JCOP 4.5 P71

Security Target Lite for JCOP 4.5 P71

Rev. 2.6 — 11 December 2023 **Evaluation document**

NSCIB-CC-2300127-01

Document information

Information	Content
Keywords	ASE, JCOP, Common Criteria, EAL6 augmented
Abstract	This document contains information to fulfill the requirements of the Common Criteria component ASE (Security Target) for the Evaluation of the JCOP 4.5 P71 developed and provided by NXP Semiconductors, Business Line Connectivity & Security, according to the Common Criteria for Information Technology Security Evaluation Version 3.1 at EAL5 augmented

Revision history

Rev	Date	Description
1.0	12 November 2021	First Release
1.1	03 December 2021	Typographical corrections
1.2	13 January 2022	Address Certifier comments, add reference to USIM PP for card content management
1.3	10 Feb 2022	Add claims related to PACE
1.4	10 June 2022	Update UGM Reference and TOE Identification
1.5	27 Oct 2022	Revise Section 1.3.4. Update IC cert: BSI-DSZ-CC-1149-2022-MA-01
1.6	09 Dec 2022	Add Appnote FCS_RNG.1 to clarify AES mode is used by the TOE
1.7	15 Dec 2022	Refinement of FCS_RNG.1 related AppNotes
1.8	12 April 2023	Recertification of the underlying hardware platform
1.9	24 July 2023	Updated table 2
2.0	22 September 2023	Updated certification ID of IC and TOE
2.1	25 September 2023	Updated application notes in section 7.2.1.2.4 and 7.2.1.2.5
2.2	20 October 2023	Updated certificate id
2.3	25 October 2023	Updated FCS_RNG.1 wording
2.4	3 November 2023	Corrected references to UGM and HW ST
2.5	17 November 2023	Corrected certificate id
2.6	11 December 2023	Updated reference to HW ST

1 ST Introduction (ASE_INT)

1.1 ST Reference and TOE Reference

Table 1. ST Reference and TOE reference

Title	JCOP 4.5 P71 Security Target Lite
Version	Revision 2.6
Date	11 December 2023
Product Type	Java Card
TOE name	JCOP 4.5 P71
Certification ID	NSCIB-CC-2300127-01
CC version	Common Criteria for Information Technology Security Evaluation Version 3.1, Revision 5, April 2017 (Part 1 [1], Part 2 [2] and Part 3 [3])

1.2 TOE Overview

1.2.1 TOE Components

The TOE is a composite product consisting of a certified Micro Controller and a software stack which is stored on the Micro Controller and which can be executed by the Micro Controller. The TOE uses one or more communication interfaces to communicate with its environment.

The complete TOE is depicted in Figure 1.

There are 2 variants of the product, each with their own Platform Build ID (PID) supporting different Communication Interfaces

- Variant 1 Supports UART, TCL and T1I2C Interfaces
- Variant 2 supports TCL and T1SPI

The TOE elements are described in more detail in Section 1.3 "TOE Description".

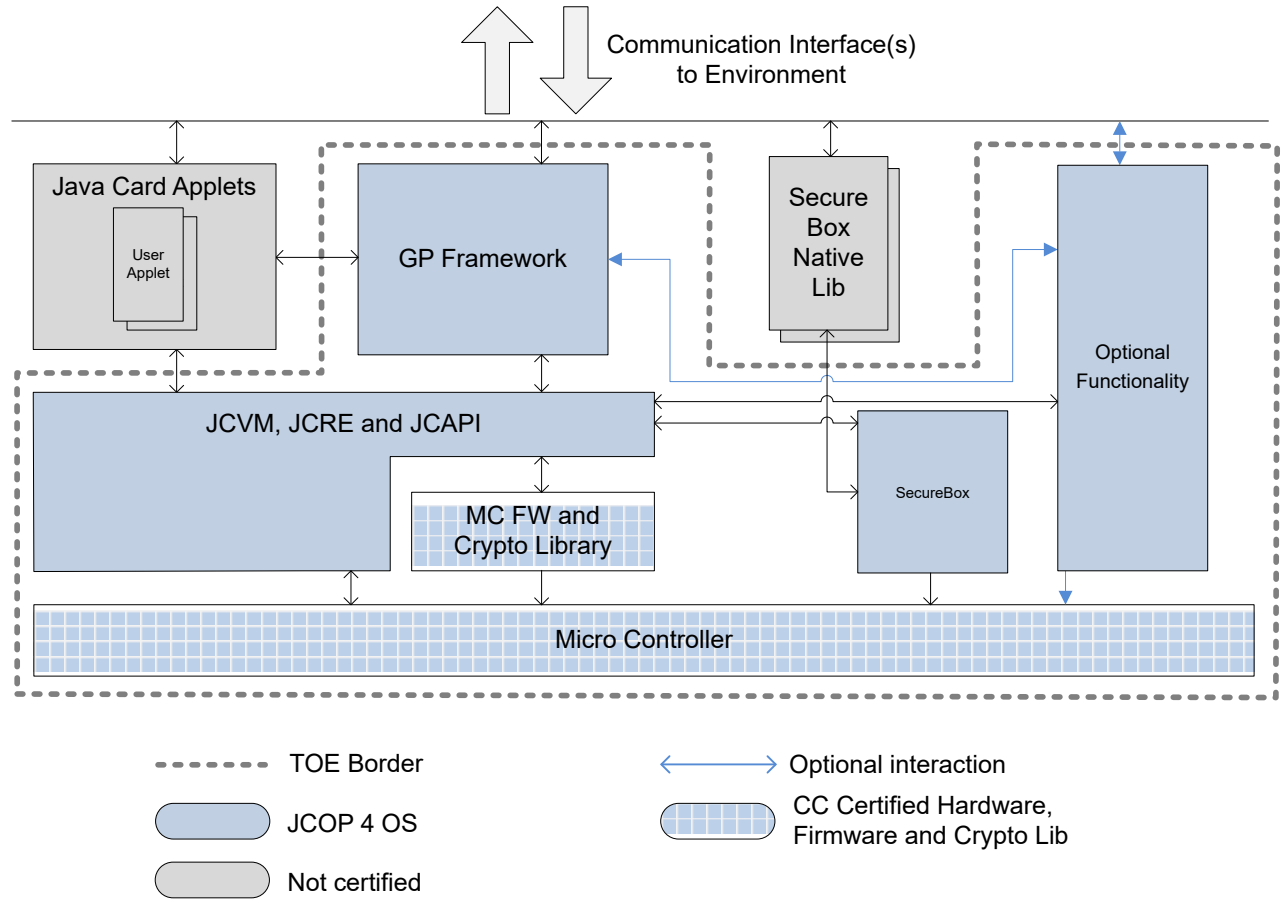


Figure 1. Components of the TOE

Figure 1 shows the components of the TOE. Part of the TOE are the JCVM, JCRE, JCAPI and the GP Framework. Also included is optional functionality and the Secure Box mechanism . Secure Box Native Libraries provide native functions for untrusted third parties and are not part of the TOE .

The figure shows Java Card applets which are small programs in Java language that can be executed by the TOE, but are not part of the TOE.

1.2.2 JCOP components

The software stack can be split into the following components:

- Firmware for booting and low level functionality of the Micro Controller (MC FW) like writing to flash memory - This is included in the hardware Certification
- Software for implementing cryptographic operations, called Crypto Library - This is included in the hardware Certification
- Software to update JCOP 4.5 OS or UpdaterOS, called OS Update Component
- Software to implement JCOP4.5 OS:
  - Software that implements low level functionality, called Native OS
  - Software for implementing a Java Card Virtual Machine [17] called JCVM.
  - Software for implementing a Java Card Runtime Environment [18], called JCRE

- Software for implementing a Java Card Application Programming Interface [16], called JCAPI
- Software for implementing content management according to GlobalPlatform [21], called GlobalPlatform (GP) Framework including support for
  - GP AMD-H : Executable Load File Upgrade [22]
  - GP AMD-I : Secure Element Management Services (SEMS) [1] [23]
- Software that implements a proprietary programming interface, called Extension API
- Software that handles personalization and configuration, callled Config Applet
- Software that implements the API and functionality for MiFare - no security claims are made on MiFare
- Software for executing native libraries, called Secure Box.
  - Provides the possibility for 3rd party native libraries to be executed in a securely encapsulated environment
  - Supports the installation of a choice of vendor specific Biometric libraries, for Match on Card, in production phase.

1.2.3 Usage and Major Security Features of the TOE

The usage of the TOE is focused on security critical applications in small form factors. One main usage scenario is the use of so called smart cards. Examples of such cards are banking cards or electronic drivers' licenses. The TOE can also be used in an electronic passport. Another usage scenario is device authentication, where the TOE can be used to prove the authenticity or originality of a device like an accessory for a gaming console.

The TOE provides a variety of security features. The hardware of the Micro Controller already protects against logical and physical attacks by applying various sensors to detect manipulations and by processing data in ways which protect against leakage of data by side channel analysis. With the software stack the TOE provides many cryptographic primitives for encryption and decryption of data but also for signing and signature verification. Also the software stack contains security features to protect against attacks.

The following list contains the features of this TOE:

- Supported communication protocols:
  - ISO 7816 T=0[39] [2]
  - ISO 7816 T=1[39] [2]
  - ISO 14443 T=CL[40]
  - T1I2C - T1 Over I2C and Global Platform APDU transport over SPI/I2C[25] or NXP legacy UM 11225[42], according to configuration [2]
  - I2C Master [41][2]
  - T1SPI - T1 Over SPI and Global Platform APDU transport over SPI/I2C[25] [3]
- Cryptographic algorithms and functionality:
  - Data Encryption Standard with 3 keys (3DES) for en-/decryption (CBC and ECB) and MAC generation and verification (Retail-MAC, CMAC and CBC-MAC).

1 Partial implemtation
2 Variant 1 only
3 Variant 2 only

  - Advanced Encryption Standard (AES) for en-/decryption (CBC, ECB and counter mode) and MAC generation and verification (CMAC, CBC-MAC).
  - Rivest Shamir Adleman asymmetric algorithm (RSA) and RSA CRT for en-/decryption and signature generation and verification.
  - Modular and ECC point arithmetic functions not provided by the standard Java Card API
  - RSA and RSA Chinese Remainder Theorem (CRT) key generation[4].
  - Elliptic Curve Cryptography (ECC) over GF(p) for signature generation and verification (ECDSA)[4].
  - ECC over GF(p) key generation[4].
  - Random number generation according to class DRG.3 or DRG.4 of AIS 20 [26].
  - Diffie-Hellman with ECDH and modular exponentiation[4].
  - SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 hash algorithm.
  - Following cryptographic algorithms are part of the TOE but without claims for security functional requirements:
    - KoreanSEED[4].
    - AES in Counter with CBC-MAC mode (AES CCM)[4].
    - Keyed-Hash Message Authentication Code (HMAC)[4].
    - HMAC based Key Derivation Function (HKDF) [35][4].
    - Elliptic Curve Direct Anonymous Attestation (ECDAA) [38][4].
    - Elliptic curve cryptography based on Edwards and Montgomery curves[4].
- Java Card functionality:
  - Executing the Java byte codes which are generated from the Java compiler when Java source code is compiled.
  - 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.
  - Support for Extended Length APDUs.
  - Garbage Collection with memory reclamation and compaction.
  - Persistent Memory Management and Transaction Mechanism.
  - Support for Biometric Templates, including integrated Match on Card (MoC) functionality
- GlobalPlatform functionality:
  - Loading of Java packages.
  - Instantiating applet instances.
  - Removing of Java packages.
  - Removing of applet instances.
  - Issuer Security Domain (ISD), Supplementary Security Domain (SSD).
  - Creating SSDs.
  - Associating applets to Security Domains.
  - Installation of keys.
  - Verification of signatures of signed applets.
  - Verification of signatures for commands.

4 Optional functionality

- CVM Management (Global PIN).
- Secure Channel Protocol (SCP01, SCP02 and SCP03).
- Delegated Management, Data Authentication Pattern (DAP).
- Post-issuance installation and deletion of applets and packages.
- Compliance to several GP configurations.
- Executable Load File Upgrade, GP Amendment H.[22]
- Secure Element Management Service, GP Amendment I [23](partial implementation).

- NXP Proprietary Functionality
  - Proprietary secure messaging accelerator interface for applets which are used for electronic passport as defined by ICAO or electronic driver license[5].
  - Proprietary secure messaging accelerator interface for applets which are used for PIV secure messaging [32][5].
  - Secure Box [5].
  - Java Card APIs for:
    - Data encryption via PUF [14].
    - Data integrity protection with an EDC.
    - Asserting results of sensitive functions.
  - Time representation and counter functionality[5] (No Security claimed).
  - OS Update Component: Proprietary functionality that can update JCOP 4.5 or UpdaterOS.
  - Attack counter based performance restriction and Update Authorized Image (UAI) features, for which no additional security claims are made.

1.2.4 TOE Type

The TOE is a Java Card with a GP Framework. It can be used to load and execute off-card verified Java Card applets.

1.2.5 Required non-TOE Hardware/Software/Firmware

Three groups of users shall be distinguished here.

The first group is the **end-users** group, which uses the TOE with one or more loaded applets in the final form factor like a banking card or an electronic passport. These users only require a communication device to be able to communicate with the TOE. The communication protocol of the TOE is standardized in either

- ISO7816 (T=1, T=0)[39]
- ISO14443 (T=CL)[40]
- T1I2C[25] or [42]
- T1SPI[25]
- I2C Master, UM10204 [41]

The second group of users are **administrators of cards**. They want to configure the card by using the Configuration Module, to install additional applets and to configure and personalise these applets. These users require the same equipment as end-users.

The third group of users wants to develop Java Card applets and execute 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

5 Optional functionality

TOE vendor and comprises elements such as PC development environment, byte code verifier, compiler, linker and debugger.

1.3 TOE Description

1.3.1 TOE Components and Composite Certification

The certification of this TOE is a composite certification. The following sections provide a more detailed description of the components of Figure 1. 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.

1.3.1.1 Micro Controller

The Micro Controller is a secure smart card controller from NXP's SmartMX3 family. The Micro Controller contains a co-processor for symmetric cryptographic operations, supporting DES and AES, as well as an accelerator for asymmetric cryptographic algorithms. The Micro Controller further contains a physical random number generator. The supported memory technologies are volatile (Random Access Memory (RAM)) and non-volatile (Read Only Memory (ROM) and FLASH) memory.

Access to all memory types is controlled by a Memory Management Unit (MMU) which allows to separate and restrict access to parts of the memory.

The Micro Controller has been certified in a previous certification and the results are re-used for this certification. The exact reference to the previous certification is given in the following Table 2 "Reference to Certified Micro Controller":

Table 2. Reference to Certified Micro Controller

Name	NXP Secure Smart Card Controller N7122 with IC Dedicated Software and Crypto Library (R1/R2/R3)
Certification ID	BSI-DSZ-CC-1149-V3-2023
Reference	[13]

1.3.1.2 Security IC Dedicated Software

1.3.1.2.1 MC FW (Micro Controller Firmware)

The Micro Controller Firmware is used for testing of the Micro Controller at production, for booting of the Micro Controller after power-up or after reset, for configuration of communication devices and for writing data to volatile and non-volatile memory.

The MC FW has been certified together with the Micro Controller and the same reference [13] as given for the Micro Controller also apply for the MC FW.

1.3.1.2.2 Crypto Library

The Crypto Library provides implementations for symmetric and asymmetric cryptographic operations, hashing, the generation of hybrid deterministic and hybrid physical random numbers and further functions like secure copy and compare.
Some of the cryptographic algorithms offered by the Crypto Lib are not certified, see Section 1.3.1.4 "Excluded functionality".

The symmetric cryptographic operations comprise the algorithms 3DES and AES and KoreanSEED. These algorithms use the symmetric co-processor of the Micro Controller.

The supported asymmetric cryptographic operations are ECC and RSA. These algorithms use the Public Key Crypto Coprocessor (PKCC) of the Micro Controller for the cryptographic operations.

The Crypto Library has been certified together with the Micro Controller and the same reference [13] as given for the Micro Controller also applies.

1.3.1.3 Security IC Embedded Software

1.3.1.3.1 JCOP 4.5 P71

The OS of the TOE consists of JCVM, JCRE, JCAPI and GP framework. It is implemented according to the Java Card Specification and GlobalPlatform. Additionally it consists of a proprietary API, which is described in the UGM [12].

The TOE can be identified by using the IDENTIFY APDU command (see UGM [12]). This command returns the card identification data, which includes a Platform ID, a Patch ID and other information that allows to identify the content in ROM, FLASH and loaded patches (if any).

The TOE also includes a Configuration Module (see Section 1.3.2 "Optional TOE Functionality") which is used for personalisation and configuration of the TOE. It must be deleted after the personalisation is finished (end of Phase 6 ”Personalisation”) by using the DELETE APDU command. Once the Configuration Module is deleted, it is no longer possible to configure the TOE.

The TOE contains further functionality for integrity protection of user data via an EDC, encryption of user data via PUF [14] and optional functionality as described in Section 1.3.2 "Optional TOE Functionality".

1.3.1.3.2 OS Updater

1.3.1.3.2.1 OS Update Component

The OS Updater Component comprises two sub-components:

- OsSelector (no security claimed): After a hardware reset it provides the functionality to either boot UpdaterOS or JCOP. OsSelector 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.
- UpdaterOS:
  - it handles APDUs to write a new OS (either JCOP 4.5 or UpdaterOS) to flash.
  - it verifies the 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.

The UpdaterOS is a standalone operating system that can only be active when JCOP is not active. Besides the capability to update JCOP 4.5 , UpdaterOS is also capable of updating itself. The UpdaterOS version can be queried by using a SELECT OS Update AID Command (see UGM [12]). UpdaterOS shares parts of the Native OS with JCOP

4.5 OS, e.g.: communication interface, wrapper to Security Software (Flash Services and CryptoLib).

Details of the OS Update delivery process are given in Section 8.6 of UGM [12]

1.3.1.4 Excluded functionality

Customer Secure Box Native Libraries are not part of the TOE. No security functional requirements are claimed on KoreanSEED, AES CCM, HMAC, HKDF, ECDAA, elliptic curve cryptography based on Edwards and Montgomery curves and FIPS self-tests, they are TSF non-interfering.

1.3.2 Optional TOE Functionality

Some dedicated functionality of the TOE as listed below can be removed:

- RSA key generation,
- Elliptic curve cryptographic functionality,
- Korean Seed cryptographic functionality,
- accelerators for eGOV APDUS READ BINARY and READ RECORD, support for PACE protocol Generic mapping
- Accelerator for integrated mapping of the PACE protocol,
- TOE self-tests according to FIPS 140-2 [27],
- PIV secure messaging as defined by NIST.SP800-73-4 [32],
- SecureBox,
- TOE Configuration Module (the TOE Configuration Module has to be deleted at the end of life cycle phase 6) [12],
- AES CCM as defined in the Java Card AEADCipher API [16], HMAC and HKDF cryptographic functionality as defined in the Java Card API [16] and the UGM [12]. Timer functionality as defined in the UGM [12].
- ECDAA [38] and elliptic curve cryptography based on Edwards and Montgomery curves.
- I2C slave protocol and T = 1 over I2C.

