	Meaning	Value	Comment
a	Hardware Type	5	NFC hardware
b	JCOP OS Version	F	JCOP8.9
ccc	RFU	000	-
xxxxxx	Build Identifier	000003	-
yy	Mask ID	06	-
zz	RFU	00	-

1.6.2 Platform Release String

The Platform Release String uniquely identifes a JCOP instance and should be relied on as the TOE identification.

Table 10. Platform String

TOE Version	Platform String
JCOP-eSE 8.9 R6.01.00.1.1 [1]	0809 60100011

[1] includes CryptoLib v1.3.0 and FlashOS 134.58.1

Table 11. Platform String Format for JCOP8.9

Variable	Meaning	Value	Parameter Settings
w	JCOP Major version	08	JCOP8.9
x	JCOP Minor version	09	
y	JCOP OS Major release	60	R6.01
zz	JCOP OS Minor release	10	
v	Variant identifier	00	Unpatched
c	JCOP instance	11	eSE

1.6.3 IC Identifier

When the System 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 12. Hardware ID Data Format

Tag	Length	Value (MSB only)	Comment
0x8C	2	0x5F [1]	SN330

[1] LSB is RFU, only MSB defines the HW ID

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 SN330 (including dedicated Flash content).

1.6.4 Current Sequence Number (CSN)

JCOP-SE 8.9 R6 is furnished with a unique sequence number related to the overall JCOP-SE Release, this is used in particular by the OS update mechanism:

1. If Updater OS is active then the "SELECT OS Update AID" command will return the Current Sequence Number of Updater OS 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 OS (Final Sequence Number).

It is important to note that any changes to the JCOP-SE will result in an updated CSN, but that does not necessarily indicate that the TOE has been changed. Any TOE changes are reflected in the Platform Release String.

1.6.5 TOE Delivery Items

The delivery comprises the following items:

Table 13. Delivery items for JCOP8.9

Type	Name	Identification	Delivery form
IC Hardware	NXP SN330 Secure Element	SN330 (see Table 12 and UGM below.	Package WLCSP
Embedded Software	JCOP-SE 8.9 integrating JCOP-eSE 8.9 including the Platform Core software (SMK, Shared code subsystem, System OS and Communications OS and any other Guest OS as well as pre-loaded Applet packages	See Section 1.6 and UGM below	On-chip software stored into the FLASH area of the TOE.

1.6.5.1 JCOP-eSE 8.9 R6 Specific Delivery Items

Table 14. Delivery items specific to JCOP-eSE 8.9 R6.01.00.1.1

Type	Generic Document Name	Identification	Delivery form
Document	JCOP User Guidance Manual (UGM)	[53]	Electronic Document (PDF via NXP Docstore)
Document	JCOP UGM System Management Addendum	[54]	Electronic Document (PDF via NXP Docstore)
Document	JCOP-eSE User Guidance Manual	[55]	Electronic Document (PDF via NXP Docstore)
Document	JCOP-eSE UGM Addendum	Rev 1.8.0, 2025-05-08 [56]	Electronic Document (PDF via NXP Docstore)
Document	JCOP UGM Addendum for CSP	[57]	Electronic Document (PDF via NXP Docstore)
Document	JCOP UGM Addendum - Amd I SEMS Application	[58]	Electronic Document (PDF via NXP Docstore)

Table 14. Delivery items specific to JCOP-eSE 8.9 R6.01.00.1.1*...continued*

Type	Generic Document Name	Identification	Delivery form
Document	Errata Sheet	[59]	Electronic Document (PDF via NXP Docstore)

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. Hence the security functionality of the TOE is not affected by the package type supplied.

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 2022 Conformance Claim", "PP Claim", and "Conformance Claim Rationale".

2.1 CC 2022 Conformance Claim

This Security Target claims conformance to the Common Criteria version 2022, revision 1.

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 2022, Revision 1, November 2022. CCMB-2022-011-001. [3].
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 2022, Revision 1, November 2022. CCMB-2022-011-002. [4].
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 2022, Revision 1, November 2022. CCMB-2022-011-003. [5].
- Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, Version 2022, Revision 1, November 2022. CCMB-2022-011-004. [6].
- Common Criteria for Information Technology Security Evaluation, Part 5: Pre-Defined packages of security requirements, Version 2022, Revision 1, November 2022. CCMB-2022-011-005. [7].

The following methodology will be used for the evaluation:

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 2022, Revision 1, November 2022. CCMB-2022-011-006. [8].

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. Section 7 of this Security Target defines the extended Security Functional Requirements, and also demonstrates that they are consistent with the above conformance claims.

This Security Target claims conformance to the assurance package **EAL5 augmented**. The augmentations to EAL5 are:

- AVA_VAN.5 “Advanced methodical vulnerability analysis”

- ALC_DVS.2 “Sufficiency of security measures”
- ALC_FLR.2 “Flaw reporting Procedures”
- ASE_TSS.2 “TOE summary specification with architectural design summary”

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

This Security Target claims demonstrable conformance to the Java Card Protection Profile - Open Configuration [14].

This Security Target claims package-conformance to the following augmentation packages of the Java Card Protection Profile - Open Configuration [14].

- SENSITIVE RESULT
- MONOTONIC COUNTERS
- CRYPTOGRAPHIC CERTIFICATE MANAGEMENT
- KEY DERIVATION FUNCTIONS (KDF)
- SYSTEM TIME

Other Packages of the PP are not claimed.

This ST is more restrictive than the PP [14] which Conformance Claim Rationale provides a rationale for.

2.2.2 Cryptographic Service Provider Protection Profile (BSI-CC-PP-0104)

The Security Target claims also strict conformance to the Cryptographic Service Provider Protection Profile [15], however relies upon the CCRA tranistion policy from CC v3.1 to CC:2022 to allow the replacement of SFRs updated by CC:2022, which are either exactly equivalent or more restrictive than those defined by the CSP PP.

2.3 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 [14] , implementing the Java Card Specification Version 3.1 [18] [19] [17] with a co-existing eUICC Application, also underpinned by the Java Card and GlobalPlatform Technologies, but accessible via separate, independent communications channels.

2.4 Conformance Claim Rationale for Java Card Component

2.4.1 SPD Statement for Java Card Component

2.4.1.1 Threats

The SPD statement that is presented in Section 4 includes the threats as presented in the PP [14], 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 [13]

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 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 [14].

The threat T.COM_EXPLOIT is included to cover communication channels attacks and it is an addition to the threats in the PP [14].

The threat T.LIFE_CYCLE is included to cover content management attacks and it is an addition to the threats in the PP [14].

The threat T.UNAUTHORIZED_CARD_MNGT refines the threats T.INSTALL and T.DELETION from the PP [14].

The threat T.INTEG-APPLI-DATA[REFINED] refines the threat T.INTEG-APPLI-DATA in the PP [14].

The threat T.RESTRICTED-MODE is included for the Restricted Mode 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 embedded on the same product inside and outside the TOE boundaries.

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 [14] makes the use of Java Card RMI optional.

2.4.1.2 Organisational Security Policies

The SPD statement presented in Section 4, copies the OSP from the PP [14], 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 [14].

The OSP OSP.KEY-CHANGE is introduced for the SD feature of the TOE and is an addition to the OSPs in PP [14].

The OSP OSP.SECURITY-DOMAINS is introduced for the SD feature of the TOE and is an addition to the OSPs in PP [14].

The SPD statement includes two of the three assumptions from the PP [14]. 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 [14]. 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 [14], the following assumptions are added:

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

2.4.1.3 Assumptions

The assumption A.PROCESS-SEC-IC is taken from the underlying Security IC Platform PP [13].

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 [14] 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.

2.4.2 Security Objectives Statement for Java Card Component

The statement of security objectives in the ST presented in Section 5 includes all mandatory security objectives as presented in the PP [14], as well as those related to the optional packages implemented by the TOE and a number of additional security objectives.

The Security Objectives related to the optional packages are mapped in the table below, using the rational defined in the PP[14] for each package.

Table 15. Security Objectives for Optional Packages

Optional Package	Additional Security Objectives
SENSITIVE RESULT	OT.SENSITIVE_RESULTS_INTEG
MONOTONIC COUNTER	OT.MTC-CTR-MGT
CRYPTOGRAPHIC CERTIFICATE MANAGEMENT	OT.CRT-MNGT
KEY DERIVATION FUNCTION (KDF)	No additional objectives
SYSTEM TIME	No additional 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.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 [14]. Additionally, some of the Security Objectives for the Operational Environment from [14] 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 [14] to which conformance is claimed.

The security objectives OT.INSTALL, OT.LOAD, and OT.DELETION from the PP [14] are not included since these functionality and objectives are covered by the refined OT.CARD-MANAGEMENT.

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 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 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 [13].

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 [14].

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 [14] to which conformance is claimed.

2.4.3 SFRs Statement for Java Card Component

The Security Functional Requirements Statement copies most SFRs as defined in the PP [14], 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 [14]. Moreover as requested by the PP [14] 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 [14] 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 [14] 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 [14] 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 [14].

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 [14] by following the information flow policy defined by GlobalPlatform specifications. The table below lists the SFRs from CarG of PP [14] and their corresponding refinements in this ST.

Table 16. CarG SFRs refinements

SFR from PP [14]	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 [14] and add functionality to the TOE making the Security Functional Requirements Statement more restrictive than the PP [14]:

This set of SFRs realize additional security functionality for the card manager, which is allowed by the PP [14].

- FDP_ROL.1/CCM
- FPT_FLS.1/CCM
- FPT_PHP.3

The set of SFRs that define the security domains mechanism as specified by GlobalPlatform, realize refinements of SFRs from PP [14] (see above Table 16) and additional security functionality which is allowed by the PP [14]. 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 [14] (see above Table 16) and additional security functionality which is allowed by the PP [14]. 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

As this Security Target claims some of the optional packages allowed by the PP, the following SFRs are also included:

- FDP_SDI.2/RESULT
- FDP_SDI.2/MONOTONIC_COUNTER
- FDP_SDI.2/CRT_MNGT
- FCS_COP.1/CRT_MNGT
- FCS_CK.5/KDF
- FPT_STM.1/SYS_TIME

The SFRs FAU_SAS.1/SCP, FIA_AFL.1/PIN and FCS_RNG.1 realize additional security functionality which is allowed by the PP [14].

The set of SFRs that define the Config Applet realize additional security functionality, which is allowed by the PP [14]. 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 [14]. 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
- FPT_FLS.1/OSU

The set of SFRs that define the Restricted Mode realize additional security functionality, which is allowed by the PP [14]. 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
- FIA_UAU.1/RM

The set of SFRs that define the Context Separation realize additional security functionality, which is allowed by the PP [14]. This set of SFRs comprise:

- FDP_ACC.2/CONTSEP
- FDP_ACF.1/CONTSEP
- FMT_MSA.1/CONTSEP
- FMT_MSA.3/CONTSEP
- FMT_SMF.1/CONTSEP
- FMT_SMR.1/CONTSEP
- FIA_UID.1/CONTSEP

2.5 Conformance Claim Rationale for CSP component

2.5.1 SPD Statement for CSP Component

The Security Problem Definition of the CSP component is the same as in CSP PP [15], no item have been added, removed or modified.

2.5.2 Security Objectives Statement for CSP Component

The Security Objectives for the TOE and its environment of the CSP component is the same as in the CSP PP [15] with following exclusions due to the overlap with the JCOP objectives defined in [14]:

- OE.SecComm from CSP PP [15] 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 [14].

2.5.3 Security Functional Requirements Statement for CSP Component

The Security Functional Requirements for the CSP component are those of the CSP PP [15] with updates in accordance with the CCRA transition Policy[2] for CC:2022 and some exclusions due to the overlap with the JCOP PP [14].

Some of the extended components defined by the CSP PP are defined by CC:2022

- FCS_RNG.1 is defined by CC:2022 and is common to JCOP
- FCS_CKM.5 is defined by CC:2022
- FIA_API.1 is defined by CC:2022
- FPT_TCT.1 uses CSP definition
- FPT_TIT.1 uses CSP definition
- FPT_ISA.1 uses CSP definition
- FPT_ESA.1 uses CSP definition
- FDT_SDC.1 is defined by CC:2022

Some of the SFRs are common to the JCOP SFRs.

- FCS_CKM.4 replaced by FCS_CKM.6 for CC:2022 transition and is common to JCOP
- FPT_PHP.3 is common to JCOP

The confidentiality of stored data by encryption mechanism is handled at the hardware level as described in [60]; SFRs FDP_SDC.1 and related SFRs FCS_CKM.1/SDEK and FCS_COP.1/SDE are then also excluded due to this overlap.

In SFR FTP_TST.1, the requirements that a test suite has to be run "at the request of the authorised user" is not implemented by the issuance of a dedicated command; however, at the execution of any command invoked by users, regular integrity checks by are performed by the underlying Java Card platform and hardware platform (memory integrity verification, control flow, etc.).

The following SFRs names are extended with the "/CSP" iterations to identify them as part of the CSP component:

- FIA_AFL.1
- FIA_ATD.1
- FIA_UID.1
- FIA_UAU.1
- FIA_UAU.5
- FIA_UAU.6
- FIA_USB.1
- FMT_MOF.1
- FMT_SMF.1
- FMT_SMR.1
- FMT_MSA.2
- FMT_MTD.3
- FMT_SAE.1
- FPT_FLS.1
- FTP_ITC.1
- FPT_TST.1
- FRU_FLT.2

In remaining SFRs, any item related to the following mechanisms are not supported (as authorized by the CSP PP [15]):

- Clustering
- Time service
- Time stamps
- Audit

Concerning the SFRs dependency, only the following differences exist:

- FCS_COP.1/VDSUCP: the import of UCP signature verification key is done during manufacturing.
- FCS_COP.1.1/DecUCP: the import of UCP decryption key is done during manufacturing.

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 [14].

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 ([20]) 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 ([20]) 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.

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 [13]), 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 Updater OS allows to update JCOP OS, the Crypto Lib and the Flash services software as well as the Updater OS 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 eUICC, 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 Main JCOP.

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 [14] .

4.1.1.1 User data

Table 17. User Data Assets

D.APP_CODE	The code of the applets and libraries loaded on the card. To be protected from unauthorized modification.

Table 17. User Data Assets*...continued*

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 [14]. 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 [14]. Issuer Security Domain cryptographic keys needed to perform card management operations on the card. To be protected from unauthorized disclosure and modification.
D.VASD_KEYS	Refinement of D.APP_KEYS of [14]. 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 18. 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.

Table 18. TSF Data Assets*...continued*

D.UPDATE_IMAGE	Can be an update for JCOP8.9 OS and Updater OS. It is sent to the TOE, received by the Updater OS. 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 [11].
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 [14].

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 [14].
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 [14].

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 Main JCOP 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 [14].

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 [14].

Note that the assumption A.DELETION as defined in PP [14] 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” ([18]) 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.
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 CSP

The Securiity Problem Definition for the CSP component of the TOE is strictly compliant with the Security Problem Definition described in the CSP PP [15].

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_VIEWS_ 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_VIEWS_ 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 [14].

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 [14] 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 [14]. 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 [14] 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 [14]. 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 [14] 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 [14] 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 [14] 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" [11].

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 Main JCOP, 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 Main JCOP shall be able to define and modify the priorities.

5.1.2 Security Objectives for CSP

The Security Objectives for the CSP component of the TOE is strictly compliant with the Security Objectives for the TOE described in the CSP PP [15].

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. 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. An external CommOS forwards the APDU correctly from the I/O interface to the configured destination
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 CSP

The Security Objectives for the Environment of the CSP component of the TOE is strictly compliant with the Security Objectives for the Environment described in the CSP PP [15].

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 Java Card Protection Profile [14]. 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 [19].
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 update OS 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 Main JCOP.
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 CSP

The rationales of Security Objectives for the CSP component of the TOE and for its Environment are strictly the same in the CSP PP [15].

6 Extended Components Definition (ASE_ECD)

6.1 Extended Components Definition for Java Card System

There are no extended components defined by the Java Card PP [14].

This Security Target further defines extended components FAU_SAS.1 as is allowed.

6.1.1 Audit Data Storage (FAU_SAS)

This section has been taken over from the certified Smartcard IC Platform Protection Profile [13]. 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 6. 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 CSP

The extended components defined in the CSP PP [15]are listed in Table 19. Some are replaced by the SFRs defined by CC:2022 Part2, which are either equivalent or more restrictive than those defined by the CSP PP.

Table 19. Extended components defined in the CSP PP

Name	Title	CC:2022 Definition	Comment
FCS_RNG.1	Generation of random numbers	Yes	Use CC:2022 definition
FCS_CKM.5	Cryptographic key derivation	Yes	Use CC:2022 definition
FIA_API.1	Authentication proof of identity	Yes	Use CC:2022 definition
FPT_TCT.1	Inter-TSF data confidentiality transfer protection	No	Use CSP definition
FPT_TIT.1	TSF data integrity transfer protection	No	Use CSP definition

Table 19. Extended components defined in the CSP PP*...continued*

Name	Title	CC:2022 Definition	Comment
FPT_ISA.1	TSF data import with security attributes	No	Use CSP definition
FPT_ESA.1	TSF data export with security attributes	No	Use CSP definition
FDP_SDC.1	Stored data confidentiality	Yes	Use CC:2022 definition

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 [14]. 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 2.4.3 around Table 16. Note that this convention only applies to the current chapter.

Table 20. Requirement Groups

Group	Description
Core(CoreG)	The CoreG contains the requirements concerning the runtime environment of the Java Card System that may also implement locgical channels. This includes the firewall policy and teh requirements related to the Java Card API. Logical channels are a Java Card specification version 2.2-3.1, and optional for version 3.2
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.

Table 20. Requirement Groups*...continued*

Group	Description
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 [14].
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 Main JCOP (kernel) 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 21. 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 ([19], §11), but its role asks anyway for a specific treatment from the security viewpoint.
S.APPLET	Any Applet instance
S.BCV	The Bytecode verifier (BCV), which acts on behalf of the verification authority who is in charge of the Byte Code Verification of the CAP files.
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.

Table 21. Java Card Subject Descriptions*...continued*

Subjects	Descriptions
S.LOCAL	Operand stack of a JCVM frame, or local variable of a JCVM frame containing an object or an array of references.
S.MEMBER	Any Object's field, static field or array position
S.CAP_FILE	A CAP file may contain multiple Java language packages. A package is a namespace within the Java programming language that may contain classes and interfaces. A CAP file may contain packages that define either user library, or one or several applets. A CAP file compliant with Java Card Specifications version 3.1 may contain multiple Java language packages. An EXTENDED CAP file as specified in Java Card Specifications version 3.1 may contain only applet packages, only library packages or a combination of library packages. A COMPACT CAP file as specified in Java Card Specifications version 3.1 or CAP files compliant to previous versions of Java Card Specification, MUST contain only a single package representing a library or one or more applets.
S.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.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.
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.MainJCOP	Main JCOP operating system (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 22. 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.

Table 22. Object Groups*...continued*

Objects	Descriptions
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 23. 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.
O.MainJCOP_ Memory_Region	Memory region (addressable memory cells or registers) that is allocated to S.MainJCOP and that is in the boundaries of the TOE.

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

Table 24. 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 25. 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".

Table 25. Security attribute description*...continued*

Security attributes	Description
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.
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 [21].

Table 25. Security attribute description*...continued*

Security attributes	Description
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 [21].
Privileges	Defined in Section 6.6.1 of [21].
Loaded Applet AID	AID of O.APPLET_LOADED.
Current Instance AID	The AID of O.APPLET_CURRENT that is to be updated.
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 [21]
Access Control Table	Security attributes used to define the access control of S.GuestOS and S.MainJCOP to objects O.GuestOS_Memory_Region and O.MainJCOP_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 26. 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.

Table 26. Operation Description*...continued*

Operations	Description
OP.INSTANCE_FIELD (O.JAVAOBJECT, field)	Read/Write a field of an instance of a class in the Java programming language.
OP.INVK_VIRTUAL (O.JAVAOBJECT, method, arg1,...)	Invoke a virtual method (either on a class instance or an array object).
OP.INVK_INTERFACE (O.JAVAOBJECT, method, arg1,...)	Invoke an interface method.
OP.JAVA (...)	Any access in the sense of [19], §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 [19], §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.MainJCOP
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 the Protection Profile [14], which adds further sub-categories:

- Firewall Policy
- Application Programming Interface
- Card Security Management
- AID Management

7.1.1.1.1 Firewall Policy

The Security Functional Requirements for the firewall policy section of the CoreG group are defined in strict compliance with the Security Problem Definition described in [14].

The following table indicates which SFRs are modified by this Security Target:

Table 27. CoreG Firewall SFRs

SFR ID	Modified
FDP_ACC.2/FIREWALL	as per [14]
FDP_ACF.1/FIREWALL	as per [14]
FDP_IFC.1/JCVM	as per [14]
FDP_IFF.1/JCVM	as per [14]
FDP_RIP.1/OBJECTS	as per [14]
FMT_MSA.1/JCRE	as per [14]
FMT_MSA.1/JCVM	as per [14]
FMT_MSA.2/FIREWALL-JCVM	as per [14]
FMT_MSA.3/FIREWALL	as per [14]
FMT_MSA.3/JCVM	as per [14]
FMT_SMF.1	as per [14]
FMT_SMR.1	as per [14]

7.1.1.1.2 Application Programming Interface

The Security Functional Requirements for the Application Programming Interface section of the CoreG group are defined in strict compliance with the Security Problem Definition described in [14].

The following table indicates which SFRs are modified by this Security Target:

Table 28. CoreG API SFRs

SFR ID	Modified
FCS_CKM.1	FCS_CKM.1.1 FCS_CKM.1.1/RSA FCS_CKM.1.1/ECC FCS_CKM.1.1/EdDSA FCS_CKM.1.1/Mont
FCS_CKM.6	FCS_CKM.6

Table 28. CoreG API SFRs*...continued*

SFR ID	Modified
FCS_COP.1	FCS_COP.1.1/GCM FCS_COP.1.1/TripleDES FCS_COP.1.1/AES FCS_COP.1.1/RSACipher FCS_COP.1.1/ECDH_P1363 FCS_COP.1.1/ECDH_25519 FCS_COP.1.1/DESMAC] FCS_COP.1.1/AESMAC] FCS_COP.1.1/RSASigPKCS1 FCS_COP.1.1/ECSignature FCS_COP.1.1/EdDSA] FCS_COP.1.1/SHA FCS_COP.1.1/AES_CMAC FCS_COP.1.1/HMAC FCS_COP.1.1/TDES_CMAC FCS_COP.1.1/DAP
FCS_RNG.1	FCS_RNG.1 FCS_RNG.1.1/HDT
FDP_RIP.1/Abort	as per [14]
FDP_RIP.1/APDU	as per [14]
FDP_RIP.1/GlobalArray	as per [14]
FDP_RIP.1/bArray	as per [14]
FDP_RIP.1/KEYS	as per [14]
FDP_RIP.1/TRANSIENT	as per [14]
FDP_ROL.1/FIREWALL	as per [14]

The following table provides the selection and assignments for the modified SFRs:

Table 29. CoreG API SFR Modifications

SFR ID	Selection / Assignment text	Selection / Assignment value
FCS_CKM.1.1	[assignment: cryptographic key generation algorithm]	JCOP RNG
	[assignment: cryptographic key sizes]	• DES: Key Lengths – LENGTH_DES3_2KEY – LENGTH_DES3_3KEY • AES: Key Lengths – LENGTH_AES_128 – LENGTH_AES_192 – LENGTH_AES_256
	[assignment: list of standards]	FCS_RNG.1 or FCS_RNG.1/HDT
FCS_CKM.1.1/ RSA	[assignment: cryptographic key generation algorithm]	RSA and RSA-CRT key generation Algorithm
	[assignment: cryptographic key sizes]	any length from 512 to 4096 bits

Table 29. CoreG API SFR Modifications*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: list of standards]	[41] and [44]
FCS_CKM.1.1/ ECC	[assignment: cryptographic key generation algorithm]	ECC key generation Algorithm
	[assignment: cryptographic key sizes]	any length from 128 to 528 bits
	[assignment: list of standards]	[42] and [44]
FCS_CKM.1.1/ EdDSA	[assignment: cryptographic key generation algorithm]	EdDSA key generation algorithm
	[assignment: cryptographic key sizes]	any length from 128 to 528 bits
	[assignment: list of standards]	[35]
FCS_CKM.1.1/ Mont	[assignment: cryptographic key generation algorithm]	MontDH key generation algorithm
	[assignment: cryptographic key sizes]	any length from 128 to 528 bits
	[assignment: list of standards]	[35]
FCS_CKM.6	**Timing and event of cryptographic key destruction**	
FCS_CKM.6.1	[assignment: list of cryptographic keys (including keying material)]	All Keys
	[selection: no longer needed, [assignment: other circumstances for key or keying material destruction]	no longer needed
FCS_CKM.6.2	[assignment: cryptographic key destruction method]	Phyiscally overwriting the key data with random data
	[assignment: list of standards]	none
Application Note:	• The keys are reset as specified in [17] 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 [17] applicable to the security target and the implemented algorithms	
FCS_COP.1		
FCS_COP.1/ GCM	[assignment: list of cryptographic operations]	encryption and decryption
	[assignment: cryptographic algorithm]	AES in GCM mode
	[assignment: cryptographic keysizes]	128, 192 and 256 bits
	[assignment: list of standards]	FIPS 197 NIST Special Publication 800-38D Recommendation for Block Cipher
FCS_COP.1/ TripleDES	[assignment: list of cryptographic operations]	encryption and decryption

Table 29. CoreG API SFR Modifications*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: cryptographic algorithm]	• 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
	[assignment: cryptographic keysizes]	LENGTH_DES3_2KEY LENGTH_DES3_3KEY
	[assignment: list of standards]	ALG_DES_ECB_ISO9797_M2 - see Java Card API [17] All others see JCOPX API [53] and [17]
FCS_COP.1/ AES	[assignment: list of cryptographic operations]	encryption and decryption
	[assignment: cryptographic algorithm]	• 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
	[assignment: cryptographic keysizes]	LENGTH_AES_128 LENGTH_AES_192 LENGTH_AES_256
	[assignment: list of standards]	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 [53], for the rest see Java Card API Spec [17]
FCS_COP.1/ RSACipher	[assignment: list of cryptographic operations]	encryption and decryption
	[assignment: cryptographic algorithm]	• ALG_RSA_NOPAD • ALG_RSA_PKCS1 • ALG_RSA_PKCS1_OAEP

Table 29. CoreG API SFR Modifications*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: cryptographic keysizes]	any kylength that is a multiple of 32 between 512 and 4096 bits
	[assignment: list of standards]	Java Card API Spec [17] and for 32 bit step range, see JCOPX [53]
FCS_COP.1/ ECDH_P1363	[assignment: list of cryptographic operations]	Deffie-Hellman Key Agreement
	[assignment: cryptographic algorithm]	• 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
	[assignment: cryptographic keysizes]	• 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.
	[assignment: list of standards]	Java Card API Spec [17] and for ALG_EC_SVDP_DH_PLAIN_XY s bit step range key size, see JCOPX [53]
FCS_COP.1/ ECDH_25519	[assignment: list of cryptographic operations]	Diffie-Hellman Key Agreement
	[assignment: cryptographic algorithm]	ALG_MONT_DH_25519
	[assignment: cryptographic keysizes]	256 bits
	[assignment: list of standards]	see JCOPX [53]
FCS_COP.1/ DESMAC	[assignment: list of cryptographic operations]	MAC generation and verification
	[assignment: cryptographic algorithm]	Triple DES in outer CDC 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