1.3.3 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 [7], see Figure 2.

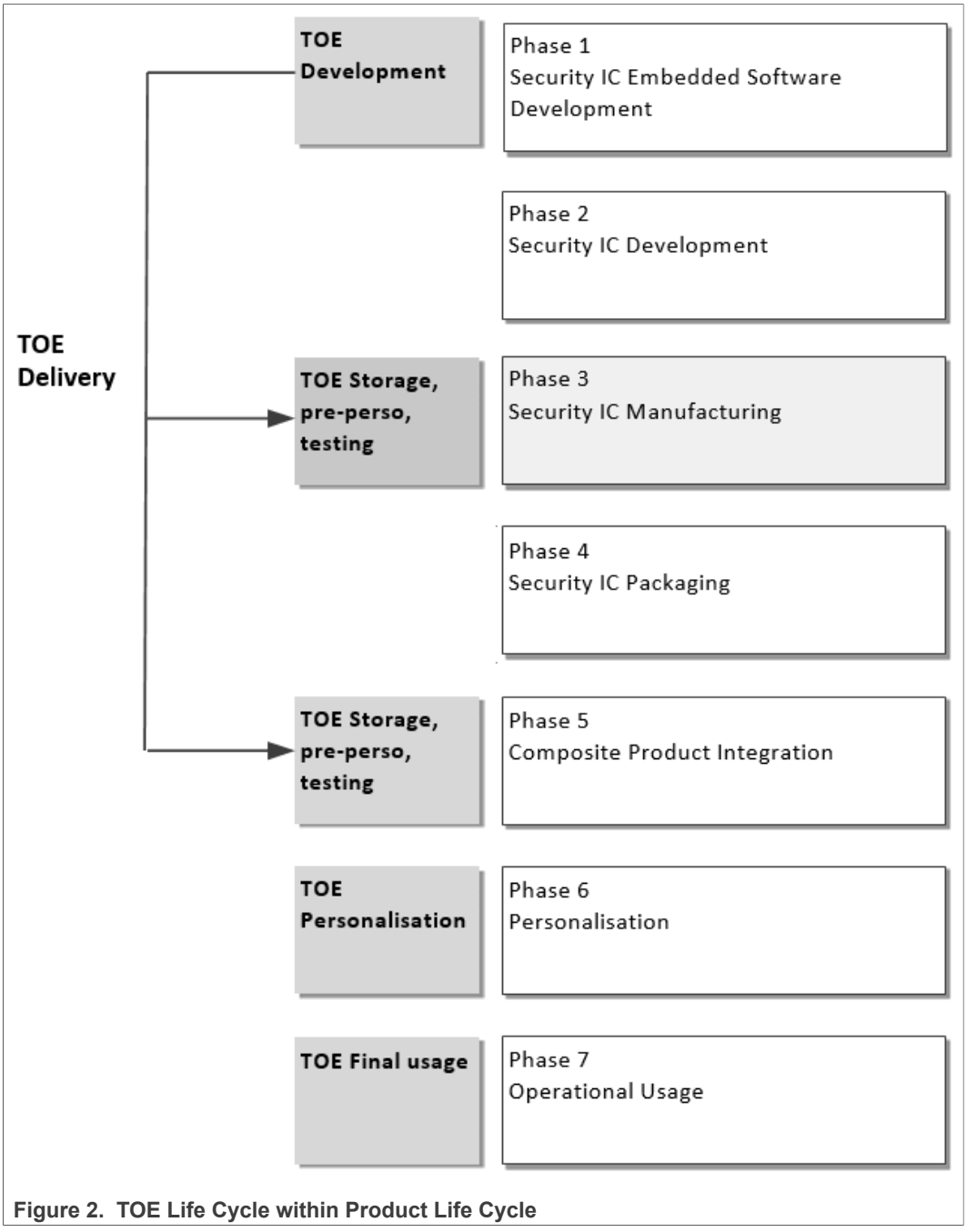


Figure 2. TOE Life Cycle within Product Life Cycle

Table 3. TOE Life Cycle phases

Phase	Name	Description
1	Security IC Embedded Software Development	The Security 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 come from phase 4, 5, or 6.

Table 3. TOE Life Cycle phases*...continued*

Phase	Name	Description
2	Security IC Development	The IC Developer • designs the IC, • develops Security IC Dedicated Software, • provides information, software or tools to the Security IC Embedded Software Developer, and • receives the embedded software from the developer, through trusted delivery and verification procedures. From the IC design, Security IC Dedicated Software and Smart-Card 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 SmartCard IC database. Configuration items may be changed.
4	Security IC Packaging	The IC Packaging Manufacturer is responsible for • IC packaging and testing.
5	Composite Product Integration	The Composite Product Manufacturer is responsible for • SmartCard product finishing process including applet loading and testing. Configuration items may be changed by using the Configuration Module.
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 by using the Configuration Module, which must be deleted at the end of this cycle by using the DELETE APDU command. • Proprietary Secure Box libraries may be loaded on to the Product.
7	Operational Usage	The Consumer 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. • Proprietary Secure Box libraries may be loaded on to the Product.

The evaluation process is limited to phases 1 to 5. User Applet development is outside the scope of this evaluation. Applets can be loaded into FLASH in phases 3, 4, 5, and 6. Applet loading in phase 7 is also allowed. This means post-issuance loading of applets can be done for a certified TOE.

The Configuration Module is loaded into FLASH and has special privileges to personalize and configure the TOE. Before life cycle Phase 7 "Operational Use" the Configuration Module is deleted and hence it is ensured that its functionality cannot be used afterwards. It is possible to load patch code into FLASH in phases 3, 4, 5, and 6. The

certification is only valid for the ROM code having the Platform Identifiers and the Patch IDs (if applicable) as stated in Table 7 "Delivery Items".

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 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 procedures must also be followed in order to ensure that the TOE is correctly and securely handled, and not damaged or comprised. 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).

1.3.4 TOE Identification

The TOE can be identified by using the Platform ID, the Platform Build ID, the OS Core ID, the Patch ID and the firmware configuration identifiers: Wafer Test ID and the Dedicated Firmware Extension.

Platform ID (PID) is read from Tag DF20, using a GET DATA command, see UGM [12] Section 5.1.1.3

Table 4. Platform ID

ID	Tag	Value
Platform ID	DF20	J3R6000373181200

The IDENTIFY command, see [12] Section 5.1.1, can be used to retrieve the Platform Build ID, OS Core ID and Patch ID. The expected response data for this TOE is detailed in the UGM [12] Section 2 and in the table below.

Table 5. IDENTIFY fields

ID	Tag	Value
ROM ID	08	B3375FE9B5508BC4
Platform Build ID	03	Variant 1 - 6D20B6197D635E7C Variant 2 - 5314F0A7BAE6B138
OS Core ID	0A	Variant 1 - 55606FD4BEECF3CD Variant 2 - 318CCEEB284A3AF9
Patch ID	02	0000000000000000

UGM[12] Section 2 details how to verify the certified Firmware Configurations, given in the table below, using the response data from a specified SELECT Command.

Table 6. Certified Firmware Configurations

Identifier	FW Configuration 1	FW Configuration 2
Wafer Test ID	0x0F	0x10
Dedicated firmware extension	0x06	0x07

1.3.4.1 TOE Delivery

The delivery comprises the following items:

Table 7. Delivery Items

Type	Name	Version	Form of Delivery
Hardware	N7122	A1	Micro Controller including on-chip software: Firmware and Crypto Lib[1]
Software	JCOP 4.5 OS	see UGM [12]	On-chip software[2], Identifiable by Platform ID, Platform Build ID, Core ID and Patch ID JCOP 4.5 OS
Document	User Guidance and Administration Manual [12]	[12]	Electronic document[3]

[1] The TOE is delivered as wafer or module. The TOE can be collected at NXP site or is being shipped to the customer. See UGM [12] for details.
[2] Integrated in the certified Micro Controller with associated firmware and Crypto Library
[3] Via the NXP Docstore [15].

1.3.5 Evaluated Package Types

A number of package types are supported for this TOE. All package types, which are covered by the certification of the used hardware (see [13]), are also allowed to be used in combination with each product of this TOE.

The package types do not influence the security functionality of the TOE. They only define which pads are connected in the package and for what purpose and in which environment the chip can be used. Note that the security of the TOE is not dependent on which pad is connected or not - the connections just define how the product can be used. If the TOE is delivered as wafer the customer can choose the connection on his own.

2 Conformance Claims (ASE_CCL)

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

2.1 CC Conformance Claim

This Security Target claims to be conformant to version 3.1 of Common Criteria for Information Technology Security Evaluation according to

- "Common Criteria for Information Technology Security Evaluation, Part 1, Version 3.1, Revision 5, April 2017" [1]
- "Common Criteria for Information Technology Security Evaluation, Part 2, Version 3.1, Revision 5, April 2017" [2]
- "Common Criteria for Information Technology Security Evaluation, Part 3, Version 3.1, Revision 5, April 2017" [3]

The following methodology will be used for the evaluation:

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017" [4]

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in Section 6 "Extended Components Definition (ASE_ECD)".

2.2 Package Claim

This Security Target claims conformance to the assurance package EAL6. The augmentation to EAL6 is ASE_TSS.2 “TOE summary specification with architectural design summary” and ALC_FLR.1 “Basic flaw remediation”.

2.3 PP Claim

The Security Target claims demonstrable conformance to the Java Card System - Open Configuration Protection Profile, December 2017, Version 3.0.5 [7], certified by Bundesamt für Sicherheit in der Informationstechnik (BSI, BSI-CC-PP-0099-2017).

The Java Card Protection Profile[7] makes the use of Java Card RMI optional. The TOE does not support Java Card RMI.

The Java Card Protection Profile[7] (Appendix 2) makes managing Biometric Templates an optional feature. The TOE supports Biometric Template management functionality, offered as a customer selectable production item.

This ST is more restrictive than the PP [7] which Section 2.4 "Conformance Claim Rationale" provides a rational for.

2.4 Conformance Claim Rationale

2.4.1 TOE Type

The TOE type as stated in Section Section 1.2 "TOE Overview" of this ST corresponds to the TOE type of the PP as stated in Section 2.1 of [7] namely a Java Card platform, implementing the Java Card Specification Version 3.0.5 [17], [18], [16].

2.4.2 SPD Statement

2.4.2.1 Threats

The Security Problem Definition (SPD) statement that is presented in Section 4 "Security Problem Definition (ASE_SPD)" includes the threats as presented in the PP [7], but also includes refined and additional threats.

The following threats are refined as a result of TOE support for the optional feature of Biometric Template Management, defined in Appendix 2 of the PP [7].

- T.CONFID-APPLI-DATA[REFINED]
- T.INTEG-APPLI-DATA[REFINED]

The following additional threats are defined:

- T.OS_OPERATE
- T.COM_EXPLOIT
- T.LIFE_CYCLE
- T.UNAUTHORIZED_CARD_MNGT
- T.CONFID-UPDATE-IMAGE.LOAD
- T.INTEG-UPDATE-IMAGE.LOAD
- T.UNAUTH-UPDATE-IMAGE.LOAD
- T.INTERRUPT-OSU
- T.CONFIG
- T.SEC_BOX_BORDER
- T.MODULE_EXEC
- T.MODULE_REPLACEMENT

The threat T.OS_OPERATE is an additional threat added to cover incorrect operating system behavior, it is an addition to the threats in the PP [7].

The threat T.COM_EXPLOIT is included to cover communication channels attacks and it is an addition to the threats in the PP [7].

The threat T.LIFE_CYCLE is included to cover content management attacks and it is an addition to the threats in the PP [7].

The threat T.UNAUTHORIZED_CARD_MNGT is taken from the USIM PP [6] and the coherency of the security objectives based on that.

The threats T.CONFID-UPDATE-IMAGE.LOAD, T.INTEG-UPDATE-IMAGE.LOAD, T.UNAUTH-LOAD-UPDATE-IMAGE and T.INTERRUPT-OSU are included for the OS Update which is additional functionality the PP [7] 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 [7].

The threat T.SEC_BOX_BORDER is included for the Secure Box which is additional functionality the PP [7] allows.

The threats T.MODULE_EXEC and T.MODULE_REPLACEMENT are included for the Modular Design which is additional functionality the PP [7] allows. Furthermore some threats from the PP [7] are refined to cover additional assets from the Modular Design. This comprises threats T.CONFID-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-APPLI-CODE, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, and T.SID.1.

Note: The threat T.EXE-CODE-REMOTE is excluded, since the TOE does not support Java Card RMI. The Java Card Protection Profile [7] makes the use of Java Card RMI optional.

2.4.2.2 Organizational Security Policies

The SPD statement presented in Section 4 "Security Problem Definition (ASE_SPD)", copies the OSP from the PP [7], and adds following additional OSPs:

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

The Organizational Security Policy (OSP) OSP.PROCESS-TOE is introduced for the pre-personalisation feature of the TOE and is an addition to the OSPs in PP [7]. This OSP is copied from the Security IC PP [5].

The OSP OSP.KEY-CHANGE is introduced for the Security Domain (SD) feature of the TOE and is an addition to the OSPs in PP [7].

The OSP OSP.SECURITY-DOMAINS is introduced for the SD feature of the TOE and is an addition to the OSPs in PP [7].

The OSP.SECURE-BOX is introduced to allow execution of third party native code and is an addition to the OSPs in PP [7].

2.4.2.3 Assumptions

The SPD statement includes two of the three assumptions from the PP [7]. 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 [7]. 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 [7], following additional assumptions are added:

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

The assumption A.PROCESS-SEC-IC is taken from the underlying certified Micro Controller [13], which is compliant to the Security IC PP [5].

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 Application Provider Security Domain (APSD) and the correctness is verified on the TOE by Verification Authority Security Domain (VASD) before the package or applet is installed or loaded. A.APPS-PROVIDER and A.VERIFICATION-AUTHORITY are additions to PP [7] for card content management environment.

2.4.3 Security Objectives Statement

The statement of security objectives in the ST presented in Section 5 "Security Objectives" includes all security objectives as presented in the PP [7], but also includes a number of additional security objectives.

The Security Objective OT.BIO-MNGT, is added for the optional feature of Biometric Template Management defined in the PP [7] Appendix 2.

The other additional security objectives are:

- OT.IDENTIFICATION
- OT.RND
- OT.CONFID-UPDATE-IMAGE.LOAD
- OT.AUTH-LOAD-UPDATE-IMAGE
- OT.SECURE_LOAD_ACODE
- OT.SECURE_ACTIVATION_ADDITIONAL_CODE
- OT.TOE_IDENTIFICATION
- OT.DOMAIN-RIGHTS
- OT.APPLI-AUTH
- OT.COMM_AUTH
- OT.COMM_INTEGRITY
- OT.COMM_CONFIDENTIALITY
- OT.CARD-CONFIGURATION
- OT.SEC_BOX_FW
- OT.SID_MODULE

The security objective OT.IDENTIFICATION are part of the security objectives of the certified Micro Controller [13] (see also Section 1.3.1.1 "Micro Controller") and Crypto Lib [13] (see also Section 1.3.1.2.2 "Crypto Library"), which are also components of this composite certification. Therefore the security objective statement is equivalent to the PP [7] for this security objective. OT.IDENTIFICATION is also included for the pre-personalisation feature of the TOE, which is additional functionality the PP allows.

The security objective OT.CONFID-UPDATE-IMAGE.LOAD, OT.AUTH-LOADUPDATE-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.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 security objectives OT.CARD-CONFIGURATION is related to the configuration of the TOE via the Configuration Module, which is additional functionality the PP [7] allows.

The security objective OT.SEC_BOX_FW is related to the Secure Box, which is additional functionality the PP allows.

The security objective OT.SID_MODULE is related to the Modular Design of the TOE, which is additional functionality the PP [7] allows.

The security objective OT.RND is related to the Random Number Generator, and is taken from the Java Card Opern Platform PP [7], where it is denoted O.RNG.

The ST contains OE.APPLET, OE.VERIFICATION and OE.CODE-EVIDENCE from Security Objectives for the Operational Environment from [7]. Additionally, some of the Security Objectives for the Operational Environment from [7] 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 [7] to which conformance is claimed.

The security objectives O.INSTALL, O.LOAD, and O.DELETION from the PP [7] are not included since these functionality and objectives are covered by the refined OT.CARD-MANAGEMENT.

Note that the objective O.REMOTE is not included, since the TOE does not support Java Card RMI. The Java Card Protection Profile makes the use of Java Card RMI optional.

A part of the security objectives for the environment defined in the PP [7] has been included in this ST. The other part of security objectives for the environment, which is present in the PP [7], is used as part of the security objectives for the TOE in this ST. The ST also introduces following additional security objectives for the environment:

- OE.PROCESS_SEC_IC
- OE.USE_DIAG
- OE.USE_KEYS
- OE.APPS-PROVIDER
- OE.VERIFICATION-AUTHORITY
- OE.KEY-CHANGE
- OE.SECURITY-DOMAINS

The security objective for the environment OE.PROCESS_SEC_IC is from the hardware platform (Micro Controller [13] see also Section 1.3.1.1 "Micro Controller") 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 [5].

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 [7].

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.

The statement of security objectives for the environment is therefore considered to be equivalent to the security objectives in the PP [7] to which conformance is claimed.

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 the PP allows.

2.4.4 Security Functional Requirements Statement

The statement of security functional requirements copies most SFRs as defined in the PP [7], 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 [7]. Moreover as requested by the PP [7] the ST adds additional threats, objectives and SFRs to fully cover and describe additional security functionality implemented in the TOE.

In the PP [7] the use of the Java Card RMI is optional. The TOE does not implement Java Card RMI, therefore this ST restricts remote access from the CAD to the services implemented by the applets on the card to none. As a result the SFRs concerning Java Card RMI

- FDP_ACF.1/JCRMI
- 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
- FMT_SMR.1/JCRMI

are not included in the ST.

The SFR FDP_ITC.2/INSTALLER from the PP [7] is replaced by FDP_ITC.2[CCM] which enforces the Security Domain access control policy and the Secure Channel Protocol information flow policy and which are more restrictive than the PACKAGE LOADING information flow control SFP from PP [7].

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

provide a concrete and more restrictive implementation of the PACKAGE LOADING information flow control SFP from PP [7].

The table below lists the SFRs from CarG of PP [7] and their corresponding refinements in this ST.

Table 8. CarG SFRs refinements

SFR from PP [7]	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]

Table 8. CarG SFRs refinements...*continued*

SFR from PP [7]	Refinement
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 [7] and add functionality to the TOE making the statement of security requirements more restrictive than the PP [7]:

FDP_ROL.1[CCM], FPT_FLS.1[CCM] and FPT_PHP.3 realize additional security functionality for the card manager which is allowed by the PP [7].