Table 29. CoreG API SFR Modifications*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: cryptographic keysizes]	LENGTH_DES3_2KEY LENGTH_DES3_3KEY
	[assignment: list of standards]	JCOPX [53]
FCS_COP.1/ AESMAC	[assignment: list of cryptographic operations]	16 byte MAC generation and verification
	[assignment: cryptographic algorithm]	AES in CBC Mode: ALG_AES_MAC_128_ NOPAD
	[assignment: cryptographic keysizes]	LENGTH_AES_128 LENGTH_AES_192 LENGTH_AES_256
	[assignment: list of standards]	Java Card API Spec [17]
FCS_COP.1/ RSASigPKCS1	[assignment: list of cryptographic operations]	digital signature generation and verification
	[assignment: cryptographic algorithm]	• 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 • 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
	[assignment: cryptographic keysizes]	any key length that is a multiple of 32 between 512 and 4096 bits
	[assignment: list of standards]	Java Card API Spec [17] and for the 32 bit step range see API specified in JCOPX [53]
FCS_COP.1/ ECSignature	[assignment: list of cryptographic operations]	digital signature generation and verification
	[assignment: cryptographic algorithm]	• 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]

Table 29. CoreG API SFR Modifications*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: cryptographic keysizes]	• 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
	[assignment: list of standards]	Java Card API Spec [17] and for 1 bit step range key size see API specified in JCOPX [53]
FCS_COP.1/ EdDSA	[assignment: list of cryptographic operations]	digital signature verification
	[assignment: cryptographic algorithm]	SIG_CIPHER_EDDSA SIG_CIPHER_EDDSAPH
	[assignment: cryptographic keysizes]	256 bit for private key, 256 bit for public key
	[assignment: list of standards]	Java Card API Spec [17]
FCS_COP.1/ SHA	[assignment: list of cryptographic operations]	Secure Hash computation
	[assignment: cryptographic algorithm]	• ALG_SHA [1] • ALG_SHA_224 • ALG_SHA_256 • ALG_SHA_384 • ALG_SHA_512 • ALG_SHA3_224 • ALG_SHA3_256 • ALG_SHA3_384 • ALG_SHA3_512
	[assignment: cryptographic keysizes]	• LENGTH_SHA • LENGTH_SHA_224 • LENGTH_SHA_256 • LENGTH_SHA_384 • LENGTH_SHA_512 • LENGTH_SHA3_224 • LENGTH_SHA3_256 • LENGTH_SHA3_384 • LENGTH_SHA3_512
	[assignment: list of standards]	Java Card API Spec [17] and JCOPX API specified in [53]
FCS_COP.1/ AES_CMAC	[assignment: list of cryptographic operations]	CMAC generation and verification

Table 29. CoreG API SFR Modifications*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: cryptographic algorithm]	• 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
	[assignment: cryptographic keysizes]	LENGTH_AES_128 LENGTH_AES_192 LENGTH_AES_256
	[assignment: list of standards]	Java Card API Spec [17] and JCOPX API specified in [53]
FCS_COP.1/ HMAC	[assignment: list of cryptographic operations]	HMAC Generation and Verification
	[assignment: cryptographic algorithm]	• ALG_HMAC_SHA_256 • ALG_HMAC_SHA_384 • ALG_HMAC_SHA_512
	[assignment: cryptographic keysizes]	• LENGTH_SHA_256 • LENGTH_SHA_384 • LENGTH_SHA_512
	[assignment: list of standards]	Java Card API Spec [17] and JCOPX API specified in [53]
FCS_COP.1/ TDES_CMAC	[assignment: list of cryptographic operations]	message authentication and verification
	[assignment: cryptographic algorithm]	• ALG_DES_CMAC8 • SIG_CIPHER)DES_CMAC8
	[assignment: cryptographic keysizes]	• LENGTH_DES3_2KEY • LENGTH_DES3_3KEY
	[assignment: list of standards]	JCOPX API specified in [53]
FCS_COP.1/ DAP	[assignment: list of cryptographic operations]	verification of the DAP signature attached to Executable Load Applications
	[assignment: cryptographic algorithm]	• ALG_RSA_SHA_PKCS1 • ALG_ECDSA_SHA_256
	[assignment: cryptographic keysizes]	• LENGTH_RSA_1024 • LENGTH_EC_FP_256
	[assignment: list of standards]	JCOPX API specified in [53] and GlobalPlatform Specification [25]
FCS_RNG.1.1 [2]	[selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]	Hybrid
	[selection: DRG.2, DRG.3, DRG.4, PTG.2, PTG.3, NTG.1] [AIS20] [AIS31]	[DRG.3] [AIS20]

Table 29. CoreG API SFR Modifications*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: list of security capabilities]	• (DRG.3.1) If initialized with a random seed using a PTRNG of class PTG.2 (as defined in [9]) 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 [9]). • (DRG.3.3) The RNG provides enhanced backward secrecy even if the current internal state is known (as defined in [9])
FCS_RNG.1.2	[assignment: a defined quality metric]	• (DRG.3.4) The RNG, initialized with a random seed using a PTRNG of class PTG.2 (as defined in [9]) as random source, generates output for which for AES-mode 2^{48} and for TDEA-mode 2^{35} strings of bit length 128 are mutually different with probability at least $1\text{-}2^{24}$ • (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 [9]).
FCS_RNG.1 :	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. • The TOE supports TDEA and AES mode, where AES mode is the default mode. Only NXP can configure the mode to TDEA, during initialization of the TOE, at pre-delivery.
FCS_RNG.1.1/ HDT	[selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]	Hybrid
	[selection: DRG.2, DRG.3, DRG.4, PTG.2, PTG.3, NTG.1] [AIS20] [AIS31]	[DRG.4] [AIS20]

Table 29. CoreG API SFR Modifications...*continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: list of security capabilities]	• (DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 (as defined in [9]) as random source • (DRG.4.2) The RNG provides forward secrecy (as defined in [9]). • (DRG.4.3) The RNG provides backward secrecy even if the current internal state is known (as defined in [9]) • (DRG.4.4) The RNG provides enhanced forward secrecy on demand (as defined in [9]) • (DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2 (as defined in [9])
FCS_RNG.1.2/ HDT	[assignment: a defined quality metric]	• (DRG.4.6) The RNG generates output for which for AES-mode 2^{48} and for TDEA-mode 2^{35} 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 [9]).
FCS_RNG.1/ HDT	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. • The TOE supports TDEA and AES mode, where AES mode is the default mode. Only NXP can configure the mode to TDEA, during initialization of the TOE, at pre-delivery.

[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.

[2] FCS_RNG is an extended component family defined in the Java Card Protection Profile [14]

7.1.1.1.3 Card Security Management

The Security Functional Requirements for the Card Security Management section of the CoreG group are defined in strict compliance with the Security Problem Definition described in [14].

The following table indicates which SFRs are modified by this Security Target:

Table 30. CoreG Card Security SFRs

SFR ID	Modified
FAU_ARP.1	FAU_ARP.1

Table 30. CoreG Card Security SFRs *...continued*

SFR ID	Modified
FDP_SDI.1/DATA	FDP_SDI.2/DATA
FPR_UNO.1	FPR_UNO.1
FPT_FLS.1	FPT_FLS.1
FPT_TDC.1	FPT_TDC.1

The following table provides the selection and assignments for the modified SFRs:

Table 31. CoreG Card Security SFR Modifications

SFR ID	Selection / Assignment text	Selection / Assignment value
FAU_ARP.1.1	[assignment: list of actions]	one of the following actions: • throw an exception • lock the card session *(after a predefined number of resetted session might switch to Restricted Mode)* • reinitialize the Java Card System and its data • *reponse with error code to S.CAD*
	Refinement:	The "potential security vilation stands for one of the following events: • CAP: CAP file inconsistency (response with error code to S.CAD), • LFC: applet life cycle inconsistency (throw an exception), • CHP: card tearing (unexpected removal of the Card out of the CAD) and power failure (reset the card session), • ABT: abort of a transaction in an unexpected context (throw an exception), • FWL: violation of the Firewall or JCVM SFPs (throw an exception), • RSC: unavailability of memory (throw an exception), • OFL: array overflow (throw an exception), • EDC: checksum mismatch of EDC arrays (throw an exception), • assignment: – Physical Tampering – CLC: Card Manager Life Cycle inconsistency (reset the card session), – CHP: General Fault Injection Detection (reset the card session) – CHP: Memory defects (reset the card session), – CHP: Integrity protected persistent data inconsistency (reset the card session), – CHP: Integrity protected transient data inconsistency (reset the card session), – Memory Access Violation – CHP: Others (reset the card session)

Table 31. CoreG Card Security SFR Modifications...*continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_SDI.2.1/ DATA	[assignment: integrity errors]	intergrity errors
	[assignment: user data attributes]	integrity protected data
FDP_SDI.2/ DATA	[assignment: action to be taken]	reset the card session for integrity errors
FPR_UNO.1.1	[assignment: list of users and/ or subjects]	all users
	[assignment: list of operations]	all operations
	[assignment: list of objects]	D.APP_KEYS, D.PIN
	[assignment: assignment: list of protected users and/or subjects]	another user
FPT_FLS.1.1	[assignment: list of types of failures in the TSF]	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 ([19], §6.2.3) or after a proximity card (PICC) activation sequence ([19]). Behavior of the TOE on power loss and reset is described in [19], §3.6 and §7.1. Behavior of the TOE on RF signal loss is described in [19], §3.6.1.
	Refinement:	The term “failure” above also covers “circumstances” for assignments taken from [13]. The TOE prevents failures for the “circumstances” defined above.
FPT_TDC.1.1	[assignment: list of TSF data types]	the CAP files, the bytecode and its data arguments
FPT_TDC.1.2	[assignment: list of interpretation rules to be applied by the TSF]	• the rules defined in [18] specification • the API tokens defined in the export files of reference implementation
	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

The Security Functional Requirements for the AID Management section of the CoreG group are defined in strict compliance with the Security Problem Definition described in [14].

The following table indicates which SFRs are modified by this Security Target:

Table 32. CoreG AID Management SFRs

SFR ID	Modified
FIA_ATD.1/AID	as per [14]
FIA_UID.2/AID	as per [14]
FIA_USB.1/AID	FIA_USB.1/AID
FMT_MTD.1/JCRE	as per [14]
FMT_MTD.2/JCRE	as per [14]

The following table provides the selection and assignments for SFRs:

Table 33. CoreG AID Management SFR Modifications

SFR ID	Selection / Assignment text	Selection / Assignment value
FIA_USB.1/ AID	**User-Subject binding**	
FIA_USB.1.1/ AID	[assignment: list of user security attributes]	CAP file AID
FIA_USB.1.2/ AID	[assignment: rules for the initial association of attributes]	Each uploaded package is associated with a unique Package AID
FIA_USB.1.3/ AID	[assignment: rules for the changing of attributes]	The initially assigned Package AID is unchangeable

7.1.1.2 InstG Security Functional Requirements

The Security Functional Requirements for the InstG section are defined in strict compliance with the Security Problem Definition described in [14].

The following table indicates which SFRs are modified by this Security Target:

Table 34. InstG SFRs

SFR ID	Modified
FDP_ITC.2/INSTALLER	Refined and forms part of Card Management SFRs (CarG) as FDP_ITC.2/CCM
FMT_SMR.1/INSTALLER	as per [14]
FPT_FLS.1/INSTALLER	as per [14]
FPT_RCV.3/INSTALLER	as per [14]

7.1.1.3 ADELG Security Functional Requirements

The Security Functional Requirements for the ADELG section are defined in strict compliance with the Security Problem Definition described in [14].

The following table indicates which SFRs are modified by this Security Target:

Table 35. ADELG SFRs

SFR ID	Modified
FDP_ACC.2/ADEL	as per [14]
FDP_ACF.1/ADEL	as per [14]
FMT_MSA.1/ADEL	as per [14]

Table 35. ADELG SFRs...*continued*

SFR ID	Modified
FMT_MSA.3/ADEL	as per [14]
FMT_SMR.1/ADEL	as per [14]
FDP_RIP.1/ADEL	as per [14]
FPT_FLS.1/ADEL	as per [14]

7.1.1.4 ODELG Security Functional Requirements

The Security Functional Requirements for the ODEL section are defined in strict compliance with the Security Problem Definition described in [14].

The following table indicates which SFRs are modified by this Security Target:

Table 36. ODELG SFRs

SFR ID	Modified
FDP_RIP.1/ODEL	as per [14]
FPT_FLS.1/ODEL	as per [14]

7.1.1.5 CARG Security Functional Requirements

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:

- Security Domain access control policy
- Secure Channel Protocol information flow policy

These SFRs provide a concrete and more restrictive implementation of the PACKAGE LOADING information flow control SFP from PP [14] by following the information flow policy defined by Global Platform specifications. The table below lists the SFRs from CarG of PP and their corresponding refinements in this ST. The Security Functional Requirements for the CARG section are defined in strict compliance with the Security Problem Definition described in , section 7.2.5.