The set of SFRs that define the security domains mechanism as specified by GlobalPlatform realize refinements of SFRs from PP [7] (see above Table 8 "CarG SFRs refinements") and additional security functionality which is allowed by the PP [7]. 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 [7] (see above Table 8 "CarG SFRs refinements"), add additional security functionality and include a JCOPX API which is allowed by the PP [7]. 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 that define the Configuration Module realize additional security functionality, which is allowed by the PP [7]. 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 Secure Box, realize additional security functionality which is allowed by the Protection Profile (PP) [7]. This set of SFRs comprise

- FDP_ACC.2[SecureBox]
- FDP_ACF.1[SecureBox]
- FMT_MSA.1[SecureBox]
- FMT_MSA.3[SecureBox]
- FMT_SMF.1[SecureBox].

The set of SFRs that define the Modular Design realize additional security functionality, which is allowed by the PP [7]. This set of SFRs are grouped as ModDesG and MDEL, for module deletion, comprising

- ModDesG
  - FDP_IFC.1[MODULAR-DESIGN]
  - FDP_IFF.1[MODULAR-DESIGN]
  - FIA_ATD.1[MODULAR-DESIGN]
  - FIA_USB.1[MODULAR-DESIGN]
  - FMT_MSA.1[MODULAR-DESIGN]
  - FMT_MSA.3[MODULAR_DESIGN
  - FMT_SMF.1[MODULAR-DESIGN]
  - FMT_SMR.1[MODULAR-DESIGN]
  - FPT_FLS.1[MODULAR-DESIGN]
- MDEL
  - FDP_ACC.2[MDEL]
  - FDP_ACF.2[MDEL]
  - FDP_RIP.1[MDEL]
  - FMT_MSA.1[MDEL]
  - FMT_MSA.3[MDEL]
  - FMT_SMF.1[MDEL]
  - FMT_SMR.1[MDEL]
  - FPT_FLS.1[MDEL]

The set of SFRs that define the OS Update mechanism realize additional security functionality, which is allowed by the PP [7]. This set of SFRs comprise:

- FDP_IFC.2[OSU]
- FDP_IFF.1[OSU]
- FIA_UAU.1[OSU]
- FIA_UAU.4[OSU]
- FIA_UID.1[OSU]
- FMT_MSA.1[OSU]
- FMT_MSA.3[OSU]
- FMT_SMF.1[OSU]
- FMT_SMR.1[OSU]
- FPT_FLS.1[OSU]

The SFRs FAU_SAS.1[SCP], FIA_AFL.1[PIN], FPT_EMSEC.1 and FPT_PHP.3 realize additional security functionality which is allowed by the PP [7].

The SFRs FCS_CKM.2 and FCS_CKM.3 realize security functionality required by the Java Card API [16] which is allowed by the PP [7].

3 Security Aspects

This chapter describes only the Security Aspects which are addional to or refinements of those in the Java Card - Open Configuration Protection Profile [7].

3.1 Confidentiality

Table 9.

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.
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

Table 10.

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 package in transit to the card. For instance, a package contains the values to be used for initializing the static fields of the package.

Table 10. *...continued*

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

Table 11.

SA.EXE-APPLI-CODE	**Execution of Application Code** Application (byte)code must be protected against unauthorized execution. This concerns: • invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([19]) • jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code. • 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: • invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([19]) • 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 instances1, and isolation of their data and code between packages (that is, controlled execution contexts) as well as between packages 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.

3.4 Bytecode Verification

Table 12.

SA.VERIFICATION	**Bytecode Verification** Bytecode must be verified prior to being executed. Bytecode verification includes: • how well-formed CAP file is and the verification of the typing constraints on the bytecode, • 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

Table 13.

SA.CARD-MANAGEMENT	**Card Management** • The card manager (CM) shall control the access to card management functions such as the installation, update or deletion of applets. • The card manager shall implement the card issuer's policy on the card.
SA.INSTALL	**Installation** • The TOE must be able to return to a safe and consistent state when the installation of a package or an applet fails or be cancelled (whatever the reasons). • 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. • The procedure of loading and installing a package shall ensure its integrity and authenticity.
SA.SID	**Subject Identification** • Users and subjects of the TOE must be identified. • 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 package 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** • 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. • Erasure, if deemed successful, shall ensure that the deleted class instance is no longer accessible.

Table 13. *...continued*

SA.DELETION	**Deletion** • Deletion of installed applets (or packages) 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. • 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. Package deletion shall make the code of the package no longer available for execution. • 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 package are functionally different operations and may obey different security rules. For instance, specific packages can be declared to be undeletable (for instance, the Java Card API packages), or the dependency between installed packages may forbid the deletion (like a package using super classes or super interfaces declared in another package).

3.6 Services

Table 14.

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** • The TOE must ensure continued correct operation of its security functions. • 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 packages.

Table 14. *...continued*

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: • Keys shall be generated in accordance with specified cryptographic key generation algorithms and specified cryptographic key sizes, • Keys must be distributed in accordance with specified cryptographic key distribution methods, • Keys must be initialized before being used, • 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: • Atomic update of PIN value and try counter, • No rollback on the PIN-checking function, • Keeping the PIN value (once initialized) secret (for instance, no clear-PIN-reading function), • Enhanced protection of PIN's security attributes (state, try counter ...) in confidentiality and integrity.

Table 14. *...continued*

SA.SCP	**Smart Card Platform** The smart card platform must be secure with respect to the SFRs. Then: • 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. • 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 the packages of the Java Card API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System. • It provides secure low-level cryptographic processing to the Java Card System. • 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. • 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). • It safely transmits low-level exceptions to the TOE (arithmetic exceptions, checksum errors), when applicable. • 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 [5]), and will be tamper resistant to actually prevent an attacker from extracting or altering security data (like cryptographic keys) by using commonly employed techniques (physical probing and sophisticated analysis of the chip). This especially matters to the management (storage and operation) of cryptographic keys.
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

Table 15.

SA.CONFIG-APPLET	**Config Applet** The Config Applet is a JCOP functionality which allows to: • Read and modify configuration items in the configuration area of the TOE, • Disable Access to configuration item.

3.8 OS Update

Table 16.

SA.OSU	**OS Update** The UpdaterOS facilitates updating the JCOP 4.5 OS and the UpdaterOS itself. It ensures that only valid updates can be installed on the TOE.

3.9 Modular Design

3.9.1 Modular Design

Table 17. SA.MODULAR-DESIGN

SA.MODULAR-DESIGN	The TOE might contain one or more Modules implementing particular functionality. The list of Modules present in the TOE must be retrievable and each module identifiable. Interfaces to a Module can be Public or TOE internal. Public Interfaces can directly be accessed by any Applet or via an APDU, TOE internal interfaces can only be accessed by the TOE itself, Applets use the corresponding JavaCard API [16].

3.9.2 Module Invocation

Table 18. SA.MODULE-INVOCATION

SA.MODULAR-INVOCATION	Invoking a module must be transparent to the user. If a Module has a TOE internal interface, is not present and is invoked by the user, the TOE must preserve a secure state by throwing an exception or returning an appropriate error status word to the CAD.

4 Security Problem Definition (ASE_SPD)

4.1 Assets

Assets are security-relevant elements to be directly protected by the TOE. Confidentiality of assets is always intended with respect to un-trusted people or software, as various parties are involved during the first stages of the smart card product life-cycle. Details concerning the threats are given in Section 4.2 "Threats" 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 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 [7].

4.1.1 User Data

User Data assets defined in Section 5.1.1 of [7] are given here as a reminder, with the inclusion of the Biometric assets defined in Appendix 2.

D.APP_CODE	The code of the applets and libraries loaded on the card. To be protected from unauthorized modification.
D.APP_C_DATA	Confidentiality - sensitive data of the applications, like the data contained in an object, a static field of a package, 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 and the PIN security attributes (PIN Try limit, PIN Try counter and State). To be protected from unauthorized modification.
D.APP_KEYS	Cryptographic keys owned by the applets. To be protected from unauthorized disclosure and modification.
D.PIN	Any end-user's PIN. To be protected from unauthorized disclosure and modification.
D.BIO	Any biometric template To be protected from unauthorized disclosure and modification.

Assets refined by this security target are given below:

D.APSD_KEYS	Refinement of D.APP_KEYS of [7]. Application Provider Security Domains cryptographic keys are 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 [7]. Issuer Security Domain cryptographic keys are 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 [7]. 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. To be protected from unauthorized modification.

4.1.2 TSF Data

TSF data defined in Section 5.1.2 of [7] are given here as a reminder.

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.

Additionally defined TSF data categories for this Security Target are given below

D.CONFIG_ITEM	A configuration that can be changed using the Configuration Mechanism.
D.MODULE_CODE	The code of a Module. The code of a module might comprise Java code, native code, code of a native Library or a combination of them. To be protected against unauthorized disclosure and modification. Further to be protected against unauthorized removal or presence forgery.
D.MODULE_DATA	Private data of a Module, like the contents of its private fields. To be protected from unauthorized disclosure and modification.
D.UPDATE_IMAGE	Can be an update for JCOP 4.5 and UpdaterOS. It is sent to the TOE, received by the UpdaterOS. 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 [10].
D.TOE_IDENTIFIER	Identification data specific to the TOE
D.ATTACK_COUNTER	The Attack Counter is incremented when a potential attack is detected. When the Attack Counter exceeds its limit, card performance is increasingly restricted. The attack counter is automatically decremented through normal use of the card.

4.1.3 Biometric Templates

Biometric Templates are an optional feature detailed in Appendix 2.2 of [7]

D.BIO	Any biometric template. To be protected from unauthorized disclosure and modification.

4.2 Threats

4.2.1 Confidentiality

4.2.1.1 T.CONFID-APPLI-DATA[REFINED]: 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.BIO,* D.APP_C_DATA, D.PIN and D.APP_KEYS.

4.2.1.2 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 and D.MODULE_CODE.

4.2.1.3 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, D.CRYPTO and D.MODULE_CODE.

4.2.2 Integrity

4.2.2.1 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 and D.MODULE_CODE.

4.2.2.2 T.INTEG-APPLI-CODE.LOAD: Integrity of Application Code - Load

The attacker modifies (part of) its own or another application code when an application package is transmitted to the card for installation. See SA.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

4.2.2.3 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.BIO,* D.APP_I_ DATA, D.PIN, D.APP_KEYS, D.ISD_KEYS, D.VASD_KEYS and S.APSD_KEYS.

This threat is a refinement of the Threat T.INTEG-APPLI-DATA from [7].

4.2.2.4 T.INTEG-APPLI-DATA.LOAD: Integrity of Application Data - Load

The attacker modifies (part of) the initialization data contained in an application package when the package 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.

4.2.2.5 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 and D.MODULE_CODE.

4.2.2.6 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, D.CRYPTO and D.MODULE_DATA.

4.2.3 Identity Usurpation

4.2.3.1 T.SID.1: Subject Identification 1

An applet or Module impersonates another application or Module, 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 and SA.MODULAR- DESIGN 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.

4.2.3.2 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).

4.2.4 Unauthorized Execution

4.2.4.1 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.

4.2.4.2 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.

4.2.4.3 T.NATIVE: Native Code Execution

An applet executes a native method to bypass a TOE Security Function such as the firewall. See SA.EXE-JCS-CODE and SA.EXE-APPLI-CODE for details.

Directly threatened asset(s): D.APP_CODE.

4.2.4.4 T.MODULE_EXEC: Code Execution of Modules

The attacker bypasses the presence check of a Module which is not present with TOE internal interface to execute arbitrary code. See SA.MODULAR-DESIGN and SA.MODULE-INVOCATION for details. Directly threatened asset(s): D.MODULE_ CODE.

4.2.5 Denial of Service

4.2.5.1 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.

4.2.6 Card Management

4.2.6.1 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 package file
- installation of a package file
- extradition of a package file or an applet
- personalization of an applet or a Security Domain
- deletion of a package 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 threat is based on the definition taken from USIM PP[6].

4.2.6.2 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.

4.2.6.3 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.

4.2.7 Services

4.2.7.1 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.

4.2.8 Miscellaneous

4.2.8.1 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 Differential Power Analysis (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.

Application note:

If sensitive result is supported by the TOE, this threat covers the following subthreat exploiting specifically the listed assets below:

- The attacker performs a physical manipulation to alter (part of) an application's integrity-sensitive data. See SA.INTEG-APPLI-DATA-PHYS for details. Directly threatened asset(s): D.APP_I_DATA, D.PIN, and D.APP_KEYS.

4.2.9 Operating System

4.2.9.1 T.OS_OPERATE: Incorrect Operating System Behavior

Modification of the correct OS behavior by unauthorized use of TOE or use of incorrect or unauthorized instructions or commands or sequence of commands, in order to obtain an unauthorized execution of the TOE code. An attacker may cause a malfunction of TSF or of the Smart Card embedded OS in order to (1) by-pass the security mechanisms (i.e. authentication or access control mechanisms) or (2) obtain unexpected result from the embedded OS behavior. Different kind of attack path may be used as:

1. Applying incorrect unexpected or unauthorized instructions, commands or command sequences,
2. Provoking insecure state by insertion of interrupt (reset), premature termination of transaction or communication between IC and the reading device.

Info: Any implementation flaw in the OS itself can be exploited with this attack path to lead to an unsecured state of the state machine of the OS. The attacker uses the available interfaces of the TOE. A user could have certain specified privileges that allow loading of selected programs. Unauthorized programs, if allowed to be loaded, may include either the execution of legitimate programs not intended for use during normal operation (such as patches, filters, Trojan horses, etc.) or the unauthorized loading of programs specifically targeted at penetration or modification of the security functions. Attempts to generate a non-secure state in the Smart Card may also be made through

premature termination of transactions or communications between the IC and the card reading device, by insertion of interrupts, or by selecting related applications that may leave files open.

4.2.10 Configuration Module

4.2.10.1 T.CONFIG: Unauthorized configuration

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

4.2.11 Secure Box

4.2.11.1 T.SEC_BOX_BORDER: SecureBox Border Infringement

An attacker may try to use malicious code placed in the Secure Box to modify the correct behavior of the Operating System (OS). With the aim to

1. disclose the Java Card System code,
2. disclose or alter applet code, disclose or alter Java Card System data, or disclose or alter applet data.

4.2.12 Module replacement

4.2.12.1 T.MODULE_REPLACEMENT: Replacement of Module

An attacker loads a Module with functionality differing from a previously deleted Module to bypass TOE Security Functions. See SA.MODULAR-DESIGN for details. Directly threatened assets: D.JCS_DATA.

4.2.13 OS Update

4.2.13.1 T.CONFID-UPDATE-IMAGE.LOAD: Confidentiality of update Image - Load

The attacker executes an application to disclose data belonging to another application. See SA.CONFID-UPDATE-IMAGE for details. Directly threatened asset(s): D.UPDATE_IMAGE, D.JCS_CODE and D.JCS_DATA.

4.2.13.2 T.INTEG-UPDATE-IMAGE.LOAD: Integrity of update Image -Load

TThe 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.

4.2.13.3 T.UNAUTH-UPDATE-IMAGE.LOAD: Load an unauthorized update

The attacker tries to upload an unauthorized Update Image. See SA.INTEG-UPDATE-IMAGE for details. Directly threatened asset(s): D.UPDATE_IMAGE, D.JCS_CODE and D.JCS_DATA.

4.2.13.4 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.TOE_IDENTIFIER, D.UPDATE_IMAGE, D.JCS_CODE and D.JCS_DATA.

4.3 Organisational Security Policies

4.3.1 OSP.VERIFICATION: File Verification

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

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

4.3.2 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.

4.3.3 OSP.KEY-CHANGE: Security Domain Keys Change

The Application Provider (AP) shall change its initial security domain keys (APSD) before any operation on its Security Domain.

4.3.4 OSP.SECURITY-DOMAINS: Security Domains

Security domains can be dynamically created, deleted and blocked during usage phase in post-issuance mode.

4.3.5 OSP.SECURE-BOX: Secure Box Border

Execution of untrusted native code shall be possible without any harm, manipulation, or influence on other parts of the TOE.

4.4 Assumptions

Note that the assumption A.DELETION is excluded. The Card Manager is part of the TOE and therefore the assumption is no longer relevant.

4.4.1 A.APPLET: Applets without Native Methods

Applets loaded post-issuance do not contain native methods. The Java Card specification explicitly ”does not include support for native methods” ([17]) outside the API.

4.4.2 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.

4.4.3 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.

4.4.4 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.

4.4.5 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.

4.4.6 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 (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.

4.4.7 A.VERIFICATION-AUTHORITY: Verification Authority

The VA is a trusted actor who is able to guarantee and check the digital signature attached to a basic or secure application.

Info: As a consequence, it guarantees the success of the application validation upon loading.

5 Security Objectives

5.1 Security Objectives for the TOE

5.1.1 Identification

5.1.1.1 OT.SID: Subject Identification

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

5.1.1.2 OT.SID_MODULE: Subject Identification of Modules

The TOE shall uniquely identify every Module before granting it access to any service.

5.1.2 Execution

5.1.2.1 OT.FIREWALL: Firewall

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

5.1.2.2 OT.GLOBAL_ARRAYS_CONFID: Confidentiality of Global Arrays

The TOE shall ensure that the APDU buffer that is shared by all applications is always cleared upon applet selection. The TOE shall ensure that the global byte array used for the invocation of the install method of the selected applet is always cleared after the return from the install method.

5.1.2.3 OT.GLOBAL_ARRAYS_INTEG: Integrity of Global Arrays

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

5.1.2.4 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.

5.1.2.5 OT.OPERATE: Correct Operation

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

5.1.2.6 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.

5.1.2.7 OT.RESOURCES: Resources availability

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

5.1.2.8 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.

5.1.3 Services

5.1.3.1 OT.ALARM: Alarm

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

5.1.3.2 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.

5.1.3.3 OT.RND: Random Numbers Generation

The TOE shall ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have sufficient entropy. The TOE shall ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

5.1.3.4 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.

5.1.3.5 OT.PIN-MNGT: Pin Management

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

Application Note:

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

5.1.3.6 OT.BIO-MNGT: Biometric Template Management

The TOE shall provide a means to securely manage biometric templates. This concerns the optional packages javacardx.biometry or javacardx.biometry1toN of the Java Card platform.

5.1.3.7 OT.TRANSACTION: Transaction

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

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

5.1.4 Object Deletion

5.1.4.1 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.

5.1.5 Applet Management

5.1.5.1 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 [7].

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.

5.1.5.2 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).

5.1.5.3 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.

5.1.5.4 OT.COMM_INTEGRITY: Communication Request Integrity

The TOE shall verify the integrity of the card management requests that the card receives.

5.1.5.5 OT.COMM_CONFIDENTIALITY: Communication Request Confidentiality

The TOE shall be able to process card management requests containing encrypted data.