Table 37. CARG SFRs

SFR ID	Modified
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
FMT_MSA.1/CM	FMT_MSA.1 /SC
FMT_MSA.3/CM	FMT_MSA.3 /SC
FMT_SMR.1/CM	FMT_SMR.1/SD
FIA_UID.1/CM	FIA_UID.1/SC
FTP_ITC.1/CM	FTP_ITC.1 /SC
FMT_SMF.1/CM	FMT_SMF.1 /SC

The following SFRs have been added to this group in this security target:

Table 38. Additional CARG SFRs

SFR ID	Modified
FDP_ITC.2/CCM	FDP_ITC.2/CCM is a refinement of FTP_ITC.2/Installer from the PP [14]
FDP_ROL.1/CCM	FPT_ROL.1/CCM
FPT_FLS.1/CCM	FPT_FLS.1/CCM
FPT_PHP.3	FPT_PHP.3

The following table provides the selection and assignments for the additional SFRs:

Table 39. CARG Additional SFR modifications

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_ITC.2/CCM		
FDP_ITC.2.1/ CCM	[assignment: access control SFP(s) and/or information flow control SFP(s)]	
FDP_ITC.2.2/ CCM	No operation	
FDP_ITC.2.3/ CCM	No operation	
FDP_ITC.2.4/ CCM	No operation	
FDP_ITC.2.5/ CCM	[assignment: additional importation control rules]	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 ([18], §4.5.2).]
	Application Note:	This SFR also covers security functionality required by Amendment A of the GP specification [22] , i.e. personalizing SDs and loading ciphered load files.
FDP_ROL.1/CCM		
FDP_ROL.1.1/ CCM	[assignment: access control SFP(s) and/or information flow control SFP(s)]	Security Domain access control policy
	[assignment: list of operations]	installation operation
	assignment: information and/or list of objects]	executable files and application instances
FDP_ROL.1.2/ CCM	[assignment: boundary limit to which rollback may be performed].	boundaries of available memory before the card content management function started
FPT_FLS.1/CCM		

Table 39. CARG Additional SFR modifications...*continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FPR_FLS.1.1/ CCM	[assignment: list of types of failures in the TSF]	the Security Domain fails to load/install an Executable File/application instance as described in [19], Section 11.1.5
FPT_PHP.3		
FPT_PHP.3.1	[assignment: physical tampering scenarios]	physical manipulation and physical probing
	[assignment: list of TSF devices/elements]	TSF
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 PP [13].	

7.1.1.5.1 Secure Channel

The SFRs in this section provide additional proprietary features, as allowed by the PP [14], and add functionality to the TOE making the SFRs more restrictive than the PP alone.

Table 40. Secure Channel SFRs

SFR ID	Modified
FCS_NRO.2/SC	FCS_NRO.2/SC
FDP_IFC.2/SC	FDP_IFC.2/SC
FDP_IFF.1/SC	FDP_IFF.1/SC
FMT_MSA.1/SC	FMT_MSA.1/SC
FMT_MSA.3/SC	FMT_MSA.3/SC
FMT_SMF.1/SC	FMT_SMF.1/SC
FIA_UID.1/SC	FIA_UID.1/SC
FIA_UAU.1/SC	FIA_UAU.1/SC
FIA_UAU.4/SC	FIA_UAU.4/SC
FTP_ITC.1/SC	FTP_ITC.1/SC

The operations on those SFRs are detailed in the following table:

Table 41. SC SFR operations

SFR ID	Selection / Assignment text	Selection / Assignment value
FCO_NRO.2/ SC	**Enforced proof of origin**	
FCO_ NRO.2.1/SC	[assignment: list of information types]	Executable load files

Table 41. SC SFR operations*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	Application Note:	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.2.2/SC	[assignment: list of attributes]	DAP Block
	[assignment: list of information fields]	Identity
FCO_ NRO.2.3/SC	[selection: originator, recipient, [assignment: list of third parties]]	originator
	[assignment: limitations on the evidence of origin]	at the time the Executable load files are received as no evidence is kept on the card for future verification
	Application Note:	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**	
FDP_IFC.2.1/ SC	[assignment: information flow control SFP]	Secure Channel Protocol information flow control policy
	[assignment: list of subjects and information]	• 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
FDP_IFC.2.2/ SC	no operations	-
FDP_IFF.1/SC	**Simple security attributes**	
FDP_IFF.1.1/ SC	[assignment: information flow control SFP]	Secure Channel Protocol information flow control policy

Table 41. SC SFR operations*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]	• 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
FDP_IFF.1.2/ SC	[assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]	Runtime behaviour rules defined by GlobalPlatform [21] for • loading; Section 9.3.5 • installation; Section 9.3.6 • extradition; Section 9.4.1 • registry update; Section 9.4.2 • content removal; Section 9.5
FDP_IFF.1.3/ SC	[assignment: additional information flow control SFP rules]	none
FDP_IFF.1.4/ SC	[assignment: rules, based on security attributes, that explicitly authorise information flows]	none
FDP_IFF.1.5/ SC	[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
Application Note:	• The subject S.SD can be the ISD or APSD • The on-card and the off-card subjects have security attributes such as MAC, Cryptogram, Challenge, Key Set, Static Keys, etc.	
FDP_UIT.1/ CCM	**Data exchange integrity**	
FDP_UIT.1.1/ CCM	[assignment: access control SFP(s) and/or information flow control SFP(s)]	Secure Channel Protocol information flow control policy and the Security Domain access control policy
	[selection: transmit, receive]	receive
	[selection: modification, deletion, insertion, replay]	modification, deletion, insertion and replay
FDP_UIT.1.2/ CCM	[selection: modification, deletion, insertion, replay]	modification, deletion, insertion, replay *of some of the pieces of the application sent by the CAD*
Application note: FDP_UIT.1/ CCM	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.	

Table 41. SC SFR operations*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FMT_MSA.1/ SC	**Management of security attributes**	
FMT_MSA.1.1/ SC	[assignment: access control SFP(s), information flow control SFP(s)]	Secure Channel Protocol information flow control policy
	[selection: change_default, query, modify, delete, [assignment: other operations]]	modify
	[assignment: list of security attributes]	• Key Set • Security Level • Secure Channel Protocol • Session Keys • Sequence Counter • ICV
	[assignment: the authorised identified roles]	the actor associated with the according security domain: • The Card Issuer for ISD • The Application Provider for APSD
Application Note: (FMT_MSA.1/ SC)	The key data used for setting up a secure channel is according to GP spec [21], Amendment D [24].	
FMT_MSA.3/ SC	**Static attribute initialisation**	
FMT_MSA.3.1/ SC	[assignment: access control SFP, information flow control SFP]	Secure Channel Protocol information flow control policy
	[selection, choose one of: restrictive, permissive, [assignment: other property]]	restrictive
FMT_MSA.3.2/ SC	[assignment: the authorised identified roles]	Card Issuer, Application Provider
FMT_SMR.1/ SD	**Security Roles**	
FMT_ SMR.1.1/SD	[assignment: the authorised identified roles]	ISD, SSD
FMT_ SMR.1.1/SD	No Operations	
FIA_UID.1/SC	**Timing of identification**	
FIA_UID.1.1/ SC	[assignment: list of TSF-mediated actions]	• application selection • initializing a secure channel with the card • requesting data that identifies the card or the Card Issuer
FIA_UID.1.2/ SC	No Operations	

Table 41. SC SFR operations...*continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
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**	
FIA_UAU.1.1/ SC	[assignment: list of TSF mediated actions]	the TSF mediated actions listed in FIA_UID.1/ SC
FIA_UAU.1.2/ SC	No Operations	
FIA_UAU.4/ SC	**Single-use authentication mechanisms**	
FIA_UAU.4.1/ SC	[assignment: identified authentication mechanism(s)]	the authentication mechanism used to open a secure communication channel with the card
FTP_ITC.1/SC	**Inter-TSF trusted channel**	
FTP_ITC.1.1/ SC	No Operation	
FTP_ITC.1.2/ SC	[selection: the TSF, another trusted IT product]	another trusted IT product
FTP_ITC.1.3/ SC	[assignment: list of functions for which a trusted channel is required]	all card management functions including: • loading • installation • extradition • registry update • content removal • changing the Application Life Cycle or Card Life Cycle
FMT_SMF.1/ SC	**Specification of Management Functions**	
FMT_SMF.1.1/ SC	[assignment: list of management functions to be provided by the TSF]	Management functions specified by GlobalPlatform [21] • loading (Section 9.3.5) • installation (Section 9.3.6) • extradition (Section 9.4.1) • registry update (Section 9.4.2) • content removal (Section 9.5)
Application Note:	All management functions related to secure channel protocols shall be relevant.	

7.1.1.5.2 Security Domains

The SFRs in this section provide additional proprietary features, as allowed by the PP [14], and add functionality to the TOE making the SFRs more restrictive than the PP alone.

Table 42. Further Proprietary Security Domain SFRs

SFR ID	ST Instantiation
FDP_ACC.1/SD	FDP_ACC.1/SD

Table 42. Further Proprietary Security Domain SFRs*...continued*

SFR ID	ST Instantiation
FDP_ACF.1/SD	FDP_ACF.1/SD
FMT_MSA.1/SD	FMT_MSA.1/SD
FMT_MSA.3/SD	FMT_MSA.3/SD
FMT_SMF.1/SD	FMT_SMF.1/SD
FMT_SMR.1/SD	FMT_SMR.1/SD

Table 43. Security Domain SFR

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_ACC.1/SD	**Subset Access Control**	
FDP_ACC.1.1/SD	[assignment: access control SFP]	Security Domain access control policy
	[assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]	• Subjects: S.INSTALLER, S.ADEL, S.CAD 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**	
FDP_ACF.1.1/SD	[assignment: access control SFP]	Security Domain access control policy

Table 43. Security Domain SFR ***...continued***

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: 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]	• Subjects: • – S.INSTALLER, defined in [10] and represented by the GlobalPlatform Environment (OPEN) on the card, the Card Life Cycle attributes (defined in Section 5.1.1 of [17]) – S.ADEL, also defined in [10] 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 [17]), a Life-cycle Status (defined in Section 5.3.2 of [17]) and a Secure Communication Security Level (defined in Section 10.6 of [17]) – S.CAD, defined in [10], 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	[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	Runtime behavior rules defined by GlobalPlatform [21] for: • loading (Section 9.3.5) • installation (Section 9.3.6) • extradition (Section 9.4.1) • registry update (Section 9.4.2) • content removal (Section 9.5)

Table 43. Security Domain SFR *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_ACF.1.3	[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]	
FDP_ACF.1.4	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	
FMT_MSA.1/ SD	**Management of security attributes**	
FMT_MSA.1.1/ SD	[assignment: access control SFP(s), information flow control SFP(s)]	
	[selection: change_default, query, modify, delete, [assignment: other operations]]	
	[assignment: list of security attributes]	
	[assignment: the authorised identified roles].	
FMT_MSA.3/ SD	**Static attribute initialisation**	
FMT_MSA.3.1	[assignment: access control SFP, information flow control SFP]	
	[selection, choose one of: restrictive, permissive, [assignment: other property]]	
FMT_MSA.3.2/ SD	[assignment: the authorised identified roles]	
FMT_SMF.1/ SD	**Specification of Management Functions**	
FMT_SMR.1/ SD	**Security Roles**	
FMT_ SMR.1.1/SD	[assignment: the authorised identified roles]	ISD, SSD
FMT_ SMR.1.1/SD	No Operations	

7.1.1.6 Optional Java Card Packages SFRs

The SFRs in this section are defined in the optional packages of the PP [14]

Table 44. Implemented Package

Package	SFRs defined by PP
SENSITIVE RESULT	FDP_SDI.2/RESULT

Table 44. Implemented Package*...continued*

Package	SFRs defined by PP
MONOTONIC COUNTERS	FDP_SDI.2/MONOTONIC_COUNTER
CRYPTOGRAPHIC CERTIFICATE MANAGEMENT	FDP_SDI.2/CRT_MNGT FCS_COP.1/CRT_MNGT
KEY DERIVATION FUNCTIONS (KDF)	FCS_CKM.5/KDF
SYSTEM_TIME	FPT_STM.1/SYS_TIME

Table 45. Additional SFR operations

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_SDI.2/ RESULT	**Stored data integrity monitoring and action**	
FDP_SDI.2.1/ RESULT	[assignment: integrity errors]	integrity errors
	[assignment: user data attributes]	sensitive API result stored in the javacardx.security.SensitiveResult class
FDP_SDI.2.2/ RESULT	assignment: action to be taken	throw an exception
FDP_SDI.2/ MONOTONIC_ COUNTER	**Stored data integrity monitoring and action**	
FDP_SDI.2.1/ MONOTONIC_ COUNTER	[assignment: integrity errors]	integrity errors
	[assignment: user data attributes]	stored user data i.e. the counter value in the MonotonicCounter object
FDP_SDI.2.2/ MONOTONIC_ COUNTER	assignment: action to be taken	throw an exception
FDP_SDI.2/ CRT_MNGT	**Stored data integrity monitoring and action**	
FDP_SDI.2.1/ CRT_MNGT	[assignment: integrity errors]	integrity errors
	[assignment: user data attributes]	cryptographic certificate
FDP_SDI.2.2/ CRT_MNGT	assignment: action to be taken	throw an exception
FCS_COP.1/ CRT_MNGT	**Cryptographic Operation**	
FCS_COP.1.1/ CRT_MNGT	[assignment: list of cryptographic operations]	verification of X.509 Certificate