5.1.6 Card Management

5.1.6.1 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 [7] 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 [7]. 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 [7] 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 [7]. 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.

5.1.7 Smart Card Platform

5.1.7.1 OT.SCP.IC IC: Physical Protection

The SCP shall provide all IC 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 [7] 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.

5.1.7.2 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 [7] 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.

5.1.7.3 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 [7] 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.

5.1.7.4 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.

5.1.8 Secure Box

5.1.8.1 OT.SEC_BOX_FW: SecureBox firewall

The TOE shall provide separation between the Secure Box native code and the Java Card System. The separation shall comprise software execution and data access.

5.1.9 Configuration Module

5.1.9.1 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.

5.1.10 OS Update

5.1.10.1 OT.CONFID-UPDATE-IMAGE.LOAD: Confidentiality of Update Image

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.

5.1.10.2 OT.AUTH-LOAD-UPDATE-IMAGE: Authorization of Update Image - Load

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

5.1.10.3 OT.SECURE_LOAD_ACODE: Secure loading of 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.

5.1.10.4 OT.SECURE_ACTIVATION_ADDITIONAL_CODE: Secure activation of the Additional Code

This objective is taken over from [11] (O.Secure_AC_Activation) with editorial modification.

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.

5.1.10.5 OT.TOE_IDENTIFICATION: Secure identification of the TOE

This objective is taken over from [11] and further refined.

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 or the Final TOE. The user shall be able to uniquely identify Initial TOE and Additional Code(s) or the final TOE which are embedded in the Final TOE.

5.2 Security Objectives for the Operational Environment

5.2.1 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.

5.2.2 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 Security Target as well as for the Protection Profile ([7]).

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.

5.2.3 OE.APPS-PROVIDER: Application Provider

The AP shall be a trusted actor that provides applications. The AP is responsible for its security domain keys.

5.2.4 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.

5.2.5 OE.KEY-CHANGE: Security Domain Key Change

The AP must change its security domain initial keys before any operation on it.

5.2.6 OE.SECURITY-DOMAINS: Security Domains

Security domains can be dynamically created, deleted and blocked during usage phase in post-issuance mode.

5.2.7 OE.USE_DIAG: Secure TOE communication protocols

Secure TOE communication protocols shall be supported and used by the environment.

5.2.8 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.

5.2.9 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.

5.2.10 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.

5.3 Security Objectives Rationale

In this section it is proven that the security objectives described in Section 4 "Security Problem Definition (ASE_SPD)" can be traced for all aspects identified in the TOE-security environment and that they are suited to cover them. At least one security objective results from each assumption, OSP, and each threat. At least one threat, one OSP or assumption exists for each security objective.

Security Problem Definition	Security Objective
T.CONFID-APPLI-DATA	OT.SID OT.FIREWALL OT.GLOBAL_ARRAYS_CONFID OT.OPERATE OT.REALLOCATION OT.ALARM OT.CIPHER OT.KEY-MNGT OT.PIN-MNGT OT.TRANSACTION OE.VERIFICATION OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT OT.RND OT.SECURE_LOAD_ACODE OT.BIO-MNGT

Security Problem Definition	Security Objective
T.CONFID-JCS-CODE	OT.NATIVE OE.VERIFICATION OT.CARD-MANAGEMENT OT.SECURE_LOAD_ACODE
T.CONFID-JCS-DATA	OT.SID OT.FIREWALL OT.OPERATE OT.ALARM OE.VERIFICATION OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT OT.SID_MODULE OT.SECURE_LOAD_ACODE
T.INTEG-APPLI-CODE	OT.NATIVE OE.VERIFICATION OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.SECURE_LOAD_ACODE
T.INTEG-APPLI-CODE.LOAD	OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.APPLI-AUTH OT.SECURE_LOAD_ACODE
T.INTEG-APPLI-DATA[REFINED]	OT.SID OT.FIREWALL OT.GLOBAL_ARRAYS_INTEG OT.OPERATE OT.REALLOCATION OT.ALARM OT.CIPHER OT.KEY-MNGT OT.PIN-MNGT OT.TRANSACTION OE.VERIFICATION OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT OE.CODE-EVIDENCE OT.DOMAIN-RIGHTS OT.RND OT.SECURE_LOAD_ACODE OT.BIO-MNGT
T.INTEG-APPLI-DATA.LOAD	OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.APPLI-AUTH OT.SECURE_LOAD_ACODE

Security Problem Definition	Security Objective
T.INTEG-JCS-CODE	OT.NATIVE OE.VERIFICATION OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.SECURE_LOAD_ACODE
T.INTEG-JCS-DATA	OT.SID OT.FIREWALL OT.OPERATE OT.ALARM OE.VERIFICATION OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT OE.CODE-EVIDENCE OT.SID_MODULE OT.SECURE_LOAD_ACODE
T.SID.1	OT.SID OT.FIREWALL OT.GLOBAL_ARRAYS_CONFID OT.GLOBAL_ARRAYS_INTEG OT.CARD-MANAGEMENT OT.SID_MODULE
T.SID.2	OT.SID OT.FIREWALL OT.OPERATE OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT
T.EXE-CODE.1	OT.FIREWALL OE.VERIFICATION
T.EXE-CODE.2	OE.VERIFICATION
T.NATIVE	OT.NATIVE OE.APPLET OE.VERIFICATION
T.MODULE_EXEC	OT.OPERATE OT.ALARM OE.APPLET OT.SCP.SUPPORT OT.SID_MODULE
T.RESOURCES	OT.OPERATE OT.RESOURCES OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT

Security Problem Definition	Security Objective
T.UNAUTHORIZED_CARD_MNGT	OT.CARD-MANAGEMENT OT.DOMAIN-RIGHTS OT.COMM_AUTH OT.COMM_INTEGRITY OT.APPLI-AUTH
T.LIFE_CYCLE	OT.CARD-MANAGEMENT OT.DOMAIN-RIGHTS
T.COM_EXPLOIT	OT.COMM_AUTH OT.COMM_INTEGRITY OT.COMM_CONFIDENTIALITY
T.OBJ-DELETION	OT.OBJ-DELETION
T.CONFIG	OT.CARD-CONFIGURATION
T.PHYSICAL	OT.SCP.IC OT.SENSITIVE_RESULTS_INTEG
T.OS_OPERATE	OT.OPERATE OT.SECURE_LOAD_ACODE OT.SECURE_ACTIVATION_ADDITIONAL_CODE
T.SEC_BOX_BORDER	OT.SEC_BOX_FW
T.RND	OT.RND
T.MODULE_REPLACEMENT	OT.OPERATE OE.APPLET OT.SCP.SUPPORT OT.SID_MODULE
T.CONFID-UPDATE-IMAGE.LOAD	OT.CONFID-UPDATE-IMAGE.LOAD OE.CONFID-UPDATE-IMAGE.CREATE
T.INTEG-UPDATE-IMAGE.LOAD	OT.SECURE_LOAD_ACODE
T.UNAUTH-UPDATE-IMAGE.LOAD	OT.SECURE_LOAD_ACODE OT.AUTH-LOAD-UPDATE-IMAGE
T.INTERRUPT-OSU	OT.SECURE_LOAD_ACODE OT.TOE_IDENTIFICATION OT.SECURE_AC_ACTIVATION
OSP.VERIFICATION	OE.VERIFICATION OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.APPLI-AUTH
OSP.PROCESS-TOE	OT.IDENTIFICATION OT.TOE_IDENTIFICATION
OSP.KEY-CHANGE	OE.KEY-CHANGE
OSP.SECURITY-DOMAINS	OE.SECURITY-DOMAINS
OSP.SECURE-BOX	OT.SEC_BOX_FW
A.APPLET	OE.APPLET

Security Problem Definition	Security Objective
A.VERIFICATION	OE.VERIFICATION OE.CODE-EVIDENCE
A.USE_DIAG	OE.USE_DIAG
A.USE_KEYS	OE.USE_KEYS
A.PROCESS-SEC-IC	OE.PROCESS_SEC_IC
A.APPS-PROVIDER	OE.APPS-PROVIDER
A.VERIFICATION-AUTHORITY	OE.VERIFICATION-AUTHORITY

5.3.1 Threats

5.3.1.1 Confidentiality

5.3.1.1.1 T.CONFID-APPLI-DATA[REFINED]

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 [18].
OT.GLOBAL_ARRAYS_CONFID	Counters this threat by preventing the disclosure of the information stored in the APDU buffer.
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.
OT.TRANSACTION	Counters this threat by providing appropriate management of keys, PIN's which are particular cases of an application's sensitive data.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.

Objective	Rationale
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.
OT.RND	Counters this threat by providing appropriate management of keys, PIN's which are particular cases of an application's sensitive data.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.1.2 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.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.1.3 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.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.
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.

Objective	Rationale
OT.SID_MODULE	Counters this threat by providing correct identification of applets.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.2 Integrity

5.3.1.2.1 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	Contributes to counter this threat by ensuring that integrity and authenticity evidences exist for the application code loaded into the platform.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.2.2 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.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.2.3 T.INTEG-APPLI-DATA[REFINED]

Objective	Rationale
OT.SID	Counters this threat by providing correct identification of applets.

Objective	Rationale
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.OPERATE	Counters the threat by ensuring that the firewall, which is dynamically enforced, shall never stop operating.
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.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode.
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.
OT.RND	Counters this threat by providing appropriate management of keys, PINs which are particular cases of an application's sensitive data.

Objective	Rationale
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.2.4 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.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.2.5 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.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.2.6 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.

Objective	Rationale
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.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes.
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.SID_MODULE	Counters this threat by providing correct identification of applets.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.

5.3.1.3 Identity Usurpation

5.3.1.3.1 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).
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.
OT.SID_MODULE	Counters this threat by providing unique subject identification.

5.3.1.3.2 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.

5.3.1.4 Unauthorized Excecution

5.3.1.4.1 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.

5.3.1.4.2 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.

5.3.1.4.3 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.APPLET	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.

5.3.1.4.4 T.MODULE_EXEC

Objective	Rationale
OT.OPERATE	Counters the threat by ensuring correct working order.
OT.ALARM	Counters this threat by obtaining clear warning and error messages when the TOE internal interface of a ”not present” Module shall be invoked.
OE.APPLET	Contributes to counter this threat by ensuring that no native applets shall be loaded in post-issuance.
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.
OT.SID_MODULE	Counters this threat by providing correct identification of Modules.

5.3.1.5 Denial of Service

5.3.1.5.1 T.RESOURCES

Objective	Rationale
OT.OPERATE	Counters the threat by ensuring correct working order.
OT.RESOURCES	Counters 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.

5.3.1.6 Card Management

5.3.1.6.1 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.
OT.APPLI-AUTH	Counters this threat by ensuring that the loading of a package is safe.

5.3.1.6.2 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.

5.3.1.6.3 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.

5.3.1.7 Services

5.3.1.7.1 T.OBJ-DELETION

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

5.3.1.8 Miscellaneous

5.3.1.8.1 T.PHYSICAL

Objective	Rationale
OT.SCP.IC	Counters physical attacks. Physical protections rely on the underlying platform and are therefore an environmental issue.
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.

5.3.1.9 Operating System

5.3.1.9.1 T.OS_OPERATE

Objective	Rationale
OT.OPERATE	Contributes to counter the threat by ensuring the correct continuation of operation of the TOE's logical security functions. Security mechanisms have to be implemented to avoid fraudulent usage of the TOE, usage of certain memory regions, or usage of incorrect or unauthorized instructions or commands or sequence of commands. The security mechanisms must be designed to always put the TOE in a known and secure state.
OT.SECURE_LOAD_ACODE	Counters this threat by checking the authenticity and integrity of the loaded Additional Code.
OT.SECURE_ACTIVATION_ ADDITIONAL_CODE	Counters this threat by atomically activating the Additional Code after all operations needed for the Additional Code to operate are completed. This prevents the TOE from executing incorrect or unauthorized instructions.

5.3.1.10 Random Numbers

5.3.1.10.1 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.

5.3.1.11 Configuration Module

5.3.1.11.1 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.

5.3.1.12 Secure Box

5.3.1.12.1 T.SEC_BOX_BORDER

Objective	Rationale
OT.SEC_BOX_FW	Counters the threat by ensuring that the native code and data in Secure Box is separated from the rest of the TOE. Due to this separation the native code in the Secure Box cannot harm the code and data outside the Secure Box.

5.3.1.13 Module replacement

5.3.1.13.1 T.MODULE_REPLACEMENT

Objective	Rationale
OT.OPERATE	Counters the threat by ensuring correct working order.
OE.APPLET	Contributes to counter this threat by ensuring that no native applets shall be loaded in post-issuance.
OT.SCP.SUPPORT	Intended to support the OT.OPERATE objective of the TOE, thus indirectly related to the threats that these objectives contribute to counter.
OT.SID_MODULE	Counters this threat by providing correct identification of Modules.

5.3.1.14 OS Update

5.3.1.14.1 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.

5.3.1.14.2 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.

5.3.1.14.3 T.UNAUTH-UPDATE-IMAGE.LOAD

Objective	Rationale
OT.SECURE_LOAD_ACODE	Counters the threat directly by ensuring the authenticity and integrity of D.UPDATE_IMAGE.
OT.SECURE_LOAD_ACODE	Counters the threat directly by ensuring the authenticity and integrity of D.UPDATE_IMAGE.
OT.AUTH-UPDATE-IMAGE.LOAD	Counters the threat directly by ensuring that only authorized (allowed version) images can be loaded

5.3.1.14.4 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.

5.3.2 Organisational Security Policies

5.3.2.1 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 package 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.
OT.APPLI-AUTH	Contributing to enforce the OSP by ensuring that the loading of a package into the card is safe.

5.3.2.2 OSP.PROCESS-TOE

Objective	Rationale
OT.IDENTIFICATION	Enforces this organisational security policy by ensuring that the TOE can be uniquely identified.
OT.TOE_IDENTIFICATION	Enforces this organisational security policy by ensuring that the TOE can be uniquely identified after loading of Additional Code.

5.3.2.3 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.

5.3.2.4 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.

5.3.2.5 OSP.SECURE-BOX

Objective	Rationale
OT.SEC_BOX_FW	Addresses directly this organizational security policy by ensuring that the native code and data in Secure Box is separated from the rest of the TOE. Due to this separation the native code in the Secure Box cannot harm the code and data outside the Secure Box.

5.3.3 Assumptions

5.3.3.1 A.APPLET

Objective	Rationale
OE.APPLET	Upholds the assumption by ensuring that no applet loaded post issuance shall contain native methods.

5.3.3.2 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.

Objective	Rationale
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.

5.3.3.3 A.USE_DIAG

Objective	Rationale
OE.USE_DIAG	Directly upholds this assumption.

5.3.3.4 A.USE_KEYS

Objective	Rationale
OE.USE_KEYS	Directly upholds this assumption.

5.3.3.5 A.PROCESS-SEC-IC

Objective	Rationale
OE.PROCESS_SEC_IC	Directly upholds this assumption.

5.3.3.6 A.APPS-PROVIDER

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

5.3.3.7 A.VERIFICATION-AUTHORITY

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

6 Extended Components Definition (ASE_ECD)

The component FCS_RNG is taken over from the Java Card PP [7]. In addition following components are defined for the TOE.

6.1 Definition of Family ”Audit Data Storage (FAU_SAS)”

This section has been taken over from the certified (BSI-PP-0084-2014) Security IC Platform Protection profile [5].

To define the security functional requirements of the TOE an additional family (FAU_SAS) of the Class FAU (Security Audit) 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.

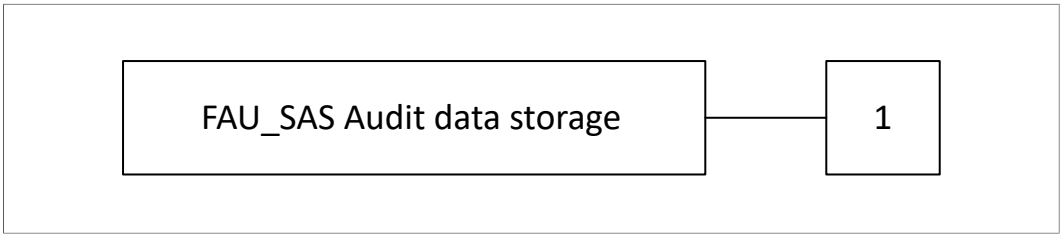
The family “Audit data storage (FAU_SAS)” is specified as follows.

FAU_SAS Audit data storage

Family behavior

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

Component leveling



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 Definition of Family ”TOE emanation (FPT_EMSEC)”

This section has been taken over from the certified (BSI-PP-0055) Protection Profile Machine Readable travel Document with "ICAO Application", Basic Access Control [9].

The sensitive family FPT_EMSEC (TOE Emanation) of the Class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. The TOE shall prevent attacks against the TOE and other secret data where the attack is based on external observable physical phenomena of the TOE. Examples of such attacks are evaluation of TOE’s electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc. This family describes the functional requirements for the limitation of intelligible emanations which are not directly addressed by any other component of CC part 2 [2].

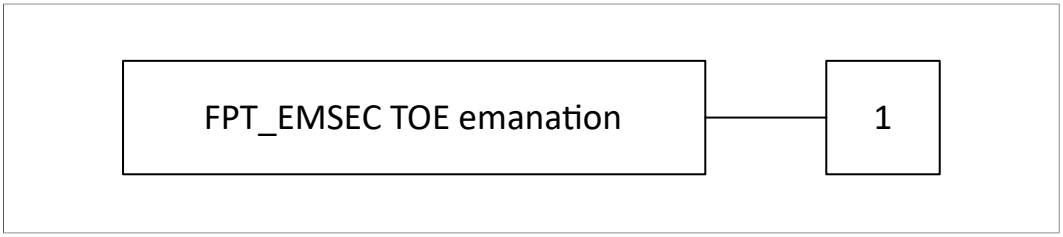
The family “TOE emanation (FPT_EMSEC)” is specified as follows.

FPT_EMSEC TOE emanation

Family behavior

This family defines requirements to mitigate intelligible emanations.

Component leveling



FPT_EMSEC:

TOE emanation has two constituents:

FPT_EMSEC.1.1:

Limit of emissions requires to not emit intelligible emissions enabling access to TSF data or user data.

FPT_EMSEC.1.2:

Interface emanation requires not emit interface emanation enabling access to TSF data or user data.

Management:

FPT_EMSEC.1. There are no management activities foreseen.

Audit:

FPT_EMSEC.1. There are no actions defined to be auditable.

FPT_EMSEC.1: TOE Emanation.

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_EMSEC.1.1:

The TOE shall not emit **[assignment: types of emissions]** in excess of **[assignment: specified limits]** enabling access to **[assignment: list of types of TSF data]** and **[assignment: list of types of user data]**.

FPT_EMSEC.1.2:

The TSF shall ensure **[assignment: type of users]** are unable to use the following interface **[assignment: type of connection]** to gain access to **[assignment: list of types of TSF data]** and **[assignment: list of types of user data]**.

7 Security Requirements (ASE_REQ)

This section defines the security requirements for the TOE.

7.1 Definitions

7.1.1 Groups

The requirements are arranged into groups taken from [7]. Further groups are added to cover additional security functional requirements.

Table 19. SFR Groups

Group	Description
Core with Logical Channels (CoreG_LC)	The CoreG_LC contains the requirements concerning the runtime environment of the Java Card System implementing logical channels. This includes the firewall policy and the requirements related to the Java Card API. Logical channels are a Java Card specification version 2.2 feature. This group is the union of requirements from the Core (CoreG) and the Logical channels (LCG) groups defined in [PP/0305] (cf. Java Card System Protection Profile Collection [8]).
Installation (InstG)	The InstG contains the security requirements concerning the installation of post-issuance applications. It does not address card management issues in the broad sense, but only those security aspects of the installation procedure that are related to applet execution.
Applet deletion (ADELG)	The ADELG contains the security requirements for erasing installed applets from the card, a feature introduced in Java Card specification version 2.2.
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 was introduced in Java Card specification version 2.2.
Object deletion (ODELG)	The ODELG contains the security requirements for the object deletion capability. This provides a safe memory recovering mechanism. This is a Java Card specification version 2.2 feature.
Secure carrier (CarG)	The CarG group contains minimal requirements for secure downloading of applications on the card. This group contains the security requirements for preventing, in those configurations that do not support on-card static or dynamic bytecode verification, the installation of a package that has not been bytecode verified, or that has been modified after bytecode verification.
Configuration (ConfG)	This group contains security requirements related to the configuration of the TOE.
Secure Box (SecBoxG)	This group contains security requirements to separate the native code executed in the Secure Box environment from the rest of the TOE.
Modular Design (ModDesG)	This group contains security requirements concerning the modular design of the TOE.
Module Deletion (MDEL)	This group contains security requirements concerning the TOE modeule deletion functionality
OS Update	SFRs Related to NXP Proprietary OS Update feature

Table 19. SFR Groups*...continued*

Group	Description
Further Security Functional Requirements	This group contains further security requirements not covered by the PP [7].

7.1.2 Subjects

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 20. TOE Subjects

Subject	Description
S.ADEL	The applet deletion manager which also acts on behalf of the card issuer. It may be an applet ([18], §11), but its role asks anyway for a specific treatment from the security viewpoint. This subject is unique and is involved in the ADEL security policy.
S.APPLET	Any applet instance.
S.CAD	The Card Acceptance Device (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 packages and installation of applets.
S.JCRE	The runtime environment under which Java programs in a smart cards are executed.
S.JCVM	The bytecode interpreter that enforces the firewall at runtime.
S.LOCAL	Operand stack of a JCVM frame, or local variable of a JCVM frame containing an object or an array of references.
S.SD	A GlobalPlatform Security Domain representing on the card a off-card entity. This entity can be the Issuer, an Application Provider, the Controlling Authority or the Verification Authority.
S.MDEL	The module deletion manager which also acts on behalf of the card issuer. This subject is is involved in the MDEL security policy, which could be viewed as a superset of ADEL.
S.MEMBER	Any object's field, static field or array position.
S.PACKAGE	A package is a namespace within the Java programming language that may contain classes and interfaces, and in the context of Java Card technology, it defines either a user library, or one or several applets.
S.SBNativeCode	The third party native code executed via the Secure Box mechanism.
S.Customer	The subject that has the Customer Configuration Token.
S.NXP	The subject that has the NXP Configuration Token.

Table 20. TOE Subjects...*continued*

Subject	Description
S.ConfigurationMechanism	On card entity which can read and write configuration items.
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.

7.1.3 Objects

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

Table 21. TOE Objects

Objects	Description
O.APPLET	Any installed applet, its code and data.
O.CODE_PKG	The code of a package, including all linking information. On the Java Card platform, a package is the installation unit.
O.JAVAOBJECT	Java class instance or array. It should be noticed that KEYS, PIN, arrays and applet instances are specific objects in the Java programming language.
O.SB_Content	The code and data elements of the native code library residing in the Secure Box. This includes SecureBox support functionality provided by the TOE, like functionality to write into FLASH memory or execute Crypto Library code.
O.NON_SB_Content	Any code and data elements not assigned to the native code library residing in the Secure Box.
O.SB_SFR	The pool of Special Function Registers assigned to be accessible by native code residing in the Secure Box.
O.NON_SB_SFR	All Special Function Registers which are not assigned to the Secure Box. Especially the Segment Tables to configure the MMU.
O.CODE_MODULE	Contains Applets, Java code, native code, native code of a library or a combination of those. The code of O.CODE_MODULE is called via a dedicated interface. The interface can be TOE internal (if the module implements functionality of the JavaCard API) or Public (if the Module implements functionality of the JCOPX API or is accessed via APDUs). Each O.CODE_MODULE has an unique internal AID.

7.1.4 Informations

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

Table 22. TOE Informations

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

Table 22. TOE Informations*...continued*

Information	Description
I.MODULE_INVOCATION	Code execution flow when invoking code inside O.CODE_MODULE.

7.1.5 Security Attributes

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

Table 23. TOE Security attributes

Security Attributes	Description
Attack Counter	Attack Counter
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.
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.
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.
Decryption Key	Key for decrypting D.UPDATE_IMAGE.
Verification Key	Key to verify integrity of D.UPDATE_IMAGE.
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 (package) indicated in the export file.
Context	Package AID or "Java Card RE".
Currently Active Context	Package AID or "Java Card RE".
Dependent Package AID	Allows the retrieval of the Package AID and applet's version number.
LC Selection Status	Multiselectable, Non-multiselectable or "None".
LifeTime	CLEAR_ON_DESELECT or PERSISTENT.[1]
Owner	The Owner of an object is either the applet instance that created the object or the package (library) where it has been defined (these latter objects can only be arrays that initialize static fields of the package). 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 or the internal AID of a Module.
Registered Applets	The set of AID of the applet instances registered on the card.
Resident Packages	The set of AIDs of the packages already loaded on the card.
Selected Applet Context	Package AID or "None".
Sharing	Standards, SIO, Java Card RE Entry Point or global array.
Static References	Static fields of a package may contain references to objects. The Static References attribute records those references.
Address Space	Accessible memory portion.

Table 23. TOE Security attributes*...continued*

Security Attributes	Description
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.
Configuration Token verification key	The keys to verify tokens for product configuration.
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.
Security Level	Secure Communication Security Level defined in Section 10.6 of [21].
Secure Channel Protocol	Secure Channel Protocol version used.
Session Key	Secure Channel's session key.
Sequence Counter	Secure Channel Session's Sequence Counter.
Initial Chaining Vector (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].
Life-Cycle Status	Defined in Section 5.3.2 of [21].
CPU Mode	The execution mode of the CPU. Can be either user mode, system mode or firmware mode.
MMU Segment Table	Defines the memory areas which can be accessed for read / write operations or code execution if the CPU is in user mode. Further defines which of the Special Function Registers of the hardware can be accessed in user mode.
Special Function Registers	Special Function Registers allow to set operation modes of functional blocks of the hardware.
Module Presence	Presence of a particular O.CODE_MODULE inside the TOE with the values "present" or "not present".
Module AID	The AID of a Module, which when loaded are listed in tag '06' of the IDENTIFY command.
Resident Modules	The set of AIDs of the Modules already present in the card.

[1] 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.

7.1.6 Operations

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 24. TOE Operations

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_AASTORE (O.JAVAOBJECT, field)	Store into reference array component.
OP.CREATE(Sharing, LifeTime) (*)[1]	Creation of an object (new or makeTransient call).
OP.DELETE_APPLET (O.APPLET,...)	Delete an installed applet and its objects, either logically or physically.
OP.DELETE_PCKG (O.Code_PKG,...)	Delete a package, either logically or physically.
OP.DELETE_PCKG_APPLET (O.Code_PKG,...)	Delete a package and its installed applets, either logically or physically.
OP.INSTANCE_FIELD (O.JAVAOBJECT, field)	Read/Write a field of an instance of a class in the Java programming language.
OP.INVK_VIRTUAL (O.JAVAOBJECT, method, arg1,...)	Invoke a virtual method (either on a class instance or an array object).
OP.INVK_INTERFACE (O.JAVAOBJECT, method, arg1,...)	Invoke an interface method.
OP.JAVA(...)	Any access in the sense of [18], §6.2.8. It stands for one of the operations OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW, OP.TYPE_ACCESS, OP.ARRAY_LENGTH.
OP.PUT(S1, S2, I)	Transfer a piece of information I from S1 to S2.
OP.THROW(O.JAVAOBJECT)	Throwing of an object (athrow, see [18], §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.SB_ACCESS	Any read, write or execution access to a memory area.
OP.SB_ACCESS_SFR	Any read/write access to a Special Function Register.
OP.INVOKE_MODULE	Invocation of an O.CODE_MODULE. The invocation of the code is transparent to the user. In case O.CODE_MODULE has a TOE internal interface and is not present in the TOE, a secure state is preserved by throwing an exception or sending an appropriate error status word to the CAD.
OP.DELETE_MODULE	Deletion of a Module.
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.

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

7.2 Security Functional Requirements

This section defines the security functional requirements for the TOE. The permitted operations (assignment, iteration, selection and refinement) of the SFRs taken from Common Criteria [2] are printed in bold. Completed operations related to the PP [7] are additionally marked within [ ] where assignments are marked with the keyword "assignment".

7.2.1 COREG_LC Security Functional Requirements

The list of SFRs of this category are taken from the PP [7].

7.2.1.1 Firewall Policy

7.2.1.1.1 FDP_ACC.2[FIREWALL] Complete access control (FIREWALL)

Hierarchical-To: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

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

Refinement: The operations involved in the policy are:

- OP.CREATE(Sharing, LifeTime)(*),
- OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...),
- OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...),
- OP.JAVA(...),
- OP.THROW(O.JAVAOBJECT),
- OP.TYPE_ACCESS(O.JAVAOBJECT, class),
- OP.ARRAY_LENGTH(O.JAVAOBJECT, field),
- OP.ARRAY_AASTORE(O.JAVAOBJECT, field).

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

7.2.1.1.2 FDP_ACF.1[FIREWALL]: Security attribute based access control (FIREWALL)

Hierarchical-To: No other components.

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

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

Subject/Object	Security attributes
S.PACKAGE	**LC Selection Status**
S.JCVM	**Active Applets**, **Currently Active Context**
S.JCRE	**Selected Applet Context**

Subject/Object	Security attributes
O.JAVAOBJECT	**Sharing** , **Context** , **LifeTime**

].

FDP_ACF.1.2[FIREWALL]:

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[assignment:**

- R.JAVA.1 ([18], §6.2.8): S.PACKAGE may freely perform
  - OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...)
  - OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...),
  - OP.THROW(O.JAVAOBJECT),
  - OP.TYPE_ACCESS(O.JAVAOBJECT, class),

upon any O.JAVAOBJECT whose Sharing attribute has value ”JCRE entry point” or ”global array”.

- R.JAVA.2 ([18], §6.2.8): S.PACKAGE may freely perform
  - OP.ARRAY_ACCESS(O.JAVAOBJECT, field)
  - OP.INSTANCE_FIELD(O.JAVAOBJECT, field)
  - OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...)
  - OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...),
  - OP.THROW(O.JAVAOBJECT),

upon any O.JAVAOBJECT whose Sharing attribute has value ”Standard” and whose LifeTime attribute has value ”PERSISTENT” only if O.JAVAOBJECT’s Context attribute has the same value as the active context.

- R.JAVA.3 ([18], §6.2.8.10): S.PACKAGE may perform
  - OP.TYPE_ACCESS(O.JAVAOBJECT, class),

upon an O.JAVAOBJECT whose Sharing attribute has value ”SIO” only if O.JAVAOBJECT is being cast into (checkcast) or is being verified as being an instance of (instanceof) an interface that extends the Shareable interface.

- R.JAVA.4 ([18], §6.2.8.6): S.PACKAGE may perform
  - OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...),

upon an O.JAVAOBJECT whose Sharing attribute has the value ”SIO”, and whose Context attribute has the value ”Package AID”, only if the invoked interface method extends the Shareable interface and one of the following conditions applies:

1. The value of the attribute LC Selection Status of the package whose AID is ”Package AID” is ”Multiselectable”,
2. The value of the attribute LC Selection Status of the package whose AID is ”Package AID” is ”Non-multiselectable”, and either ”Package AID” is the value of the currently selected applet or otherwise ”Package AID” does not occur in the attribute Active Applets.

- R.JAVA.5: S.PACKAGE may perform
  - OP.CREATE(Sharing, LifeTime)(*)

upon O.JAVAOBJECT only if the value of the Sharing parameter is ”Standard” or ”SIO”.

- R.JAVA.6 ([18], §6.2.8.10): S.PACKAGE may freely perform
  - OP.ARRAY_ACCESS(O.JAVAOBJECT, field)
  - OP.ARRAY_LENGTH(O.JAVAOBJECT, field)

upon any O.JAVAOBJECT whose Sharing attribute has value ”global array”.

].

FDP_ACF.1.3[FIREWALL]:

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

- The subject S.JCRE can freely perform OP.JAVA(...) and OP.CREATE(Sharing, LifeTime)(*), with the exception given in FDP_ACF.1.4[FIREWALL], provided it is the Currently Active Context.
- The only means that the subject S.JCVM shall provide for an application to execute native code is the invocation of a Java Card API method (through
  - OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...),
  - OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...))

]

FDP_ACF.1.4[FIREWALL]:

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

- Any subject with OP.JAVA(...) upon an O.JAVAOBJECT whose LifeTime attribute has value ”CLEAR_ON_DESELECT” if O.JAVAOBJECT ’s Context attribute is not the same as the Selected Applet Context.
- Any subject attempting to create an object by the means of OP.CREATE(Sharing, LifeTime)(*) ”CLEAR_ON_DESELECT” LifeTime parameter if the active context is not the same as the Selected Applet Context.
- S.PACKAGE performing OP.ARRAY_AASTORE(O.JAVAOBJECT, field). of the reference of an O.JAVAOBJECT whose Sharing attribute has value ”global array” or ”Temporary JCRE entry point”.
- S.PACKAGE performing OP.PUTFIELD or OP.PUTSTATIC of the reference of an O.JAVAOBJECT whose Sharing attribute has value ”global array” or ”Temporary JCRE entry point”.

]

7.2.1.1.3 FDP_IFC.1[JCVM]: Subset information flow control (JCVM)

Hierarchical-To: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

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

7.2.1.1.4 FDP_IFF.1[JCVM]: Simple security attributes (JCVM)

Hierarchical-To: No other components.

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

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

Subject/Object	Security attributes
S.JCVM	Currently Active Context

].

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

- **An operation OP.PUT (S1, S.MEMBER, I.DATA) is allowed if and only if the Currently Active Context is "Java Card RE"**.
- **Other OP.PUT operations are allowed regardless of the Currently Active Context's value**.

].

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

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

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

7.2.1.1.5 FDP_RIP.1[OBJECTS]: Subset residual information protection (OBJECTS)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.1.6 FMT_MSA.1[JCRE]: Management of security attributes (JCRE)

Hierarchical-To: No other components.

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

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

7.2.1.1.7 FMT_MSA.1[JCVM]: Management of security attributes (JCVM)

Hierarchical-To: No other components.

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

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

7.2.1.1.8 FMT_MSA.2[FIREWALL-JCVM]: Secure security attributes (FIREWALL-JCVM)

Hierarchical-To: No other components.

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

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

7.2.1.1.9 FMT_MSA.3[FIREWALL]: Static attribute initialisation (FIREWALL)

Hierarchical-To: No other components.

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

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

FMT_MSA.3.2[FIREWALL]: The TSF shall not allow the **[assignment: none]** to specify alternative initial values to override the default values when an object or information is created.

7.2.1.1.10 FMT_MSA.3[JCVM]: Static attribute initialisation (JCVM)

Hierarchical-To: No other components.

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

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

FMT_MSA.3.2[JCVM]: The TSF shall not allow the **[assignment: none]** to specify alternative initial values to override the default values when an object or information is created.

7.2.1.1.11 FMT_SMF.1: Specification of Management Functions

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

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

].

7.2.1.1.12 FMT_SMR.1 Security roles

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

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

- **Java Card RE (JCRE)**
- **Java Card VM (JCVM)**

].

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

7.2.1.2 Application Programming Interface

The following SFRs are related to the Java Card API.

7.2.1.2.1 FCS_CKM.1 Cryptographic key generation

Hierarchical-To: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation], FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: JCOP RNG]** and specified cryptographic key sizes **[assignment: DES: 112, 168 bit, AES: 128, 192, 256 bit]** that meet the following: **[assignment: FCS_RNG.1 or FCS_RNG.1[HDT]]**.

FCS_CKM.1.1[RSA]: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: RSA key generation]** and specified cryptographic key sizes **[assignment: 512, 736, 768, 896, 1024, 1280, 1536, 1984, 2048, 4096 bit and from 2000 bit to 4096 bit in one bit steps]** that meet the following: **[assignment: FIPS 186-4]**.

FCS_CKM.1.1[ECDSA]: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: ECDSA (ECC over GF(p)) key generation]** and specified cryptographic key sizes **[assignment: 160, 192, 224, 256, 320, 384, 512 and 521 bits]** that meet the following: **[assignment: ISO/IEC 14888-3, ANSI X9.62 and FIPS 186-4]**.

FCS_CKM.1.1[PUF]: The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[assignment: key derivation function based on PUF]** and specified cryptographic key sizes **[assignment: 128 bits]** that meet the following: **[assignment: [14]]**.

AppNotes:

- FCS_CKM.1.1[RSA] or FCS_CKM.1.1[ECDSA] are applicable only if the corresponding Module for the cryptographic operation is present in the TOE.
- The functionality of FCS_CKM.1.1[RSA] and FCS_CKM.1.1[ECDSA] is provided by the Crypto Library [13].

7.2.1.2.2 FCS_CKM.4 Cryptographic key destruction

Hierarchical-To: No other components.

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

FCS_CKM.4.1: The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[assignment: physically overwriting the keys in a randomized manner]** that meets the following: **[assignment: none]**.

FCS_CKM.4.1[PUF]: The TSF shall destroy cryptographic keys derived by PUF block in accordance with a specified cryptographic key destruction method **[assignment: flushing of key registers]** that meets the following: **[assignment: none]**.

AppNote:

- FCS_CKM.4 for ECC keys is applicable only if the corresponding Module for the cryptographic operation is present in the TOE.

7.2.1.2.3 FCS_COP.1 Cryptographic operation

Following iterations of FCS_COP.1 use constants as defined in Java Card API Spec [16] and [12] where appropriate.

Hierarchical-To: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.