Table 45. Additional SFR operations*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment:cryptographic algorithm]	RSA_SSA_PSS with digests • SHA1_HASH • SHA224_HASH • SHA256_HASH • SHA384_HASH • SHA512_HASH PKSC#1 v1.5 with digests • ALG_MD5 • SHA1_HASH • SHA224_HASH • SHA256_HASH • SHA384_HASH • SHA512_HASH ECDSA with digests: • SHA1_HASH • SHA224_HASH • SHA256_HASH • SHA384_HASH • SHA512_HASH
	[assignment: cryptographic key sizes]	• RSA – LENGTH_RSA_512 – LENGTH_RSA_736 – LENGTH_RSA_768 – LENGTH_RSA_896 – LENGTH_RSA_1024 – LENGTH_RSA_1280 – LENGTH_RSA_1536 – LENGTH_RSA_1984 – LENGTH_RSA_2048 – LENGTH_RSA_3072 – LENGTH_RSA_4096 • ECDSA – 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_521 – LENGTH_EC_FP_528
	[assignment: list of standards]	[48]
FCS_CKM.5/ KDF	**Cryptographic Key Derivation**	
FCS_CKM.5.1/ KDF	[assignment: key type]	Output is either: Data Output: Byte Array up to 0x7FFF bytes Dedicated key types: DES, AES, GENERIC_ SECRET, HMAC, KOREAN_SEED, SM4

Table 45. Additional SFR operations*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[Assignment: Input Paramters]	Input Data according to the Specification 3 • ALG_KDF_ANSI_X9_63 Secret, SharedInfo • ALG_KDF_COUNTER_MODE CounterLength, InputDataAfterCounter, InputDataBeforeCounter, Secret • ALG_KDF_HKDF Secret, Salt, Info • ALG_KDF_ICAO_MRTD Secret, Counter • ALG_PRF_TLS12 Secret, Seed
	[assignment: key derivation algorithm]	• ALG_KDF_ANSI_X9_63 – Signature.ALG_HMAC_SHA1 – Signature.ALG_HMAC_SHA256 – Signature.ALG_HMAC_SHA384 – Signature.ALG_HMAC_SHA512 – Signature.ALG_CMAC_128 • ALG_KDF_COUNTER_MODE • ALG_KDF_HKDF • ALG_KDF_ICAO_MRTD • ALG_PRF_TLS12
	[assignment: list of key sizes]	ByteArray up to 0x7FFF bytes HMAC Key up to 896 bytes GenericSecretKey up to 256 bytes LENGTH_DES, LENGTH_DES3_2KEY, LENGTH_DES3_3KEY LENGTH_AES_192, LENGTH_AES_256, LENGTH_AES_512 LENGTH_KOREAN_SEED_128, LENGTH_KOREAN_SEED_256 LENGTH_SM4
	[assignment: list of standards]	[17] implementing [49], [40], [47], [16] and [46]
FPT_STM.1/ SYS_TIME	**Reliable Time Stamps**	
FPT_STM.1.1/ SYS_TIME	No Operations	

7.1.1.7 Further Security Functional Requirements for JCOP

The SFRs in this section provide additional proprietary features, as allowed by the PP [14], and add functionality to the TOE making the SFRs more restrictive than the PP alone.

Table 46 lists the additional Java Card SFRs, whilst the SFRS dedicated to specific additional functionality are sub-categorised below:

- Configuration Applet
- OS Update

- Restricted Mode
- Context Seperation

Table 46. Further Proprietary SFRs

SFR ID	ST Instantiation
FAU_SAS.1/SCP	FAU_SAS.1/SCP
FIA_AFL.1/PIN	FIA_AFL.1/PIN
FPT_EMS.1	FPT_EMS.1
FCS_CKM.2	FCS_CKM.2
FCS_CKM.2	FCS_CKM.3

The SFRs for further defined functionalities are sub-categorised below:

The following table provides the operations on the additional SFRs mentioned above:

Table 47. Additional SFR operations

SFR ID	Selection / Assignment text	Selection / Assignment value
FAU_SAS.1/ SCP	**Audit storage**	
FAU_SAS.1.1/ SCP	[assignment: list of subjects]	
	[assignment: type of persistent memory]	
FIA_AFL.1/ PIN	**Authentication Failure Handling (PIN)**	
FIA_AFL.1.1/ PIN	[selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]	an administrator configurable positive integer within [1 and 127]
	[assignment: list of authentication events].	
FIA_AFL.1.2/ PIN	[assignment: list of authentication events].	surpassed
	[assignment: list of actions]	block any user authentication using 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	**TOE Emanation**	
FPT_EMS.1.1	[assignment: types of emission]	variations in power consumption or timing during command execution
	[assignment: list of types of attack surface]	any
	[assignment: list of types of TSF data]	TSF data: D.CRYPTO
	[assignment: list of types of user data]	D.PIN, D.APP_KEYs

Table 47. Additional SFR operations*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FPT_PHP.3	Resistance to Physical Attack	
FPT.PHP.3.1	[assignment: physical tampering scenarios]	physical manipulation andphysical probing
	[assignment: list of TSF devices/elements]	TSF
Refinement	The TSF will implement appropriate mechanisms to continuouslycounter physical manipulation and physical probing. Due to thenature 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 ensuringthat security functional requirements are enforced. Hence, ”automatic response” means here (i) assuming that there mightbe an attack at any time and (ii) countermeasures are provided atany time.	
Application Note:	This SFR is taken from the certified Security IC PlatformProtection Profile [13]	
FCS_CKM.2	**Cryptographic Key Distribution**	
FCS_CKM.2.1	[assignment: cryptographic key distribution method]	methods: set keys and components of DES, AES, RSA, RSA-CRT, ECC, ECDH, HMAC, XEC, GenericSecret keys
	[assignment: list of standards]	[17], [53]
Application Note:	• The keys can be accessed as specified in KeyClass specified in [17] and in [53] for proprietary keys • The component shall be instantiated according to the version of the Java Card API applying to the Security Target and the implemented algorithms in [17] and in [53] for proprietary classes	
FCS_CKM.3	**Cryptographic Key Access**	
FCS_CKM.2.1	[assignment: type of cryptographic key access]	assignment: management of DES, AES, RSA, RSA-CRT, ECC, ECDH, HMAC, XEC, and GenericSecret Keys
	[assignment: cryptographic key access method]	methods/commands defined in packages javacard.security of [17] and in [53] for proprietary classes
	[assignment: list of standards]	[17], [53]
Application Note:	• The keys can be accessed as specified in KeyClass specified in [17] and in [53] for proprietary keys • The component shall be instantiated according to the version of the Java Card API applying to the Security Target and the implemented algorithms in [17] and in [53] for proprietary classes	

7.1.1.7.1 Configuration Applet

The SFRs in this section provide additional proprietary features, as allowed by the PP [14], and add functionality to the TOE making the SFRs more restrictive than the PP alone.

Table 48. Configuration Applet SFRs

SFR ID	Modified
FDP_IFC.2/CFG	FDP_IFC.2/CFG

Table 48. Configuration Applet SFRs*...continued*

SFR ID	Modified
FDP_IFF.1/CFG	FDP_IFF.1/CFG
FIA_UID.1/CFG	FIA_UID.1/CFG
FMT_MSA.1/CFG	FMT_MSA.1/CFG
FMT_MSA.3/CFG	FMT_MSA.3/CFG
FMT_SMF.1/CFG	FMT_SMF.1/CFG
FMT_SMR.1/CFG	FMT_SMR.1/CFG

Table 49. Configuration Applet SFRs

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_IFC.2/CFG	**Complete Information Flow Control (CFG)**	
FDP_IFC.2.1/CFG	[assignment: information control SFP]	Configuration Information flow control SFP
	[assignment: list of subjects and information]	S.Customer, S.NXP, S.ConfigurationMechanism and D.CONFIG_ITEM
FDP_IFC.2.2/CFG	No Operation	
FDP_IFF.1/CFG	**Simple Security Attributes (CFG)**	
FDP_IFF.1.1/CFG	[assignment: information flow control SFP]	CONFIGURATION Information flow control SFP
	[assignment : list of subjects and information controlled under the indicated SFP, and for each, the security attributes]	• 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

Table 49. Configuration Applet SFRs *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_IFF.1.2/ CFG	[assignment: for each operation, the security attribute-based relationship that hold between subject and information security attributes]	• 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	[assignment: additional information flow control SFP rules]	none
FDP_IFF.1.4	[assignment: rules, based on security attributes, that explicitly authorize information flows]	none
FDP_IFF.1.5	[assignment: rules, based on security attributes, that explicitly deny information flows]	• 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.
FIA_UID.1/ CFG	**Timing of Identification (CFG)**	
FIA_UID.1.1/ CFG	[assignment: list of TSF-mediated actions]	to select the Runtime Configuration interfaces
FIA_UID.1.2/ CFG	No operations	
FMT_MSA.1/ CFG	**Management of security attributes (CFG)**	
FMT_MSA.1.1/ CFG	[assignment: access control SFP(s), information flow control SFP(s)]	CONFIGURATION information flow control SFP
	[selection: change_default, query, modify, delete, [assignment: other operations]]	modify [none]
	[assignment: list of security attributes]	NXP Configuration Access and Customer Configuration Access
	[assignment: the authorized identified roles]	S.NXP and S.Customer

Table 49. Configuration Applet SFRs *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FMT_MSA.3/ CFG	**Static Attribute Initialisation (CFG)**	
FMT_MSA.3.1/ CFG	[assignment: access control SFP, information flow control SFP]	CONFIGURATION information flow control SFP
	[selection, choose one of: restrictive, permissive, [assignment: other property]]	restrictive [none]
FMT_MSA.3.2/ CFG	[assignment: the authorized identified roles]	none
FMT_SMF.1/ CFG	**Specification of Management Functions (CFG)**	
FMT_SMF.1.1/ CFG	[assignment: list of management functions to be provided by the TSF]	disable the NXP Configuration Access, disable the Customer Configuration Access
FMT_SMR.1/ CFG	**Security Roles (CFG)**	
FMT_ SMR.1.1/CFG	[assignment: the authorized identified roles]	S.NXP and S.Customer
FMT_ SMR.1.2/CFG	No Operation	

7.1.1.7.2 OS Update

The SFRs in this section provide additional proprietary features, as allowed by the PP [14], and add functionality to the TOE making the SFRs more restrictive than the PP alone.

Table 50. OS Update SFRs

SFR ID	Modified
FDP_IFC.2/OSU	FDP_IFC.2/OSU
FDP_IFF.1/OSU	FDP_IFF.1/OSU
FMT_MSA.1/OSU	FMT_MSA.1/OSU
FMT_MSA.3/OSU	FMT_MSA.3/OSU
FMT_SMR.1/OSU	FMT_SMR.1/OSU
FMT_SMF.1/OSU	FMT_SMF.1/OSU
FIA_UID.1/OSU	FIA_UID.1/OSU
FIA_UAU.1/OSU	FIA_UAU.1/OSU
FIA_UAU.4/OSU	FIA_UAU.4/OSU
FPT_FLS.1/OSU	FPT_FLS.1/OSU

Table 51. OSUpdate (OSU) SFRs

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_IFC.2/ OSU	**Complete Information flow control**	
FDP_IFC.2.1/ OSU	[assignment: information flow control SFP]	OS Update information flow control SFP
	[assignment: list of subjects and information]	S.OSU and D.UPDATE_IMAGE
FDP_IFC.2.2/ OSU	No Operation	
FDP_IFF.1/ OSU	**Simple Security Attributes**	
FDP_IFF.1.1/ OSU	[assignment: information flow control SFP]	OS Update information flow control SFP
	[assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]	• 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	[assignment: for each operation, the security attribute-based relationship that hold between subject and information security attributes]	• 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	[assignment: additional information flow control SFP rules]	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	[assignment: rules, based on security attributes, that explicitly authorize information flows]	none
FDP_IFF.1.5/ OSU	[assignment: rules, based on security attributes, that explicitly deny information flow]	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.	

Table 51. OSUpdate (OSU) SFRs *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FMT_MSA.1/ OSU	**Management of Security Attributes**	
FMT_MSA.1.1/ OSU	[assignment: access control SFP(s), information flow control SFP(s)]	OS Update information flow control SFP
	[selection: change_default, query, modify, delete, [assignment: other operations]	modify
	[assignment: list of security attributes]	Current Sequence Number
	[assignment: the authorized identified roles]	S.OSU
FMT_MSA.3/ OSU	**Static Attribute Initialisation**	
FMT_MSA.3.1/ OSU	[assignment: access control SFP, information flow control SFP]	OS Update information flow control SFP
	[selection, choose one of: restrictive, permissive, [assignment: other property]]	restrictive [none]
FMT_MSA.3.2/ OSU	[assignment: the authorized identified roles]	none
FMT_SMR.1/ OSU	**Security Roles**	
FMT_ SMR.1.1/OSU	[assignment: the authorized identified roles]	S.OSU
FMT_ SMR.1.2/OSU	No operation	
FMT_SMF.1/ OSU	**Specification of Management Functions**	
FMT_SMF.1.1/ OSU	[assignment: list of management functions to be provided by the TSF]	• 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 SequenceNumber.	
FIA_UID.1/ OSU	**Timing of Identification**	
FIA_UID.1.1/ OSU	[assignment: list of TSF-mediated actions]	OP.TRIGGER_UPDATE
FIA_UID.1.2/ OSU	No Operation	
FIA_UAU.1/ OSU	**Timing of Authentication**	
FIA_UAU.1.1/ OSU	[assignment: list of TSF mediated actions]	OP.TRIGGER_UPDATE

Table 51. OSUpdate (OSU) SFRs *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FIA_UAU.1.2/ OSU	No Operation	
FIA_UAU.4/ OSU	**Single-use authentication mechanisms**	
FIA_UAU.4.1/ OSU	[assignment: identified authentication mechanism(s)]]	the authentication mechanism used to load D.UPDATE_IMAGE
FPT_FLS.1/ OSU	**Failure with Preservation of Secure State**	
FPT_FLS.1.1/ OSU	[assignment: list of types of failures in the TSF]	• 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.