FCS_COP.1.1[PUF_AES]: The TSF shall perform **[assignment: decryption and encryption]** in accordance with a specified cryptographic algorithm **[assignment: AES in CBC mode]** and cryptographic key size **[assignment: 128 bits]** that meets the following: **[assignment: FIPS 197, NIST SP 800-38A]**.

FCS_COP.1.1[PUF_MAC]: The TSF shall perform **[assignment: CBC-MAC used for calculation of a PUF authentication]** in accordance with a specified cryptographic algorithm **[assignment: AES in CBC-MAC]** and cryptographic key size **[assignment: 128 bit]** that meet the following: **[assignment: FIPS 197, NIST SP 800-38A and ISO/IEC 9797-1]**.

FCS_COP.1.1[TripleDES]: The TSF shall perform **[assignment: data encryption and decryption]** in accordance with a specified cryptographic algorithm **[assignment: ALG_DES_CBC_ISO9797_M1, ALG_DES_CBC_ISO9797_M2, ALG_DES_CBC_NOPAD, ALG_DES_ECB_ISO9797_M1, ALG_DES_ECB_ISO9797_M2, ALG_DES_ECB_NOPAD]** and cryptographic key sizes **[assignment: LENGTH_DES3_2KEY, LENGTH_DES3_3KEY bit]** that meet the following: **[assignment: Java Card API Spec [16]]**.

FCS_COP.1.1[AES]: The TSF shall perform **[assignment: data encryption and decryption]** in accordance with a specified cryptographic algorithm **[assignment: ALG_AES_BLOCK_128_CBC_NOPAD, ALG_AES_BLOCK_128_CBC_NOPAD_STANDARD, ALG_AES_BLOCK_128_ECB_NOPAD, ALG_AES_CBC_ISO9797_M1, ALG_AES_CBC_ISO9797_M2, ALG_ AES_CBC_ISO9797_M2_STANDARD, ALG_AES_ECB_ISO9797_M1, ALG_AES_ECB_ISO9797_M2, ALG_AES_CTR]** and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_AES_256 bit]** that meet the following: **[assignment: Java Card API Spec [16] and JCOPX API [12]]**.

FCS_COP.1.1[RSACipher]: The TSF shall perform **[assignment: data encryption and decryption]** in accordance with a specified cryptographic algorithm **[assignment: ALG_RSA_NOPAD, ALG_RSA_PKCS1, ALG_RSA_PKCS1_OAEP]** and cryptographic key sizes **[assignment: LENGTH_RSA_2048, LENGTH_RSA_4096 and from 2000 bit to 4096 bit in one bit steps]** that meet the following: **[assignment: Java Card API Spec [16] and for the one bit step range see API specified in JCOPX [12]**.

FCS_COP.1.1[ECDHPACEKeyAgreement]: The TSF shall perform **[assignment: ECDH PACE key agreement]** in accordance with a specified cryptographic algorithm **[assignment: ALG_EC_PACE_GM, ALG_EC_SVDP_DHC_PACE]** and cryptographic key sizes **[assignment: LENGTH_EC_FP_160, LENGTH_EC_FP_192, LENGTH_EC_FP_224, LENGTH_EC_FP_256, LENGTH_EC_FP_320, LENGTH_EC_FP_384, LENGTH_EC_FP_521 and from 160 bit to 521 bit in 1 bit steps]** that meet the following: **[assignment:** Java Card API Spec [16] and for the one bit step range see API specified in JCOPX [12] **]**.

FCS_COP.1.1[PIV]: The TSF shall perform **[assignment: key establishment protocol for the PIV Card Application]** in accordance with a specified cryptographic algorithm **[assignment: One-Pass Diffie-Hellman, C(1e, 1s, ECC CDH) Scheme from SP800-56A in a manner that is based on a simplified profile of OPACITY with Zero Key Management [ANSI 504-1]]** and cryptographic key sizes **[assignment: LENGTH_EC_FP_160, LENGTH_EC_FP_192, LENGTH_EC_FP_224, LENGTH_EC_FP_256, LENGTH_EC_FP_320, LENGTH_EC_FP_384, LENGTH_EC_FP_521 and from 160 bit to 521 bit in 1 bit steps]** that meet the following: **[assignment: NIST SP 800-73-4 and JCOPX** [12]**]**.

FCS_COP.1.1[ECDH_P1363]: The TSF shall perform **[assignment: Diffie-Hellman Key Agreement]** in accordance with a specified cryptographic algorithm **[assignment: ALG_EC_SVDP_DH, ALG_EC_ SVDP_DH_KDF, ALG_EC_SVDP_DH_PLAIN, ALG_EC_SVDP_DHC, ALG_EC_SVDP_ DHC_KDF, ALG_EC_SVDP_DHC_PLAIN, ALG_EC_SVDP_DH_PLAIN_XY]** and cryptographic key sizes **[assignment: LENGTH_EC_FP_160, LENGTH_EC_FP_192, LENGTH_ EC_FP_224, LENGTH_EC_FP_256, LENGTH_EC_FP_320, LENGTH_EC_FP_384, LENGTH_EC_FP_521 and from 160 bit to 521 bit in 1 bit steps]** that meet the following: **[assignment: Java Card API Spec [16] and JCOPX API [12]]**.

FCS_COP.1.1[DESMAC]: The TSF shall perform **[assignment: MAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: Triple-DES in outer CBC for Mode ALG_DES_MAC4_ISO9797_1_M1_ALG3, ALG_DES_MAC4_ISO9797_1_M2_ALG3, ALG_DES_MAC4_ISO9797_M1, ALG_DES_MAC4_ISO9797_M2, ALG_DES_MAC8_ ISO9797_1_M1_ALG3, ALG_DES_MAC8_ISO9797_1_M2_ALG3, ALG_DES_MAC8_ISO9797_M1, ALG_DES_MAC8_ISO9797_M2, ALG_DES_MAC8_NOPAD]** and cryptographic key sizes **[assignment: LENGTH_DES3_2KEY, LENGTH_DES3_3KEY]** that meet the following: **[assignment: Java Card API Spec** [16]**]**.

FCS_COP.1.1[AESMAC]: The TSF shall perform **[assignment: 16 byte MAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: AES in CBC Mode ALG_AES_MAC_128_NOPAD, ALG_AES_MAC_128_ISO9797_1_M2_ALG3]** and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_ AES_256 bit]** that meet the following: **[assignment: Java Card API Spec** [16]**]**.

FCS_COP.1.1[RSASignaturePKCS1]: The TSF shall perform **[assignment: digital signature generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: ALG_RSA_SHA_ISO9796**[6]**, ALG_RSA_SHA_ISO9796_MR**[6]**, ALG_RSA_SHA_PKCS1**[6]**, ALG_RSA_SHA_PKCS1_PSS**[6]**, 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_**

6 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.

384_PKCS1_PSS, ALG_RSA_SHA_512_PKCS1, ALG_RSA_SHA_512_PKCS1_PSS, ALG_RSA_SHA_ISO9796, ALG_RSA_SHA_256_ISO9796 or SIG_CIPHER_RSA in combination with MessageDigest.ALG_SHA_256, MessageDigest.ALG_SHA_384, MessageDigest.ALG_SHA_512 and in combination with Cipher.PAD_PKCS1_PSS, Cipher.PAD_ISO9796 or Cipher.PAD_ISO9796_MR] and cryptographic key sizes **[assignment: LENGTH_RSA_2048, LENGTH_RSA_4096 and from 2000 bit to 4096 bit in one bit steps]** that meet the following: **[assignment: Java Card API Spec** [16] **and for the one bit step range see API specified in JCOPX [12]]**.

FCS_COP.1.1[ECSignature]: The TSF shall perform **[assignment: digital signature generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: ALG_ECDSA_SHA_ 224, ALG_ECDSA_SHA_256, ALG_ECDSA_SHA_384, ALG_ECDSA_SHA_512 or SIG_CIPHER_ECDSA or SIG_CIPHER_ECDSA_PLAIN in combination with MessageDigest.ALG_SHA_224, MessageDigest.ALG_SHA_256, MessageDigest.ALG_SHA_384 or MessageDigest.ALG_SHA_ 512]** and cryptographic key sizes **[assignment: LENGTH_EC_FP_160, LENGTH_EC_ FP_192, LENGTH_EC_FP_224, LENGTH_EC_FP_256, LENGTH_EC_FP_320, LENGTH_EC_FP_384, LENGTH_EC_FP_521 and from 160 bit to 521 bit in 1 bit steps]** that meet the following: **[assignment: Java Card API Spec [16] and JCOPX API [12]**.

FCS_COP.1.1[ModMath] The TSF shall perform **[assignment: secure modular arithmetic: addition, subtraction, reduction and multiplication]** in accordance with a specified cryptographic algorithm **[assignment: none]** and cryptographic key sizes **[assignment: none]** that meet the following: **[assignment: JCOPX API [12]]**.

FCS_COP.1.1[SHA]: The TSF shall perform **[assignment: secure hash computation]** in accordance with a specified cryptographic algorithm **[assignment: ALG_SHA**[6]**, ALG_SHA_224, ALG_ SHA_256, ALG_SHA_384, ALG_SHA_512]** and cryptographic key sizes **[assignment: LENGTH_SHA, LENGTH_SHA_224, LENGTH_SHA_256, LENGTH_SHA_384, LENGTH_SHA_512]** that meet the following: **[assignment: Java Card API Spec [16] and JCOPX API [12]]**.

FCS_COP.1.1[AES_CMAC]: The TSF shall perform **[assignment: CMAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: ALG_AES_CMAC8, ALG_ AES_CMAC16, ALG_AES_CMAC16_STANDARD, ALG_AES_CMAC_128]** and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_ AES_256 bit]** that meet the following: **[assignment: see Java Card API Spec [16] and JCOPX API [12]]**.

FCS_COP.1.1[DAP]: The TSF shall perform **[assignment: verification of the DAP signature attached to Executable Load Applications]** in accordance with a specified cryptographic algorithm **[assignment: ALG_ECDSA_SHA_256, ALG_RSA_SHA_PKCS1 and ALG_AES_CMAC16]** and cryptographic key sizes **[assignment: LENGTH_EC_FP_256, LENGTH_RSA_1024, LENGTH_AES_128, LENGTH_AES_192 and LENGTH_ AES_256 respectively]** that meet the following: **[assignment: GP Spec [21], [22], [23]]**.

AppNotes:

- Following SFRs are applicable only if the corresponding Module for the cryptographic operation is present in the TOE:
  - FCS_COP.1.1[ECDHPACEKeyAgreement],
  - FCS_COP.1.1[PIV],
  - FCS_COP.1.1[ECDH_P1363],
  - FCS_COP.1.1[ECSignature],
  - FCS_COP.1.1[ModMath],
  - FCS_COP.1.1[DAP] (for ECC).
- For resistance against attackers with High Attack Potential, the user should always refer to the guidance given by the Certification Body in the jurisdiction. The website www.keylength.com provides a good reference to recommended key lengths.

7.2.1.2.4 FCS_RNG.1 Quality metric for random numbers

Hierarchical-To: No other components.

Dependencies: No dependencies

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

- (DRG.3.1) If initialized with a random seed using a PTRNG of class PTG.2 (as defined in [26]) 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 [26]).
- (DRG.3.3) The RNG provides backward secrecy even if the current internal state is known (as defined in [26]).

]

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

- (DRG.3.4) The RNG, initialized with a random seed using a PTRNG of class PTG.2 (as defined in [26]) as random source, generates output for which 2^{48} strings of bit length 128 are mutually different with probability at least $1 - 2^{-24}$.
- (DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [26]).

]

AppNote: This functionality is provided by the certified Crypto Lib, see [13].

7.2.1.2.5 FCS_RNG.1[HDT] Quality metric for random numbers

Hierarchical-To: No other components.

Dependencies: No dependencies

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

- (DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 (as defined in [26]) as random source.
- (DRG.4.2) The RNG provides forward secrecy (as defined in [26]).
- (DRG.4.3) The RNG provides backward secrecy even if the current internal state is known (as defined in [26]).

- (DRG.4.4) The RNG provides enhanced forward secrecy on demand (as defined in [26]).
- (DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2 (as defined in [26]).

]

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

- (DRG.4.6) The RNG generates output for which 2^{48} strings of bit length 128 are mutually different with probability at least $1 - 2^{-24}$.
- (DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [26]).

]

AppNote: This functionality is provided by the certified Crypto Lib, see [13].

7.2.1.2.6 FDP_RIP.1[ABORT] Subset residual information protection (ABORT)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.2.7 FDP_RIP.1[APDU] Subset residual information protection (APDU)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.2.8 FDP_RIP.1[GlobalArray_Refined] Subset residual information protection (Global Array)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.2.9 FDP_RIP.1[bArray] Subset residual information protection (bArray)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.2.10 FDP_RIP.1[KEYS] Subset residual information protection (KEYS)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.2.11 FDP_RIP.1[TRANSIENT] Subset residual information protection (TRANSIENT)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.2.12 FDP_ROL.1[FIREWALL] Basic rollback (FIREWALL)

Hierarchical-To: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

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

FDP_ROL.1.2[FIREWALL]: The TSF shall permit operations to be rolled back within the **[assignment: scope of a select(), deselect(), process(), install() or uninstall() call, notwithstanding the restrictions given in** [18], §7.7, **within the bounds of the Commit Capacity** ([18], §7.8), **and those described in [16]]**.

7.2.1.3 Card Security Management

7.2.1.3.1 FAU_ARP.1 Security alarms

Hierarchical-To: No other components.

Dependencies: FAU_SAA.1 Potential violation analysis

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

- **throw an exception,**
- **lock the card session,**
- **reinitialize the Java Card System and its data,**
- **[assignment: response with error code to S.CAD]**

] upon detection of a potential security violation.

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

- CAP file inconsistency,
- typing error in the operands of a bytecode,
- applet life cycle inconsistency,
- card tearing (unexpected removal of the Card out of the CAD) and power failure,

- abort of a transaction in an unexpected context, (see abortTransaction(), [JCAPI3] and ([JCRE3], 7.6.2)
- violation of the Firewall or JCVM SFPs,
- unavailability of resources,
- array overflow,
- assignment:
  - EDC: checksum mismatch of EDC arrays (**throw an exception**)
  - MOD: functionality of a not present Module is invoked (**throw an exception, response with error code to S.CAD**)
  - SRE: violation of Sensitive Result integrity errors (**throw an exception**)
  - CHP: Abnormal environmental condition (Frequency, Voltage, Temperature) (**reinitialize the Java Card System**)
  - Physical Tampering
    - CLC: Card Manager Life Cycle inconsistency (**throw an exception**)
    - CHP: General Fault Injection Detection (**reinitialize the Java Card System**)
  - CHP: FLASH defects (**reinitialize the Java Card System**)
  - CHP: Integrity protected persistent data inconsistency (**reinitialize the Java Card System**),
  - CHP: Integrity protected transient data inconsistency (**reinitialize the Java Card System**),
  - Memory Access Violation
    - CHP: Others (**reinitialize the Java Card System**)
  -

Application Note:

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

App Note:

- The action ”reinitialize the Java Card System and its data” is supported by the TOE. The Java Card System is reinitialized by performing a reset. Additionally the internal

Attack Counter may be updated before the reset depending on the detected abnormal event.

- The action ”lock the card session” which is assigned in the PP is not supported by the TOE. Instead the action ”reinitialize the Java Card System and its data” is executed which is a more strict reaction.
- No particular action is taken for the potential security violation ”card tearing” since this is a normal operating condition.

7.2.1.3.2 FDP_SDI.2[DATA] Stored data integrity monitoring and action

Hierarchical-To: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies.

FDP_SDI.2.1[DATA]: The TSF shall monitor user data stored in containers controlled by the TSF for **[assignment: integrity errors]** on all objects, based on the following attributes: **[assignment: integrity protected data]**.

FDP_SDI.2.2[DATA]: Upon detection of a data integrity error, the TSF shall **[assignment: perform the action defined in FAU_ARP.1]**.

Refinement: The following data elements have the user data attribute ”integrity protected data”:

- **D.APP_KEYs**
- **D.PIN, *D.BIO***

7.2.1.3.3 FPR_UNO.1 Unobservability

Hierarchical-To: No other components.

Dependencies: No dependencies.

FPR_UNO.1.1: The TSF shall ensure that **[assignment: all users]** are unable to observe the operation **[assignment: all operations]** on **[assignment: D.APP_KEYs, D.PIN, D.BIO D.Crypto, *D.BIO*]** by **[assignment: another user]**.

7.2.1.3.4 FPT_FLS.1 Failure with preservation of secure state

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.1.3.5 FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

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

- **the rules defined in [17] specification,**

- **the API tokens defined in the export files of reference implementation,**
- **[assignment:**
  – the ISO 7816-6 rules,
  – the EMV specification]

] when interpreting the TSF data from another trusted IT product.

7.2.1.4 AID Management

7.2.1.4.1 FIA_ATD.1[AID] User attribute definition (AID)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

- **Package AID,**
- **Applet's Version Number,**
- **Registered Applets,**
- **Applet Selection Status ([18], §4.6)**

].

Refinement: "Individual users" stands for applets.

7.2.1.4.2 FIA_UID.2[AID] User identification before any action (AID)

Hierarchical-To: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

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

7.2.1.4.3 FIA_USB.1[AID] User-subject binding (AID)

Hierarchical-To: No other components.

Dependencies: FIA_ATD.1 User attribute definition

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

FIA_USB.1.2[AID]: The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **[assignment: Each uploaded package is associated with an unique Package AID]**.

FIA_USB.1.3[AID]: The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[assignment: The initially assigned Package AID is unchangeable]**.

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

Hierarchical-To: No other components.

Dependencies: FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions

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

7.2.1.4.5 FMT_MTD.3[JCRE] Secure TSF data (JCRE)

Hierarchical-To: No other components.

Dependencies: FMT_MTD.1 Management of TSF data

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

7.2.2 INSTG Security Functional Requirements

The list of SFRs of this category are taken from the PP [7]. The SFR FDP_ITC.2[INSTALLER] has been refined and is now part of the card management SFRs (FDP_ITC.2[CCM]) in Section 7.2.6 "CarG Security Functional Requirements".

7.2.2.1 FMT_SMR.1[INSTALLER] Security roles (INSTALLER)

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

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

FMT_SMR.1.2[INSTALLER]: The TSF shall be able to associate users with roles.

7.2.2.2 FPT_FLS.1[INSTALLER] Failure with preservation of secure state (INSTALLER)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1[INSTALLER]: The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the installer fails to load/install a package/ applet as described in [18], §11.1.5]**.

7.2.2.3 FPT_RCV.3[INSTALLER] Automated recovery without undue loss (INSTALLER)

Hierarchical-To: FPT_RCV.2 Automated recovery

Dependencies: AGD_OPE.1 Operational user guidance

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

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

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

FPT_RCV.3.4[INSTALLER]: The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

7.2.3 ADELG Security Functional Requirements

The list of SFRs of this category are taken from the PP [7].