7.1.1.7.3 Restricted Mode

The SFRs in this section provide additional proprietary features, as allowed by the PP [14], and add functionality to the TOE making the SFRs more restrictive than the PP alone.

Table 52. Restricted Mode SFRs

SFR ID	Modified
FDP_ACC.2/RM	FDP_ACC.2/RM
FDP_ACF.1/RM	FDP_ACF.1/RM
FMT_MSA.1/RM	FMT_MSA.1/RM
FMT_MSA.3/RM	FMT_MSA.3/RM
FMT_SMR.1/RM	FMT_SMR.1/RM
FMT_SMF.1/RM	FMT_SMF.1/RM
FIA_UID.1/RM	FIA_UID.1/RM
FIA_UAU.1/RM	FIA_UAU.1/RM

Table 53. Restricted Mode (RM) SFRs

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_ACC.2/ RM	**Complete Access Control**	
FDP_ACC.2.1/ RM	[assignment: access control SFP]	Restricted Mode Access Control SFP
	[assignment: list of subjects andobjects]	S.ACAdmin
FDP_ACC.2.2/ RM	No Operation	
FDP_ACF.1/ RM	**Security Attribute based Access Control**	

Table 53. Restricted Mode (RM) SFRs ***...continued***

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_ACF.1.1/ RM	[assignment: access control SFP	Restricted Mode Access Control SFP
	[assignment: 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]	S.ACAdmin : Security Attribute - Attack Counter
FDP_ACF.1.2/ RM	[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	The Attack Counter can be reset by S.ACAdmin
FDP_ACF.1.3/ RM	[assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects]	none
FDP_ACF.1.4/ RM	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	Deny all operations on all objects when the TOE is in restricted mode,except for operations listed in FMT_SMF.1/RM
FMT_MSA.1/ RM	**Management of Security Attributes**	
FMT_MSA.1.1/ RM	[assignment: access control SFP(s), information flow control SFP(s)]	Restricted Mode Access Control
	[selection: change_default, query, modify, delete, [assignment: other operations]	change_default, [reset]
	[assignment: list of security attributes]	Attack Counter
	[assignment: the authorized identified roles]	S.ACAdmin
FMT_MSA.3/ RM	**Static Attribute Initialisation**	
FMT_MSA.3.1/ RM	[assignment: access control SFP, information flow control SFP]	Restricted Mode Access Control SFP
	[selection, choose one of: restrictive, permissive, [assignment: other property]]	restrictive [none]
FMT_MSA.3.2/ RM	[assignment: the authorized identified roles]	none
FMT_SMR.1/ RM	**Security Roles**	
FMT_ SMR.1.1/RM	[assignment: the authorized identified roles]	S.ACAdmin

Table 53. Restricted Mode (RM) SFRs *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FMT_SMR.1.2/RM	No operation	
FMT_SMF.1/RM	**Specification of Management Functions**	
FMT_SMF.1.1/RM	[assignment: list of management functions to be provided by the TSF]	• 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**	
FIA_UID.1.1/RM	[assignment: list of TSF-mediated actions]	• select ISD • identify the card • query the debug logging information • send Restricted Mode Unlock Request
FIA_UID.1.2/RM	No Operation	
FIA_UAU.1/RM	**Timing of Authentication**	
FIA_UAU.1.1/RM	[assignment: list of TSF mediated actions]	• OP.TRIGGER_UPDATE • select ISD • identify the card • query the debug logging information • send Restricted Mode Unlock Request
FIA_UAU.1.2/RM	No Operation	

7.1.1.7.4 Context Seperation

The SFRs in this section provide additional proprietary features, as allowed by the PP [14], and add functionality to the TOE making the SFRs more restrictive than the PP alone.

Table 54. Context Seperation SFRs

SFR ID	Modified
FDP_ACC.2/CONTSEP	FDP_ACC.2/CONTSEP
FDP_ACF.1/CONTSEP	FDP_ACF.1/CONTSEP
FMT_MSA.1/CONTSEP	FMT_MSA.1/CONTSEP
FMT_MSA.3/CONTSEP	FMT_MSA.3/CONTSEP
FMT_SMF.1/CONTSEP	FMT_SMF.1/CONTSEP
FMT_SMR.1/CONTSEP	FMT_SMR.1/CONTSEP
FIA_UID.1/CONTSEP	FIA_UID.1/CONTSEP

Table 55. Context Seperation (CONTSEP) SFRs

SFR ID	Selection / Assignment text	Selection / Assignment value
FDP_ACC.2/ CONTSEP	**Complete Access Control**	
FDP_ACC.2.1/ CONTSEP	[assignment: access control SFP]	Context Seperationl SFP
	[assignment: list of subjects and objects]	Subjects: S.MainJCOP, S.GuestOS Objects: O.MainJCOP_Memory_Region, O.GuestOS_Memory_Region
FDP_ACC.2.2/ CONTSEP	No Operation	
FDP_ACF.1/ CONTSEP	**Security Attribute based Access Control**	
FDP_ACF.1.1/ CONTSEP	[assignment: access control SFP	Context Seperation SFP
	[assignment: 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]	Subjects: S.MainJCOP, S.GuestOS Objects: O.MainJCOP_Memory_Region, O.GuestOS_Memory_Region Security Attributes: Access Control Table
FDP_ACF.1.2/ CONTSEP	[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	• 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.MainJCOP according to the Access Control Table. • S.GuestOS cannot perform operation OP.CONT_ACCESS over O.MainJCOP_ Memory_Region. • S.MainJCOP can perform operation OP.CONT_ACCESS over its own O.MainJCOP_Memory_Region. • S.MainJCOP can perform operation OP.CONT_ACCESS over O.GuestOS_ Memory_Region only if so authorized in Access Control Table.
FDP_ACF.1.3/ CONTSEP	[assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects]	S.GuestOS can perform operation OP.CONT_ ACCESS over O.MainJCOP_Memory_Region only through dedicated call gates mechanism.
FDP_ACF.1.4/ CONTSEP	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	none
FMT_MSA.1/ CONTSEP	**Management of Security Attributes**	

Table 55. Context Seperation (CONTSEP) SFRs *...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
FMT_MSA.1.1/ CONTSEP	[assignment: access control SFP(s), information flow control SFP(s)]	Context Seperation Access Control SFP
	[selection: change_default, query, modify, delete, [assignment: other operations]	modify [none]
	[assignment: list of security attributes]	Access Control Table
	[assignment: the authorized identified roles]	S.MainJCOP
FMT_MSA.3/ CONTSEP	**Static Attribute Initialisation**	
FMT_MSA.3.1/ CONTSEP	[assignment: access control SFP, information flow control SFP]	Context Seperation Access Control SFP
	[selection, choose one of: restrictive, permissive, [assignment: other property]]	restrictive [none]
FMT_MSA.3.2/ CONTSEP	[assignment: the authorized identified roles]	S.MainJCOP
FMT_SMR.1/ CONTSEP	**Security Roles**	
FMT_ SMR.1.1/ CONTSEP	[assignment: the authorized identified roles]	• One S.MainJCOP running in O.MainJCOP_ Memory_Region • Several S.GuestOS running in their own O.GuestOS_Memory_Region
FMT_ SMR.1.2/ CONTSEP	No operation	
FMT_SMF.1/ CONTSEP	**Specification of Management Functions**	
FMT_SMF.1.1/ CONTSEP	[assignment: list of management functions to be provided by the TSF]	OP.Modification_Of_Access_Control_Table
FIA_UID.1/ CONTSEP	**Timing of Identification**	
FIA_UID.1.1/ CONTSEP	[assignment: list of TSF-mediated actions]	No Action
FIA_UID.1.2/ CONTSEP	No Operation	

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.
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.

SFR	Rationale
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.
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.

SFR	Rationale
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.
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.

SFR	Rationale
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.

SFR	Rationale
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.
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.
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.

SFR	Rationale
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.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.

SFR	Rationale
FDP_RIP.1/Abort	Contributes to meet the objective.
FDP_RIP.1/APDU	Contributes to meet the objective.
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.

SFR	Rationale
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.
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.
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.

SFR	Rationale
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.

SFR	Rationale
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 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.
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.
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.
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.
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 Main JCOP to create new entries.
FMT_MSA.1/ CONTSEP	Contributes to cover the objective by allowing only Main JCOP to modify the Memory Region Access Control Table.
FMT_SMR.1/ CONTSEP	Contributes to cover the objective by maintaining the roles S.Main JCOP, 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 Main JCOP 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 Main JCOP to create new entries.
FMT_MSA.1/ CONTSEP	Contributes to cover the objective by allowing only Main JCOP to modify the Memory Region Access Control Table.
FMT_SMR.1/ CONTSEP	Contributes to cover the objective by maintaining the roles S.MainJCOP, 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 Main JCOP 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 Main JCOP to create new entries.
FMT_MSA.1/ CONTSEP	Contributes to cover the objective by allowing only Main JCOP to modify the Memory Region Access Control Table.
FMT_SMR.1/ CONTSEP	Contributes to cover the objective by maintaining the roles S.MainJCOP, 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 56. SFRs Dependencies.

Requirements	CC Dependencies	Satisfied dependencies
FAU_ARP.1	FAU_SAA.1 Potential violation analysis	see §7.3.3.1 of [14]
FAU_SAS.1/SCP	No Dependencies	
FCO_NRO.2/SC	FIA_UID.1 Timing of identification	FIA_UID.1/SC

Table 56. 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 [14]
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.6 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.6
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] FCS_CKM.6 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.6
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 or FCS_CKM.5 Cryptographic key derivation]	see §7.3.3.1 of [14]
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 Timing and event of Cryptographic key destruction.	see §7.3.3.1 of [14]
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

Table 56. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
FDP_ACC.2/FIREWALL	FDP_ACF.1 Security attribute based access control	see §7.3.3.1 of [14]
FDP_ACC.2/ADEL	FDP_ACF.1 Security attribute based access control	see §7.3.3.1 of [14]
FDP_ACC.2/RM	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/RM
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 [14]
FDP_ACF.1/ADEL	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	see §7.3.3.1 of [14]
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 [14]
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 [14]
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

Table 56. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
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
FDP_RIP.1/OBJECTS	No 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 [14]
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	

Table 56. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied 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 [14]
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	
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 [14]
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 [14]
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 [14]
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

Table 56. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
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
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 [14]
FMT_MSA.3/FIREWALL	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [14]
FMT_MSA.3/JCVM	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [14]
FMT_MSA.3/ADEL	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [14]
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

Table 56. SFRs Dependencies*....continued*

Requirements	CC Dependencies	Satisfied dependencies
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 [14]
FMT_MTD.3/JCRE	FMT_MTD.1 Management of TSF data	see §7.3.3.1 of [14]
FMT_SMF.1	No dependencies	
FMT_SMF.1/ADEL	No dependencies	
FMT_SMF.1/OSU	No 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 [14]
FMT_SMR.1/INSTALLER	FIA_UID.1 Timing of identification	see §7.3.3.1 of [14]
FMT_SMR.1/ADEL	FIA_UID.1 Timing of identification	see §7.3.3.1 of [14]
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	

Table 56. SFRs Dependencies...*continued*

Requirements	CC Dependencies	Satisfied dependencies
FPT_TDC.1	No dependencies	
FPT_RCV.3/INSTALLER	AGD_OPE.1 Operational user guidance	see §7.3.3.1 of [14]
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 CSP

7.2.1 CSP Security Functional Requirements

The Security Functional Requirements for the CSP component of the TOE are defined in strict compliance with the Security Problem Definition described in the CSP PP [15].

Table 57 provides the selection and assignments for CSP specific SFRs. Note that all Crypto and RNG functions are provided by the JCOP platform. Use of the CSP API restricts the user to the limits imposed by the CSP PP , as the platform functionality exceeds those limits, all CSP SFRs have been prefixed with CSP to avoid conflict .

Table 57. CSP related SFRs

SFR ID	Selection / Assignment text	Selection / Assignment value
CSP. FDP_ACC.1.1/KM	(1) [selection: Administrator, Crypto-Officer]	Administrator

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
CSP.FMT_MSA.1.1/KM	(1) [selection: Administrator, Crypto-Officer] (4) [selection: Administrator, Crypto-Officer] (5) [selection: Administrator, Crypto-Officer, Key Owner]	Administrator Administrator Administrator, Key Owner
CSP.FMT_MSA.3.2/KM	[selection: Administrator, Crypto-Officer]	Administrator
CSP.FMT_MTD.1.1/KM	(1) [selection: Administrator, Crypto-Officer, Key Owner]]	Administrator, Key Owner
	(2) [selection: Administrator, Crypto-Officer]	Administrator
	(3) [selection: Administrator, Crypto-Officer, Key Owner]	Administrator, Key Owner
	(4) [selection: Administrator, Crypto-Officer, Key Owner]	Administrator, Key Owner
CSP.FMT_MTD.1.1/RK	(1) [selection: Administrator, Crypto-Officer] (2) [selection: Administrator, Crypto-Officer]	Administrator Administrator
CSP.FCS_CKM.1.1/AES	[selection: 256 bits, no other key size]	256 bits
CSP.FCS_CKM.5.1/AES	[assignment: key type]	AES Key
	[assignment: input parameters]	derivation data
	[assignment: key derivation algorithm]	AES key Generation using bit string derived from input parameters with KDF
	[assignment: list of key sizes]	256 bits
	[assignment: list of standards]	NIST SP800-56C [39]
CSP.FCS_CKM.1.1/ECC	[selection: elliptic curves in the table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • brainpoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[selection: key size in the table 2 ([15])]	• 256-bit • 384-bit • 521-bit • 256-bit • 384-bit • 521-bit
	[selection: standards in the table 2 ([15])]	• RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • FIPS PUB 186-4 B.4 and D.1.2.3 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.4 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.5 [FIPS PUB 186-4]
CSP.FCS_CKM.5.1/ECC	[assignment: input parameters]	derivation data, at least equal in byte length to the requested keysize.
	[selection: elliptic curves in table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • brainpoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521
	[assignment: KDF]	CSP KDF
	[selection: key size in the table 2 ([15])]	• 256-bit • 384-bit • 521-bit • 256-bit • 384-bit • 521-bit
	[selection: standards in the table 2 ([15])]	• RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • FIPS PUB 186-4 B.4 and D.1.2.3 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.4 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.5 [FIPS PUB 186-4]
CSP.FCS_CKM.1.1/ RSA(CSP)	[assignment: cryptographic key sizes]	from 2000 bit, up to 4096 bit in 8 bit steps

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
CSP.FCS_CKM.5.1/ ECDHE	[selection: AES-256, none other]	AES-256
	[selection: elliptic curves in table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • BRainPoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521
	[selection: DH group in table 3 ([15])]	• 256-bit random ECP group • 384-bit random ECP group • 521-bit random ECP group • brainpoolP256r1 • brainpoolP384r1 • BRainPoolP512r1
	[assignment: key derivation function]	ANSI X9.63 key derivation function
	[selection:256 bits, none other]	256-bit
CSP.FCS_CKM.1.1/ ECKA-EG	[selection: elliptic curves in table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • BRainPoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521
	[selection: key size in the table 2 ([15])]	• 256-bit • 384-bit • 521-bit • 256-bit • 384-bit • 521-bit
	[selection: standards in the table 2 ([15])]	• RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • FIPS PUB 186-4 B.4 and D.1.2.3 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.4 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.5 [FIPS PUB 186-4]
CSP.FCS_CKM.5.1/ ECKA-EG	[selection: AES-256, none other]	AES-256

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[selection: elliptic curves in table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • BRainPoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521
	[selection:256 bits, none other]	256 bits
CSP.FCS_CKM.1.1/ AES_RSA	[selection: AES-256, none other]	AES-256
	[selection:256 bits, none other]	256 bits
CSP.FCS_CKM.5.1/ AES_RSA	[selection: AES-256, none other]	AES-256
	[selection:256 bits, none other]	256 bits
CSP.FCS_COP.1.1/KW	[selection: KW, KWP]	KWP
	[selection:256 bits, none other]	256 bits
CSP.FCS_COP.1.1/KU	[selection: KW, KWP]	KWP
	[selection:256 bits, none other]	256 bits
CSP.FPT_ISA.1.5/CK	(2) [assignment: additional importation control rules]	none
CSP.FPT_ESA.1.4/CK	[assignment: additional exportation control rules]	none
CSP.FCS_COP.1.1/ED	[selection: AES-256, no other algorithm]	AES-256
	[selection: CRT, OFB, CFB, no other]	CRT, OFB, CFB
	[selection: 256 bits, no other key size]	256 bits
CSP.FCS_COP.1.1/HEM	[selection: FCS_CKM.1/ ECKA-EG, FCS_CKM.1/ AES_RSA, FCS_CKM.5/ ECDHE]	FCS_CKM.1/ECKA-EG, FCS_CKM.1/ AES_RSA
	[selection: AES-256, none other]	AES-256
	[selection: CBC [36], GCM [38]]	CBC [36], GCM [38]
	[selection: CMAC [37], GMAC [38], HMAC [45]]	CMAC [37], GMAC [38], HMAC [45]
	[selection: 256 bits, no other key size]	256 bits

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
CSP.FCS_COP.1.1/HDM	[selection: FCS_CKM.5/ ECDHE, FCS_CKM.5/ ECKA-EG, FCS_CKM.5/ AES_RSA]	FCS_CKM.5/ECKA-EG, FCS_CKM.5/ AES_RSA
	[selection: CMAC [37], GCM [38], HMAC [45]]	CMAC [37], GCM [38], HMAC [45]
	[selection: AES-256, none other]	AES-256
	[selection: CBC [36], GMAC [38]]	CBC [36], GMAC [38]
	[assignment: cryptographic key sizes]	256 bits
CSP.FCS_COP.1.1/MAC	[selection: AES-256, none other]	AES-256
	[selection: GMAC [38], no other]	GMAC [38]
	[selection: 256 bits, no other key size]	256 bits
CSP.FCS_COP.1.1/HMAC	[selection: HMAC-SHA-1, HMAC-SHA384, no other]	HMAC-SHA-1, HMAC-SHA384
	[assignment: cryptographic key sizes]	from 128 bit to 896 bit in 8 bit steps
CSP.FCS_COP.1.1/CDS-ECDSA	[selection: elliptic curves in the table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • BRainPoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521
	[selection: key size in the table 2 ([15])]	• 256-bit • 384-bit • 521-bit • 256-bit • 384-bit • 521-bit
	[selection: standards in the table 2 ([15])]	• RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • FIPS PUB 186-4 B.4 and D.1.2.3 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.4 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.5 [FIPS PUB 186-4]

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
CSP.FCS_COP.1.1/VDS-ECDSA	[selection: elliptic curves in the table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • BRainPoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521
	[selection: key size in the table 2 ([15])]	• 256-bit • 384-bit • 521-bit • 256-bit • 384-bit • 521-bit
	[selection: standards in the table 2 ([15])]	• RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • RFC5639 [RFC5639], TR-03111, section 4.1.3 [TR-03111] • FIPS PUB 186-4 B.4 and D.1.2.3 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.4 [FIPS PUB 186-4] • FIPS PUB 186-4 B.4 and D.1.2.5 [FIPS PUB 186-4]
CSP.FCS_COP.1.1/CDS-RSA	[assignment: cryptographic key sizes]	2000 bit up to 4096 bit in 8 bit steps
CSP.FCS_COP.1.1/VDS-RSA	[assignment: cryptographic key sizes]	2000 bit up to 4096 bit in 8 bit steps
CSP.FDP_DAU.2.1/Sig	[selection: FCS_COP.1/CDS-RSA, FCS_COP.1/CDS-ECDSA]	FCS_COP.1/CDS-RSA, FCS_COP.1/CDS-ECDSA
CSP.FDP_DAU.2.1/Att	[selection: FCS_COP.1/CDS-RSA, FCS_COP.1/CDS-ECDSA, ECDAA according to [selection: [50], [51]], [assignment: other cryptographic authentication mechanism]]	FCS_COP.1/CDS-RSA, FCS_COP.1/CDS-ECDSA according to respectively RSA and ECDSA digital signature mechanisms
CSP.FTP_ITC.1.1/CSP	[selection: logically separated from other communication channels, using physical separated ports]	logically separated from other communication channels
	[selection: Authentication of TOE and remote entity according to the case in table 4] ([15])	• FIA_API.1/PACE, FIA_UAU.5.1 (2) • FIA_API.1/CA, FIA_UAU.5.1 (4) or (5), and (6)

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	[assignment: according to the case in table 4] ([15]) [selection: cryptographic operation according to the case in table 4] ([15])	• modfication, disclosure • modfication, disclosure • FCS_COP.1/TCM • FCS_COP.1/TCE
CSP.FCS_CKM.1.1/PACE	[selection: elliptic curves in the table 2 ([15])]	• brainpoolP256r1 • brainpoolP384r1 • BRainPoolP512r1 • Curve P-256 • Curve P-384 • Curve P-521
	[selection: 128 bits, 192 bits, 256 bits]	128 bits, 192 bits, 256 bits
CSP.FCS_CKM.1.1/TCAP	[selection: 128 bits, 192 bits, 256 bits]	128 bits, 192 bits, 256 bits
CSP.FCS_COP.1.1/TCE	[selection: CBC [36], GCM [38]]	CBC [36], GCM [38]
	[selection: 128 bits, 192 bits, 256 bits]	128 bits, 192 bits, 256 bits
CSP.FCS_COP.1.1/TCM	[selection: CMAC [37], GMAC [38]]	CMAC [37], GMAC [38]
	[selection: 128 bits, 192 bits, 256 bits]	128 bits, 192 bits, 256 bits
CSP.FMT_MTD.1.1/RAD	(1) [selection: Administrator, User Administrator]	Administrator
	(2) [selection: Administrator, User Administrator]	Administrator
	(3) no operation	n/a
	(4) [selection: Administrator, User Administrator]	Administrator
	(5) [assignment: time frame]	time frame chosen by the Administrator, by closing SCP, or enters DPD
	(5) [selection: Administrator, User Administrator]	User Administrator
	(6) [selection: Unidentified user, Unauthenticated user]	Unauthenticated user
	(6) [selection: Administrator, User Administrator]	Administrator

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
CSP.FIA_AFL.1.1/CSP	[selection: [assignment: positive integer number]number], an [selection: Administrator, User Administrator] configurable positive integer within [assignment: range of acceptable values]]	Administrator configurable positive integer within a range of values determined by this administrator
	[assignment: list of authentication events]	consecutive failed authentication attempt
CSP.FIA_AFL.1.2/CSP	[assignment: met, surpassed]	met
	[assignment: list of actions]	explicitly delete the password
CSP.FMT_SAE.1.1/CSP	[selection: Administrator, User Administrator]	User Administrator
CSP.FIA_UID.1.1/CSP	(3) [assignment: list of other TSF-mediated actions]	none
CSP.FIA_UAU.1.1/CSP	(3) [selection: a role, a set of role]	a role
	(4) [assignment: list of other TSF-mediated actions]	none
CSP.FIA_UAU.5.2/CSP	(7) [assignment: additional rules]	none
CSP.FIA_UAU.6.1/CSP	(4) [assignment: list of other conditions under which re-authentication is required]	none
CSP.FDP_ACC.1.1/Oper	(1) [selection: Administrator, Crypto-Officer]	Administrator
	(1) [assignment: other roles]	none
CSP.FDP_ACF.1.1/Oper	(1) [selection: Administrator, Crypto-Officer]	Administrator
	(1) [assignment: other roles]	None
CSP.FDP_ACF.1.2/Oper	(1) [selection: Administrator, Crypto-Officer]	Administrator

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
	(3) [assignment: other rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]	none
CSP.FDP_ACF.1.3/Oper	(1 - Table 5) [selection: Administrator, Crypto-Officer, Key Owner]	Administrator, Key Owner
	(1 - Table 5) [selection: Administrator, Crypto-Officer, Key Owner]	Administrator, Key Owner
	(2) [assignment: additional rules, based on security attributes, that explicitly authorise access of subjects to objects]	none
CSP.FDP_ACF.1.4/Oper	(3) [assignment: additional rules, based on security attributes, that explicitly deny access of subjects to objects]	none
CSP.FMT_SMF.1.1/CSP	(4) [assignment: additional list of security management functions to be provided by the TSF]	none
CSP.FMT_SMR.1.1/CSP	[selection: Administrator, Crypto-Officer, User Administrator, Update Agent]	Administrator, User Administrator, Update Agent
	[selection: [assignment: other roles], no other roles]	no other roles
CSP.FMT_MSA.2.1/CSP	(4) [assignment: additional security attributes]	none
CSP.FMT_MOF.1.1/CSP	(1) [selection: Administrator, User Administrator]	User Administrator
	(2) [selection: Administrator, User Administrator]	User Administrator
	(3) [selection: Administrator, User Administrator]	User Administrator
	(4) [selection: Administrator, User Administrator]	User Administrator
CSP.FDP_SDC.1.1/CSP	[assignment: memory area]	NVM

Table 57. CSP related SFRs*...continued*

SFR ID	Selection / Assignment text	Selection / Assignment value
CSP.FPT_TST.1.1/CSP	[assignment: parts of TSF]	the TSF
CSP.FCS_COP.1.1/ VDSUCP	[assignment: cryptographic algorithm]	ECDSA with NIST P-256, Brainpool P256r1
	[assignment: cryptographic key sizes]	256 bits
	[assignment: list of standards]	FIPS 186-4 [52]
CSP.FCS_COP.1.1/ [DecUCP	[assignment: cryptographic algorithm]	AES-128 in CBC mode
	[assignment: cryptographic key sizes]	128 bits
	[assignment: list of standards]	[36]
CSP.FCS_RNG.1	See FCS_RNG.1	
CSP.FDP_ACC.1.1/UCP	[selection: Administrator, Update Agent]	Update Agent
CSP.FDP_ACF.1.1/UCP	[selection: Administrator, Update Agent]	Update Agent
CSP.FDP_ACF.1.2/UCP	(1) [selection: Administrator, Update Agent]	Update Agent
	(2) [selection: Administrator, Update Agent]	Update Agent
	(2) (b) the Version Number of the Update Code Package is equal or higher than the Version Number of the TSF	
CSP.FDP_ACF.1.3/UCP	[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]	None
CSP.FDP_ACF.1.4/UCP	[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]	None

7.2.2 CSP Rationale

The rationale of Security Functional Requirements for the CSP component is strictly the same as in the CSP PP [15].

7.2.3 CSP Dependencies Analysis

The Security Functional Requirements dependencies for the the CSP component are the same in the CSP PP [15] with only the following differences:

- FCS_COP.1/VDSUCP: the import of UCP signature verification key is done during manufacturing.
- FCS_COP.1.1/DecUCP: the import of UCP decryption key is done during manufacturing.