7.2.3.1 FDP_ACC.2[ADEL] Complete access control (ADEL)

Hierarchical-To: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

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

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

Refinement: The operations involved in the policy are:

- OP.DELETE_APPLET,
- OP.DELETE_PCKG(O.CODE_PKG, ...),
- OP.DELETE_PCKG_APPLET(O.CODE_PKG, ...),

7.2.3.2 FDP_ACF.1[ADEL] Security attribute based access control (ADEL)

Hierarchical-To: No other components.

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

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

Subject/Object	Security Attributes
S.JCVM	Active Applets
S.JCRE	Selected Applet Context, Registered Applets, Resident Packages,
O.CODE_PKG	Package AID, Dependent Package AID, Static References
O.APPLET	Applet Selection Status
O.JAVAOBJECT	Owner

]

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

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

1. the owner of O is a registered applet instance A (O is reachable from A),
2. a static field of a resident package P contains a reference to O (O is reachable from P),
3. there exists a valid remote reference to O (O is remote reachable),
4. there exists an object O' that is reachable according to either (1) or (2) or (3) or (4) above and O' contains a reference to O (the reachability status of O is that of O').

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

- R.JAVA.14 ([18], §11.3.4.1, Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon an O.APPLET only if,
  1. S.ADEL is currently selected,
  2. there is no instance in the context of O.APPLET that is active in any logical channel and
  3. there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance distinct from O.APPLET, or O.JAVAOBJECT is reachable from a package P , or ([18], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.15 ([18], §11.3.4.1, Multiple Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon several O.APPLET only if,
  1. S.ADEL is currently selected,
  2. there is no instance of any of the O.APPLET being deleted that is active in any logical channel and
  3. there is no O.JAVAOBJECT owned by any of the O.APPLET being deleted such that either O.JAVAOBJECT is reachable from an applet instance distinct from any of those O.APPLET, or O.JAVAOBJECT is reachable from a package P or ([18], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.16 ([18], §11.3.4.2, Applet/Library Package Deletion): S.ADEL may perform OP.DELETE_PCKG(O.CODE_PKG, ...) upon an O.CODE_PKG only if,
  1. S.ADEL is currently selected,
  2. no reachable O.JAVAOBJECT, from a package distinct from O.CODE_PKG that is an instance of a class that belongs to O.CODE_PKG, exists on the card and
  3. there is no resident package on the card that depends on O.CODE_PKG.
- R.JAVA.17 ([18], §11.3.4.3, Applet Package and Contained Instances Deletion): S.ADEL may perform OP.DELETE_PCKG_APPLET(O.CODE_PKG, ...) upon an O.CODE_PKG only if,
  1. S.ADEL is currently selected,
  2. no reachable O.JAVAOBJECT, from a package distinct from O.CODE_PKG, which is an instance of a class that belongs to O.CODE_PKG, exists on the card,
  3. there is no package loaded on the card that depends on O.CODE_PKG, and
  4. for every O.APPLET of those being deleted it holds that: (i) there is no instance in the context of O.APPLET that is active in any logical channel and (ii) there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance not being deleted, or O.JAVAOBJECT is reachable from a package not being deleted, or ([18], §8.5) O.JAVAOBJECT is remote reachable.

]

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

FDP_ACF.1.4[ADEL]: The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

any subject but S.ADEL to O.CODE_PKG or O.APPLET for the purpose of deleting them from the card.

7.2.3.3 FDP_RIP.1[ADEL] Subset residual information protection (ADEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.3.4 FMT_MSA.1[ADEL] Management of security attributes (ADEL)

Hierarchical-To: No other components.

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

FMT_MSA.1.1[ADEL]: The TSF shall enforce the **[assignment: ADEL access control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: Registered Applets and Resident Packages]** to **[assignment: S.JCRE]**.

7.2.3.5 FMT_MSA.3[ADEL] Static attribute initialisation (ADEL)

Hierarchical-To: No other components.

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

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

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

7.2.3.6 FMT_SMF.1[ADEL] Specification of Management Functions (ADEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.3.7 FMT_SMR.1[ADEL] Security roles (ADEL)

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

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

FMT_SMR.1.2[ADEL]: The TSF shall be able to associate users with roles.

7.2.3.8 FPT_FLS.1[ADEL] Failure with preservation of secure state (ADEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1[ADEL]: The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the applet deletion manager fails to delete a package/ applet as described in [18], §11.3.4]**.

7.2.4 RMIG Security Functional Requirements

Not used in this ST because RMI is optional in the PP [7] and the TOE does not support RMI.

7.2.5 ODELG Security Functional Requirements

The list of SFRs of this category are taken from the PP [7].

7.2.5.1 FDP_RIP.1[ODEL] Subset residual information protection (ODEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.5.2 FPT_FLS.1[ODEL] Failure with preservation of secure state (ODEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

7.2.6 CarG Security Functional Requirements

The card management SFRs from the PP [7] are refined and replaced by the following SFRs.

7.2.6.1 FDP_UIT.1[CCM] Data exchange integrity (CCM)

Hierarchical-To: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], [FTP_ ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]

FDP_UIT.1.1[CCM]: The TSF shall enforce the **[assignment: Secure Channel Protocol information flow control policy and the Security Domain access control policy]** to **[selection: receive]** user data in a manner protected from **[selection: modification, deletion, insertion and replay]** errors.

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

7.2.6.2 FDP_ROL.1[CCM] Basic rollback (CCM)

Hierarchical-To: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FDP_ROL.1.1[CCM]: The TSF shall enforce **[assignment: Security Domain access control policy]** to permit the rollback of the **[assignment: installation operation**] on the **[assignment: executable files and application instances]**.

FDP_ROL.1.2[CCM]: The TSF shall permit operations to be rolled back within the **[assignment: boundaries of available memory before the card content management function started]**.

7.2.6.3 FDP_ITC.2[CCM] Import of user data with security attributes (CCM)

Hierarchical-To: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], [FTP_ ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path], FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1[CCM]: The TSF shall enforce the **[assignment: Security Domain access control policy and the Secure Channel Protocol information flow policy]** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2[CCM]: The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3[CCM]: The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4[CCM]: The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5[CCM]: The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **[assignment: Package loading is allowed only if, for each dependent package, its AID attribute is equal to a resident package AID attribute, the major (minor) Version attribute associated to the dependent package is lesser than or equal to the major (minor) Version attribute associated to the resident package ([17], §4.5.2)]**.

7.2.6.4 FPT_FLS.1[CCM] Failure with preservation of secure state (CCM)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1[CCM]: The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the Security Domain fails to load/install an Executable File/application instance as described in [18], Section 11.1.5]**.

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

Hierarchical-To: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1[SD]: The TSF shall enforce the **[assignment: Security Domain access control policy]** on:

[assignment:

- **Subjects: S.INSTALLER, S.ADEL, S.CAD (from [7]) and S.SD**
- **Objects: Delegation Token, DAP Block and Load File**
- **Operations: GlobalPlatform's card content management APDU commands and API methods]**.

7.2.6.6 FDP_ACF.1[SD] Security attribute based access control (SD)

Hierarchical-To: No other components.

Dependencies: FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1[SD]: The TSF shall enforce the **[assignment: Security Domain access control policy]** to objects based on the following: **[assignment:**

- **Subjects:**
  - **S.INSTALLER, defined in [7] and represented by the GlobalPlatform Environment (OPEN) on the card, the Card Life Cycle attributes (defined in Section 5.1.1 of [21])**
  - **S.ADEL, also defined in [7] 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 [21]), a Life-cycle Status (defined in Section 5.3.2 of [21]) and a Secure Communication Security Level (defined in Section 10.6 of [21])**
  - **S.CAD, defined in [7], the off-card entity that communicates with the S.INSTALLER and S.ADEL through S.SD**.
- **Objects:**
  - **The Delegation Token, in case of Delegated Management operations, with the attributes Present or Not Present**
  - **The DAP Block, in case of application loading, with the attributes Present or Not Present**
  - **The Load File or Executable File, in case of application loading, installation, extradition or registry update, with a set of intended privileges and its targeted associated SD AID**.
- **Mapping subjects/objects to security attributes:**
  - **S.INSTALLER: Security Level, Card Life Cycle, Life-cycle Status, Privileges, Resident Packages, Registered Applets**
  - **S.ADEL: Active Applets, Static References, Card Life Cycle, Life-cycle Status, Privileges, Applet Selection Status, Security Level**
  - **S.SD: Privileges, Life-cycle Status, Security Level**
  - **S.CAD: Security Level]**.

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

- **loading (Section 9.3.5 of [21])**
- **installation (Section 9.3.6 of [21])**
- **extradition (Section 9.4.1 of [21])**
- **registry update (Section 9.4.2 of [21])**
- **content removal (Section 9.5 of [21])]**.

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

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

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

Hierarchical-To: No other components.

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

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

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

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

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

Hierarchical-To: No other components.

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

FMT_MSA.3.1[SD]: The TSF shall enforce the **[assignment: Security Domain access control policy]** to provide **[selection: restrictive]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2[SD]: The TSF shall allow the **[assignment: Card Issuer or the Application Provider]** to specify alternative initial values to override the default values when an object or information is created.

Refinement: Alternative initial values shall be at least as restrictive as the default values defined in FMT_MSA.3.1[SD].

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

Hierarchical-To: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1[SD]: The TSF shall be capable of performing the following management functions: **[assignment:**

- **Management functions specified in GlobalPlatform specifications [GP]:**
  - **card locking (Section 9.6.3 of [21])**
  - **application locking and unlocking (Section 9.6.2 of [21])**
  - **card termination (Section 9.6.4 of [21])**
  - **card status interrogation (Section 9.6.6 of [21])**
  - **application status interrogation (Section 9.6.5 of [21])].**

7.2.6.10 FMT_SMR.1[SD] Security roles (SD)

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1[SD]: The TSF shall maintain the roles **[assignment: ISD, SSD]**.

FMT_SMR.1.2[SD]: The TSF shall be able to associate users with roles.

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

Hierarchical-To: FCO_NRO.1 Selective proof of origin.

Dependencies: FIA_UID.1 Timing of identification.

FCO_NRO.2.1[SC]: The TSF shall enforce the generation of evidence of origin for transmitted **[assignment: Executable load files]** at all times.

FCO_NRO.2.2[SC]: The TSF shall be able to relate the **[assignment: DAP Block]** of the originator of the information, and the **[assignment: identity]** of the information to which the evidence applies.

FCO_NRO.2.3[SC]: The TSF shall provide a capability to verify the evidence of origin of information to **[selection: originator]** given **[assignment: at the time the Executable load files are received as no evidence is kept on the card for future verification]**.

7.2.6.12 FDP_IFC.2[SC] Complete information flow control (SC)

Hierarchical-To: FDP_IFC.1 Subset information flow control

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.2.1[SC]: The TSF shall enforce the **[assignment: Secure Channel Protocol information flow control policy]** on **[assignment:**

- **the subjects S.CAD and S.SD, involved in the exchange of messages between the TOE and the CAD through a potentially unsafe communication channel,**
- **the information controlled by this policy are the card content management commands, including personalization commands, in the APDUs sent to the card and their associated responses returned to the CAD]**

and all operations that cause that information to flow to and from subjects covered by the SFP.

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

7.2.6.13 FDP_IFF.1[SC] Simple security attributes (SC)

Hierarchical-To: No other components.

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

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

- **Subjects:**
  - **S.SD receiving the Card Content Management commands (through APDUs or APIs).**
  - **S.APPLET receiving commands (through the JCOPX API [12]).**
  - **S.CAD the off-card entity that communicates with S.SD or S.APPLET.**
- **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];**
  - **any command, in case of JCOPX API [12]].**

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

- **Runtime behavior rules defined by GlobalPlatform for:**
  - **loading (Section 9.3.5 of [21]);**
  - **installation (Section 9.3.6 of [21]);**
  - **extradition (Section 9.4.1 of [21]);**
  - **registry update (Section 9.4.2 of [21]);**
  - **content removal (Section 9.5 of [21]).**
- **Runtime behavior rules defined by GlobalPlatform, implemented in JCOPX API [12]].**

FDP_IFF.1.3[SC]: The TSF shall enforce the **[assignment: no additional information flow control SFP rules]**.

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

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

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

AppNote: The subject S.SD can be the ISD or APSD.

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

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

Hierarchical-To: No other components.

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

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

- **Key Set,**

- **Security Level,**
- **Secure Channel Protocol,**
- **Session Keys,**
- **Sequence Counter,**
- **ICV]**

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

- **The Card Issuer for ISD,**
- **The Application Provider for APSD,**
- **The Applet for JCOPX API [12]]**.

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

Hierarchical-To: No other components.

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

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

FMT_MSA.3.2[SC]: The TSF shall allow the **[assignment: Card Issuer, Application Provider, Applet]** to specify alternative initial values to override the default values when an object or information is created.

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

Hierarchical-To: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1[SC]: The TSF shall be capable of performing the following management functions: **[assignment:**

- **Management functions specified in GlobalPlatform specifications [GP]:**
  - **loading (Section 9.3.5 of [21]),**
  - **installation (Section 9.3.6 of [21]),**
  - **extradition (Section 9.4.1 of [21]),**
  - **registry update (Section 9.4.2 of [21]),**
  - **content removal (Section 9.5 of [21]),**
- **Attach and retrieve sessions]**.

AppNote: All management functions related to secure channel protocols shall be relevant.

7.2.6.17 FIA_UID.1[SC] Timing of identification (SC)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FIA_UID.1.1[SC]: The TSF shall allow **[assignment:**

- **application selection,**
- **initializing a secure channel with the card,**
- **requesting data that identifies the card or the Card Issuer]**

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

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

7.2.6.18 FIA_UAU.1[SC] Timing of authentication (SC)

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1[SC]: The TSF shall allow **[assignment: the TSF mediated actions listed in FIA_UID.1[SC]]** on behalf of the user to be performed before the user is authenticated.

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

7.2.6.19 FIA_UAU.4[SC] Single-use authentication mechanisms

Hierarchical-To: No other components.

Dependencies: No dependencies.

FIA_UAU.4.1[SC]: The TSF shall prevent reuse of authentication data related to **[assignment: the authentication mechanism used to open a secure communication channel with the card]**.

7.2.6.20 FTP_ITC.1[SC] Inter-TSF trusted channel (SC)

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

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

FTP_ITC.1.3[SC]: The TSF shall initiate communication via the trusted channel for **[assignment: all card management functions:**

- **loading,**
- **installation,**
- **extradition,**
- **registry update,**
- **content removal,**
- **changing the Application Life Cycle or Card Life Cycle]**.

7.2.7 ConfG Security Functional Requirements

The list of SFRs of this category define additional requirements related to the configuration of the TOE.

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

Hierarchical-To: FDP_IFC.1 Subset information flow control

Dependencies: FDP_IFF.1 Simple security attributes

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

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

7.2.7.2 FDP_IFF.1[CFG] Simple security attributes (CFG)

Hierarchical-To: No other components.

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

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

Subject/Information	Security attributes
S.Customer	Customer Configuration Token
S.NXP	NXP Configuration Token
S.ConfigurationMechanism	NXP Configuration Access , Customer Configuration Access
D.CONFIG_ITEM	access privilege

].

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

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

].

FDP_IFF.1.3[CFG]: The TSF shall enforce the additional information flow control SFP rules **[assignment: none]**.

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

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

- **If the NXP Configuration Access is disabled then nobody can read or write D.CONFIG_ITEM.**
- **If the Customer Configuration Access is disabled then S.Customer can not read or write D.CONFIG_ITEM.**

].

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

Hierarchical-To: No other components.

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

FMT_MSA.1.1[CFG]: The TSF shall enforce the **[assignment: CONFIGURATION information flow control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: NXP Configuration Access and Customer Configuration Access]** to **[assignment: none]**.

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

Hierarchical-To: No other components.

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

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

FMT_MSA.3.2[CFG]: The TSF shall allow the **[assignment: nobody]** to specify alternative initial values to override the default values when an object or information is created.

7.2.7.5 FMT_SMR.1[CFG] Security roles (CFG)

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1[CFG]: The TSF shall maintain the roles **[assignment: S.NXP and S.Customer]**.

FMT_SMR.1.2[CFG]: The TSF shall be able to associate users with roles.

AppNote: The roles of the CONFIGURATION information flow control SFP are defined by the NXP Configuration Token and the Customer Configuration Token.

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

Hierarchical-To: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1[CFG]: The TSF shall be capable of performing the following management functions: **[assignment: none]**.

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1[CFG]	The TSF shall be capable of performing the following management functions: **[assignment: none]**.

7.2.7.7 FIA_UID.1[CFG] Timing of identification (CFG)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FIA_UID.1.1[CFG]: The TSF shall allow **[assignment: to select the ISD]** on behalf of the user to be performed before the user is identified.

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

7.2.8 SecBoxG Security Functional Requirements

The SFRs in this group provide additional requirements to separate the native code executed in the Secure Box environment from the rest of the TOE.

7.2.8.1 FDP_ACC.2[SecureBox] Complete access control (SecureBox)

Hierarchical-To: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1[SecureBox]: The TSF shall enforce the **[assignment: SecureBox access control SFP]** on **[assignment: S.SBNativeCode, O.SB_Content, O.NON_SB_Content, O.SB_SFR, O.NON_SB_SFR]** and all operations among subjects and objects covered by the SFP.

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

Refinement: The operations involved in this policy are:

- **OP.SB_ACCESS**,
- **OP.SB_ACCESS_SFR**.

7.2.8.2 FDP_ACF.1[SecureBox] Security attribute based access control (SecureBox)

Hierarchical-To: No other components.

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1[SecureBox]: The TSF shall enforce the **[assignment: SecureBox access control SFP]** to all objects based on the following: **[assignment: S.SBNativeCode, O.SB_Content, O.NON_SB_Content, O.SB_SFR, O.NON_SB_SFR and the attributes CPU Mode, the MMU Segment Table and the Special Function Registers related to system management].**

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

- **Code assigned to S.SBNativeCode is only executed in CPU Mode User Mode**.
- **Code assigned to S.SBNativeCode is only able to perform OP.SB_ACCESS to O.SB_Content. The ROM, FLASH, and RAM which belongs to O.SB_Content is controlled by the MMU Segment Table used by the Memory Management Unit**.
- **Code assigned to S.SBNativeCode is able to perform OP.SB_ACCESS_SFR to O.SB_SFR**.

].

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

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

- **For S.SBNative Code it is not possible to perform OP.SB_ACCESS to O.NON_SB_Content.**
- **For S.SBNative Code it is not possible to perform OP.SB_ACCESS_SFR to O.NON_SB_SFR.**

].

7.2.8.3 FMT_MSA.1[SecureBox] Management of security attributes (SecureBox)

Hierarchical-To: No other components.

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

FMT_MSA.1.1[SecureBox]: The TSF shall enforce the **[assignment: SecureBox access control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: CPU Mode and the MMU Segment Table]** to **[assignment: S.JCRE]**.