8 Security Assurance Requirements (ASE_REQ)

8.1 Security Assurance Requirements

The assurance requirements of this evaluation are EAL5, augmented by AVA_VAN.5, ALC_DVS.2, ASE_TSS.2, and 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

This Security Target augments from EAL4 to EAL5 in order to meet increasing assurance expectations on the resistance to attackers with high attack potential.

This Security Target augments EAL5 with ALC_FLR.2 to cover policies and procedures that are applied to track and correct flaws and to support surveillance of the TOE. Furthermore, ASE_TSS.2 is chosen to give architectural information on the security functionality of the TOE, which enhances comprehensibility.

The assurance level EAL5 is an elaborated pre-defined level of the CC, part 3 [1]. The assurance components in an EAL level are chosen in a way that they build a mutually supportive and complete set of components. The additional requirements chosen for augmentation do not add any dependencies, which are not already fulfilled for the corresponding requirements contained in EAL5. Therefore, the components AVA_VAN.5, ALC_DVS.2, ASE_TSS.2 and ALC_FLR.2 serve additional assurance to EAL5, but the mutual support of the requirements is still guaranteed.

8.3 Dependencies of Security Assurance Requirements

The dependencies of the Security Assurance Requirements are given in Table 58. They are derived from Appendix C of CC [5] . The table indicates whether the SAR is directly or indirectly required. Only applicable dependencies from the highest level assurance components are considered.

Table 58. Dependencies of the Security assurance requirements

Name	Directly required	Indirectly required
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1	ADV_FSP.2
ADV_FSP.5	ADV_IMP.1, ADV_TDS.1	ADV_TDS.3, ALC_TAT.1
ADV_IMP.1	ADV_TDS.3, ALC_CMC.4, ALC_TAT.1	ADV_FSP.4, ALC_CMS.1, ALC_DVS.1, ALC_LCD.1
ADV_INT.2	ADV_IMP.1, ADV_TDS.3, ALC_TAT.1	ADV_FSP.4,
ADV_TDS.4	ADV_FSP.5	ADV_IMP.1, ADV_TDS.3, ALC_TAT.1

Table 58. Dependencies of the Security assurance requirements *...continued*

Name	Directly required	Indirectly required
AGD_OPE.1	ADV_FSP.1	none
AGD_PRE.1	none	none
ALC_CMC.4	ALC_CMS.1, ALC_DVS.1, ALC_LCD.1	none
ALC_CMS.5	none	none
ALC_DEL.1	none	none
ALC_DVS.2	none	none
ALC_FLR.2	none	none
ALC_LCD.1	none	none
ALC_TAT.2	ADV_IMP.1	ADV_FSP.4, ADV_TDS.3
ASE_CCL.1	ASE_ECD.1, ASE_INT.1, ASE_REQ.1	none
ASE_ECD.1	none	none
ASE_INT.1	none	none
ASE_OBJ.2	ASE_SPD.1	none
ASE_REQ.2	ASE_ECD.1, ASE_OBJ.2	ASE_SPD.1
ASE_SPD.1	none	none
ASE_TSS.2	ADV_ARC.1, ASE_INT.1, ASE_REQ.1	ADV_FSP.2, ADV_TDS.1, ASE_ECD.1
ATE_COV.2	ADV_FSP.2, ATE_FUN.1	ADV_TDS.1, ATE_COV.1
ATE_DPT.3	ADV_ARC.1, ADV_TDS.4, ATE_FUN.1	ADV_FSP.5, ADV_IMP.1, ALC_TAT.1, ARE_COV.1
ATE_FUN.1	ATE_COV.1	ADV_FSP.2, ADV_TDS.1, ATE_FUN.1
ATE_IND.2	ADV_FSP.2, AGD_OPE.1, AGD_ PRE.1, ATE_COV.1, ATE_FUN.1	ADV_TDS.1
AVA_VAN.5	ADV_ARC.1, ADV_FSP4, ADV_IMP.1, ADV_TDS.3, AGD_OPE.1, AGD_ PRE.1, ATE_DPT.1	ALC_TAT.1, ATE_COV.1, ATE_FUN.1

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 59 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 59. 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 [19], [18].
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 [21] and GlobalPlatform Amendments A [22],C [23], D [24], E [25] and J [29]. In addition to the GP specification, the Java Card Runtime Environment specification [19] is followed for application loading, installation, and deletion. AID management is provided by SF.OPEN according to the GlobalPlatform Specification [21], the Java Card Runtime Environment Specification [19], and the Java Card API Specification [17]. 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 [17] . 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 [17] 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 JCOP8.9 OS or Updater OS 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 [19]. 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 [19] 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 uage.
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 [17] and the GlobalPlatform Specification [21].

SF.PERS_MEM	**Persistent Memory Management** SF.PERS_MEM provides atomic write operations and transaction management according to the Java Card Runtime Environment Specification [19]. 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 [53]. 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 guarantee code execution, data storage, and hardware virtualization in a completely isolated way from other contexts. . The default context separation will be applied at boot by Main JCOP and only Main JCOP can change the context separation setting after that . One specific Context is reserved for Main JCOP 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 attempt to perform a forbiden access will be prevented and will trigger a security alarm.

9.3 Security Functionality of CSP

SF.CRYPTO_CSP	**CSP specific cryptography** This TSF provides key creation, key management, key deletion and cryptographic functionality specific to the CSP component. It provides the API in accordance to CSP specification [43] to fulfill: FCS_CKM.1/AES, FCS_CKM.5/AES, FCS_CKM.1/ECC, FCS_CKM.5/ECC, FCS_CKM.1/RSA(CSP), FCS_CKM.5[ECDHE], FCS_CKM.1/ECKA-EG, FCS_CKM.5/ECKA-EG, FCS_CKM.1/AES-RSA, FCS_CKM.5/AES-RSA, FCS_CKM.1/PACE, FCS_CKM.1/TCAP, FCS_COP.1/Hash, FCS_COP.1/KW, FCS_COP.1/KU, FCS_COP.1/ED, FCS_COP.1/HEM, FCS_COP.1/HDM, FCS_COP.1/MAC, FCS_COP.1/HMAC, FCS_COP.1/CDS-ECDSA, FCS_COP.1/VDS-ECDSA, FCS_COP.1/CDS-RSA, FCS_COP.1/VDS-RSA, FCS_COP/VDSUCP, FCS_COP/DecUCP, FCS_COP.1/TCE, FCS_COP.1/TCM . All those are implemented in the JCOP platform. The specific storage of keys requested by FDP_SDC.1/CSP is covered by AES encryption implemented by the JCOP platform.
SF.ACCESS_CSP	**CSP access protection features** This TSF handles the access to CSP features by external or local users. It bases on Java Card and GlobalPlatform features to implement different access control (FDP_ACC.1/Oper, FDP_ACF.1/Oper, DP_ACC.1/KM) and the conditions realization granting the access: identification/authentication of users (FIA_ATD.1/CSP, FIA_AFL.1/CSP, FIA_USB.1/CSP, FIA_UID.1/CSP, FIA_UAU.1/CSP, FIA_UAU.5/CSP, FIA_UAU.6/CSP, FIA_API.1/CA, FIA_API.1/PACE) and trusted channels establishment (FTP_ITC.1/CSP). Related security attributes and data are also based on Java Card features (FMT_SMF.1/CSP, FMT_SMR.1/CSP, FMT_MOF.1/CSP, FMT_MSA.1/KM, FMT_MSA.2/CSP, FMT_MSA.3/KM, FMT_MTD.1/KM, FMT_MTD.1/RAD, FMT_MTD.1/RK, FMT_MTD.3/CSP, FMT_SAE.1/CSP)
SF.SERVICES_CSP	**CSP other services** This TSF handles other services as generation of data validity evidences (FDP_DAU.2/Att, FDP_DAU.2/Sig), exchanges of cryptographic keys (FPT_TDC.1/CK, FPT_TCT.1/CK, FPT_TIT.1/CK, FPT_ISA.1/CK, FPT_ESA.1/CK), exchanges of certificates (FPT_TDC.1/Cert), FPT_TIT.1/Cert, FPT_ISA.1/Cert) and exchanges of user data (FDP_ETC.1/CSP, FDP_ETC.2/CSP, FDP_ITC.2/UD).
SF.SELF-PROTECTION_CSP	**CSP specific self-protections** This TSF extends the scope of self-protection features provided by the Java Card platform to the CSP component needs (FPT_FLS.1/CSP, FRU_FLT.2/CSP,). This also includes Self-testing which is an ongoing process performed by JCOP (FPT_TST.1/CSP).
SF.UCP_CSP	**CSP specific Update Code Package** This TSF provides support for the management of the Update Code Package feature (FDP_ITC/UCP, FPT_TDC/UCP, FDP_ACC/UCP, FDP_ACF/UCP, FDP_RIP/UCP)

10 Bibliography

10 . 1 Evaluation documents

[1] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, CCMB-2017-04-003, April 2017

[2] Common Criteria Recognition Arrangement, Management Committee Policies and Procedures. Transition Policy to CC:2022 and CEM:2022. CCMC-2023-04-001, 20 April 2023

[3] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 2022, Revision 1, November 2022. CCMB-2022-011-001.

[4] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 2022, Revision 1, November 2022. CCMB-2022-011-002.

[5] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 2022, Revision 1, November 2022. CCMB-2022-011-003.

[6] Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, Version 2022, Revision 1, November 2022. CCMB-2022-011-004.

[7] Common Criteria for Information Technology Security Evaluation, Part 5: Pre-Defined packages of security requirements, Version 2022, Revision 1, November 2022. CCMB-2022-011-005.

[8] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 2022, Revision 1, November 2022. CCMB-2022-011-006.

[9] 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),

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

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

[12] JIL: Composite product evaluation for Smart Cards and similar devices, Joint Interpretation Library, Version 1.6; April 2024.

[13] 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.

[14] 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.

[15] BSI. Common Criteria Protection Profile Cryptographic Service Provider, 19 February 2019 (BSI-CC-PP-0104).

[16] ICAO Doc9303, Machine Readable Travel Documents, 7th Edition, 2015.

10 . 2 Standards

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

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

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

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

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

[22] GlobalPlatform Confidential Card Content Management - Amendment A v1.2, GPC_SPE_007, July 2019.

[23] GlobalPlatform Contactless Services - Amendment C v1.2.1, GPC_SPE_025, Dec 2015.

[24] GlobalPlatform Card Technology Secure Channel Protocol ’03’ - Amendment D v1.2, GPC_SPE_014, April 2020.

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

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

[27] GlobalPlatform Executable Load File Upgrade - Amendment H v1.1, GPC_SPE_120, March 2018.

[28] GlobalPlatform Secure Element Management Service - Amendment I v1.0, GPC_SPE_121, March 2018.

[29] GlobalPlatform Broker Interface - Amendment J v1.1, GPC_SPE_157, February 2022.

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

[31] GlobalPlatform Card Secure Element Configuration - v2.0, GPC_GUI_049, August 2018.

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

[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 7748. Elliptic Curves for Security.

[36] NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation: Methods and Techniques, Morris Dworkin, National Institute of Standards and Technology, December 2001.

[37] NIST SP 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, Morris Dworkin, National Institute of Standards and Technology, May 2005.

[38] National Institute of Standards and USA Technology. NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.

[39] NIST SP 800-56C, Recommendation for Key Derivation methods in Key-Establishment Schemes; Revision 2, 18 August 2020 .

[40] NIST SP 800-108, Recommendation for Key Derivation Using Pseudorandom Functions..

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

[42] 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.
[43] BSI CSP-API Definition, 5 November 2018.
[44] FIPS PUB 186-4-2013: Digital Signature Standard, Federal Information Processing Standards Publication, 2013, July, National Institute of Standards and Technology
[45] RFC 2104: HMAC: Keyed-Hashing for Message Authentication, Request For Comments, February 1997
[46] RFC 5246: The Trasnport Layer Security (TLS) Protocol; Version 1.2
[47] RFC 5869: HMAC-based Extract and Expand Key DerivationFcuntion (HKDF)
[48] RFC 3279: Algorithms and Identifier for the In-ternet X.509 Public Key Infrastructure Certificat and Certificat Revocation List (CRL) Profile
[49] ANSI X9.63: Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography, December 2011, American National Standards Institute

10 . 3 Developer documents

[50] Trusted Platform Module Library, Part 1: Architecture, Family “2.0”, Level 00, Revision 01.38, September 2016.
[51] ECDAAFIDO Alliance, Alliance Proposed Standard FIDO ECDAA Algorithm, https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/fido-ecdaa-algorithm-v1.2-ps-20170411.html, April 2017.
[52] FIPS PUB 186-4: Digital Signature Standard (DSS), US Department of Commerce/ National Institute of Standards and Technology, 2013.
[53] NXP. JCOP 8.9 R6 (SN330) User Guidance manual Rev 1.8.0, 2025-05-08
[54] NXP. JCOP8.9 R6 (SN330) User Guidance Manual Addendum System Management Rev 1.8.0, 2025-05-08
[55] NXP. JCOP-eSE 8.9 R6 (SN330) User Guidance Manual for JCOP eSE Rev 1.8.0, 2025-05-08
[56] NXP. JCOP-eSE 8.9 R6 (SN330) User Guidance Manual Addendum for JCOP eSE Rev 1.8.0, 2025-05-08
[57] NXP. JCOP8.9 CSP User Manual Addendum Rev 1.8.0, 2025-05-08
[58] NXP. JCOP8.9 Amd I SEMS Application User Manual Addendum Rev 1.8.0, 2025-05-08
[59] NXP. JCOP 8.9 R6 (SN330) Errata Sheet Rev 1.8.0, 2025-05-08
[60] NXP SN330 Series - Secure Element Security Target, v1.3 10-Jan-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: 6 August 2025
Document identifier: NXP-ST01-JCOP-eSE_8.9_R6.01