AppNote: The dependency with FMT_SMR.1 is not applicable. Only **S.JCRE** is allowed to modify security attributes for the Secure Box before **S.SBNativeCode** is executed.

7.2.8.4 FMT_MSA.3[SecureBox] Static attribute initialisation (SecureBox)

Hierarchical-To: No other components.

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

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

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

AppNote: The dependency with FMT_SMR.1 is not applicable. The TOE does not allow to specify alternative initial values for the security attributes of the Secure Box.

7.2.8.5 FMT_SMF.1[SecureBox] Specification of Management Functions (SecureBox)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1[SecureBox]: The TSF shall be capable of performing the following management functions: **[assignment:**

- **Switch the CPU Mode**
- **Change the values in the MMU Segment Table to assign RAM to the Secure Box**
- **Change the values in the MMU Segment Table to assign FLASH to the Secure Box**

].

7.2.9 ModDesG Security Functional Requirements

The SFRs in this group provide additional requirements related to the Modular Design of the TOE.

7.2.9.1 FDP_IFC.1[MODULAR-DESIGN] Subset information flow control (MODULAR-DESIGN)

Hierarchical-To: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1[MODULAR-DESIGN]: The TSF shall enforce the **[assignment: modular design information flow control SFP]** on **[assignment: S.APPLET, S.SD, S.JCRE, I.MODULE_INVOCATION and OP.INVOKE_MODULE]**.

7.2.9.2 FDP_IFF.1[MODULAR-DESIGN] Simple security attributes (MODULAR-DESIGN)

Hierarchical-To: No other components.

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

FDP_IFF.1.1[MODULAR-DESIGN]: The TSF shall enforce the **[assignment: modular design information flow control SFP]** based on the following types of subject and information security attributes: **[assignment: S.APPLET, S.SD, S.JCRE and I.MODULE_INVOCATION with the security attribute Module Presence of the invoked O.CODE_MODULE]**.

FDP_IFF.1.2[MODULAR-DESIGN]: The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[assignment: Operation OP.INVOKE_MODULE is allowed for S.APPLET, S.SD and S.JCRE on I.MODULE_INVOCATION if the security attribute Module Presence of the invoked O.CODE_MODULE has the value ”present”]**.

FDP_IFF.1.3[MODULAR-DESIGN]: The TSF shall enforce the additional information flow control SFP rules **[assignment: none]**.

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

FDP_IFF.1.5[MODULAR-DESIGN]: The TSF shall explicitly deny an information flow based on the following rules **[assignment: prevent access to O.CODE_MODULE if the security attribute Module Presence has the value ”not present”]**.

7.2.9.3 FIA_ATD.1[MODULAR-DESIGN] User attribute definition (MODULAR-DESIGN)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FIA_ATD.1.1[MODULAR-DESIGN]: The TSF shall maintain the following list of security attributes belonging to individual users: **[assignment:**

- **Module Presence,**
- **Package AID**

].

Refinement: ”Individual users” stands for Modules.

7.2.9.4 FIA_USB.1[MODULAR-DESIGN] User-subject binding (MODULAR-DESIGN)

Hierarchical-To: No other components.

Dependencies: FIA_ATD.1 User attribute definition

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

FIA_USB.1.2[MODULAR-DESIGN]: The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **[assignment: Each Module is associated with an unique Package AID]**.

FIA_USB.1.3[MODULAR-DESIGN]: The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[assignment: The Package AID of a Module is unchangeable]**.

AppNote: The user is a Module and the subjects are the **S.APPLET**, **S.SD** and **S.JCRE**.

7.2.9.5 FMT_MSA.1[MODULAR-DESIGN] Management of security attributes (MODULAR-DESIGN)

Hierarchical-To: No other components.

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

FMT_MSA.1.1[MODULAR-DESIGN]: The TSF shall enforce the **[assignment: MDEL access control SFP and modular design information flow control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: Module Presence** of **O.CODE_MODULE]** to **[assignment: S.MDEL]**.

7.2.9.6 FMT_MSA.3[MODULAR-DESIGN] Static attribute initialisation (MODULAR-DESIGN)

Hierarchical-To: No other components.

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

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

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

7.2.9.7 FMT_SMF.1[MODULAR-DESIGN] Specification of Management Functions (MODULAR-DESIGN)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1[MODULAR-DESIGN]: The TSF shall be capable of performing the following management functions: **[assignment: modify the list of Resident Modules]**.

7.2.9.8 FMT_SMR.1[MODULAR-DESIGN] Security roles (MODULAR-DESIGN)

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1[MODULAR-DESIGN]: The TSF shall maintain the roles: **[assignment: Module Invoker]**.

FMT_SMR.1.2[MODULAR-DESIGN]: The TSF shall be able to associate users with roles.

7.2.9.9 FPT_FLS.1[MODULAR-DESIGN] Failure with preservation of secure state (MODULAR-DESIGN)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1[MODULAR-DESIGN]: The TSF shall preserve a secure state when the following types of failures occur: **[assignment: OP.INVOKE_MODULE is performed on a TOE internal interface of O.CODE_MODULE where the security attribute Module Presence has the value ”not present”]**.

AppNote: A secure state is being preserved by throwing an exception or sending an error status word to the CAD.

7.2.9.10 FIA_UID.1[MODULAR-DESIGN] Timing of identification (MODULAR-DESIGN)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FIA_UID.1.1[MODULAR-DESIGN]: The TSF shall allow **[assignment:**

- **direct invocation of Modules with public interface and the security attribute Module Presence having the value ’present’,**
- **invocation of Modules via JavaCard API with TOE internal interface and the security attribute Module Presence having the value ’present’**

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

FIA_UID.1.2[MODULAR-DESIGN]: The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

7.2.10 Module Deletion Security Functional Requirements

The SFRs in this section provide the SFRs related to JCOP proprietary Module Deletion feature.

7.2.10.1 FDP_ACC.2[MDEL] Complete access control (MDEL)

Hierarchical-To: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

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

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

Refinement: The operations involved in the policy are:

- OP.DELETE_MODULE.

7.2.10.2 FDP_ACF.1[MDEL] Security attribute based access control (MDEL)

Hierarchical-To: No other components.

Dependencies: FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation

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

Subject/Object	Security Attributes
S.JCRE	Selected Applet Context, Registered Applets, Resident Packages, Resident Modules
O.CODE_MODULE	Module Presence, Module AID

]

FDP_ACF.1.2[MDEL]: The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[assignment: deletion of O.CODE_MODULE with a TOE internal interface is allowed even if other Resident Packages or other Resident Modules depend on it]**.

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

FDP_ACF.1.4[MDEL]: The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

any subject but S.MDEL to O.CODE_MODULE for the purpose of deleting them from the card.

7.2.10.3 FDP_RIP.1[MDEL] Subset residual information protection (ADEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FDP_RIP.1.1[MDEL]: The TSF shall ensure that any previous information content of a resource is made unavailable upon the **[selection: deallocation of the resource from]** the following objects: **[assignment: Modules when one of the deletion operations in FDP_ACC.2[MDEL] is performed on them]**.

7.2.10.4 FMT_MSA.1[MDEL] Management of security attributes (MDEL)

Hierarchical-To: No other components.

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

FMT_MSA.1.1[MDEL]: The TSF shall enforce the **[assignment: MDEL access control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: Resident Modules]** to **[assignment: S.JCRE]**.

7.2.10.5 FMT_MSA.3[MDEL] Static attribute initialisation (MDEL)

Hierarchical-To: No other components.

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

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

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

7.2.10.6 FMT_SMF.1[MDEL] Specification of Management Functions (MDEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1[MDEL]: The TSF shall be capable of performing the following management functions: **[assignment: modify the list of Resident Modules]**.

7.2.10.7 FMT_SMR.1[MDEL] Security roles (MDEL)

Hierarchical-To: No other components.

Dependencies: FIA_UID.1 Timing of identification

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

FMT_SMR.1.2[ADEL]: The TSF shall be able to associate users with roles.

7.2.10.8 FPT_FLS.1[MDEL] Failure with preservation of secure state (MDEL)

Hierarchical-To: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1[MDEL]: The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the module deletion manager fails to delete a Module]**.

7.2.11 OS Update Security Functional Requirements

The SFRs in this section relate to the proprietary JCOP OS Update feature.

7.2.11.1 FDP_IFC.2[OSU]: Complete Information flow control (OSU)

Hierarchical to:	FDP_IFC.1 Subset information flow control.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.2.1 [OSU]	The TSF shall enforce the **[assignment: OS Update information flow control SFP]** on **[assignment: S.OSU and D.UPDATE_IMAGE]**.
FDP_IFC.2.2 [OSU]	The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

7.2.11.2 FDP_IFF.1[OSU]: Simple security attributes

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
FDP_IFF.1.1 [OSU]	The TSF shall enforce the **[assignment: OS Update information flow control SFP]** based on the following types of subject and information security attributes **[assignment:** • **S.OSU: security attributes Current Sequence Number, Verification Key, Package Decryption Key** • **D.UPDATE_IMAGE: security attributes Received Sequence Number, Image Type** **]**.
FDP_IFF.1.2[OSU]	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[assignment:** • **S.OSU shall only accept D.UPDATE_IMAGE which signature can be verified with Verification Key.** • **S.OSU shall only accept D.UPDATE_IMAGE for the update process that can be decrypted with Package Decryption Key.** **]**.
FDP_IFF.1.3 [OSU]	The TSF shall enforce the additional information flow control SFP rules **[assignment: S.OSU shall only authorize**

D.UPDATE_IMAGE for the update process if the following rules apply:

- **If Image Type equals Reset then Received Sequence Number shall equal Current Sequence Number.**
- **If Image Type equals Upgrade then Received Sequence Number shall be higher than Current Sequence Number.**
- **If Image Type equals Downgrade then Received Sequence Number shall be lower than Current Sequence Number.**

].

FDP_IFF.1.4 [OSU]	The TSF shall explicitly authorise an information flow based on the following rules: **[assignment: none]**
FDP_IFF.1.5[OSU]	The TSF shall explicitly deny an information flow based on the following rules: **[assignment: D.Update_image which is not included in the pre-loaded OS Update plan]**
Application note	The on-card S.OSU role interacts with the off-card S.UpdateImageCreator via OSU commands. The D.UPDATE_IMAGE is split up into smaller chunks and transmitted as payload within the OSU Commands to the TOE.
Application note	Decrypting the D.UPDATE_IMAGE with the Package Decryption Key prevents the authorization of the D.UPDATE_IMAGE for the update process on a not certified system. The Package Decryption Key is only available on a certified TOE.

7.2.11.3 FIA_UAU.1[OSU]: Timing of authentication (OSU)

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification.
FIA_UAU.1.1 [OSU]	The TSF shall allow **[assignment: OP.TRIGGER_UPDATE]** on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2 [OSU]	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

7.2.11.4 FIA_UAU.4[OSU]: Single-use authentication mechanisms (OSU)

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

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

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

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

7.2.11.6 FMT_MSA.1 [OSU]: Management of security attributes (OSU)

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1 [OSU]	The TSF shall enforce the **[assignment: OS Update information flow control SFP]** to restrict the ability to **[selection: modify]** the security attributes **[assignment: Current Sequence Number]** to **[assignment: S.OSU]**.

7.2.11.7 FMT_MSA.3[OSU]: Static attribute initialisation (OSU)

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

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

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

7.2.11.9 FMT_SMR.1[OSU]: Security roles (OSU)

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

7.2.11.10 FPT_FLS.1[OSU]: Failure with preservation of secure state (OSU)

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1 [OSU]	The TSF shall preserve a secure state when the following types of failures occur: **[assignment:** • **Corrupted D.UPDATE_IMAGE is received.** • **Unauthorized D.UPDATE_IMAGE is received.** • **The OS Update Process is interrupted.** • **The activation of the additional code failed.** **]**.

7.2.12 Further Security Functional Requirements

The SFRs in this section provide additional proprietary features not covered by the PP [7].

7.2.12.1 FAU_SAS.1[SCP] Audit Data Storage (SCP)

Hierarchical-To: No other components.

Dependencies: No other components.

FAU_SAS.1.1[SCP]: The TSF shall provide **[assignment: test personnel before TOE Delivery]** with the capability to store the **[assignment: Initialisation Data and/ or Prepersonalisation Data and/or supplements of the SmartCard Embedded Software]** in the **[assignment: audit records]**.

7.2.12.2 FIA_AFL.1[PIN] Basic Authentication Failure Handling (PIN)

Hierarchical-To: No other components.

Dependencies: FIA_UAU.1 Timing of authentication.

FIA_AFL.1.1[PIN]: The TSF shall detect when **[selection: an administrator configurable positive integer within [1 and 127]]** unsuccessful authentication attempts occur related to **[assignment: any user authentication using D.PIN]**.

FIA_AFL.1.2[PIN]: When the defined number of unsuccessful authentication attempts has been **[selection:surpassed]**, the TSF shall **[assignment: block the authentication with D.PIN]**.

AppNote: 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.

7.2.12.3 FIA_AFL.1[BIO] Basic Authentication Failure Handling (BIO)

Hierarchical-To: No other components.

Dependencies: FIA_UAU.1 Timing of authentication.

FIA_AFL.1.1[BIO]: The TSF shall detect when **[selection: an administrator configurable positive integer within [1 and 127]]** unsuccessful authentication attempts occur related to **[assignment: any user authentication using D.BIO]**.

FIA_AFL.1.2[BIO]: When the defined number of unsuccessful authentication attempts has been **[selection:surpassed]**, the TSF shall **[assignment: block the authentication with D.BIO]**.

AppNote: 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[BIO] is organized.

7.2.12.4 FPT_EMSEC.1 TOE emanation

Hierarchical-To: No other components.

Dependencies: No dependencies.

FPT_EMSEC.1.1: The TOE shall not emit **[assignment: variations in power consumption or timing during command execution]** in excess of **[assignment: non-useful information]** enabling access to **[assignment: TSF data: D.CRYPTO]** and **[assignment: User data: D.PIN, D.APP_KEYs]**.

FPT_EMSEC.1.2: The TSF shall ensure **[assignment: that unauthorized users]** are unable to use the following interface **[assignment: electrical contacts or Radio**

Frequency (RF) field] to gain access to **[assignment: TSF data: D.CRYPTO]** and **[assignment: User data: D.PIN, D.APP_KEYs]**.

7.2.12.5 FPT_PHP.3 Resistance to physical attack

Hierarchical-To: No other components.

Dependencies: No dependencies.

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

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

AppNote: This SFR is taken from the certified Security IC Platform Protection Profile [5].

7.2.12.6 FCS_CKM.2 Cryptographic key distribution

Hierarchical-To: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1: The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[assignment: methods: set keys and components of DES, AES, RSA, RSA-CRT, ECC and secure messaging]** that meets the following: **[assignment: [16], [12]]**.

AppNote:

- The keys can be accessed as specified in [16] Key class and [12] for proprietary classes.
- This component shall be instantiated according to the version of the Java Card API applying to the security target and the implemented algorithms [16] and [12] for proprietary classes.
- FCS_CKM.2 for ECC keys is applicable only if the corresponding Module for the cryptographic operation is present in the TOE.

7.2.12.7 FCS_CKM.3 Cryptographic key access

Hierarchical-To: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.3.1: The TSF shall perform **[assignment: management of DES, AES, RSA, RSA-CRT, ECC, Diffie-Hellman and EC Diffie-Hellman]** in accordance with a specified cryptographic key access method **[assignment: methods/commands defined in packages javacard.security of [16] and [12] for proprietary classes]** that meets the following: **[assignment:** [16] and [12]**]**.

AppNote:

- The keys can be accessed as specified in [16] and [12] for proprietary classes.
- This component shall be instantiated according to the version of the Java Card API applicable to the security target and the implemented algorithms [16] and [12] for proprietary classes.
- FCS_CKM.3 for ECC keys is applicable only if the corresponding Module for the cryptographic operation is present in the TOE.

7.2.12.8 FDP_SDI.2[SENSITIVE_RESULT] Stored data integrity monitoring and action (Sensitive Result)

Hierarchical-To: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies.

FDP_SDI.2.1[SENSITIVE_RESULT]: The TSF shall monitor user data stored in containers controlled by the TSF for **[assignment: integrity errors]** on all objects, based on the following attributes: **[assignment: sensitive API result stored in the com.nxp.id.jcopx.security.SensitiveResultX class]**.

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

7.3 Security Assurance Requirements

The assurance requirements of this evaluation are EAL6 augmented by ASE_TSS.2 and ALC_FLR.1. The assurance requirements ensure, among others, the security of the TOE during its development and production.

7.3.1 ADV_SPM.1 Formal TOE security policy model

Hierarchical-To: No other components.

Dependencies: ADV_FSP.4 Complete functional specification

ADV_SPM.1.1D: The developer shall provide a formal security policy model for the **[assignment: FIREWALL access control SFP (FDP_ACC.2[FIREWALL])]**.

7.4 Security Requirements Rationale for the TOE

7.5 SFR Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FAU_ARP.1	FAU_SAA.1 Potential violation analysis	see §7.4.3.1 of [7]
FAU_SAS.1[SCP]	No dependencies	
FCO_NRO.2[SC]	FIA_UID.1 Timing of identification	FIA_UID.1[SC]
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	see §7.4.3.1 of [7]

Requirements	CC Dependencies	Satisfied Dependencies
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] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4
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.4 Cryptographic key destruction	FCS_CKM.1 FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	see §7.4.3.1 of [7]
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] FCS_CKM.4 Cryptographic key destruction	see §7.4.3.1 of [7]
FCS_RNG.1	No dependencies	
FCS_RNG.1[HDT]	No dependencies	
FDP_ACC.1[SD]	FDP_ACF.1 Security attribute based access control	FDP_ACF.1[SD]
FDP_ACC.2[FIREWALL]	FDP_ACF.1 Security attribute based access control	see §7.4.3.1 of [7]
FDP_ACC.2[ADEL]	FDP_ACF.1 Security attribute based access control	see §7.4.3.1 of [7]
FDP_ACC.2[SecureBox]	FDP_ACF.1 Security attribute based access control	FDP_ACF.1(SecureBox)
FDP_ACF.1[FIREWALL]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	see §7.4.3.1 of [7]
FDP_ACF.1[ADEL]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	see §7.4.3.1 of [7]
FDP_ACF.1[SecureBox]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.2[SecureBox] FMT_MSA.3[SecureBox]
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_IFC.1[JCVM]	FDP_IFF.1 Simple security attributes	see §7.4.3.1 of [7]
FDP_IFC.2[SC]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[SC]
FDP_IFC.2[CFG]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[CFG]
FDP_IFC.1[MODULAR-DESIGN]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[MODULAR-DESIGN]

Requirements	CC Dependencies	Satisfied Dependencies
FDP_IFF.1[JCVM]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	see §7.4.3.1 of [7]
FDP_IFF.1[SC]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.2[SC] FMT_MSA.3[SC]
FDP_IFF.1[CFG]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	FDP_IFC.2[CFG] FMT_MSA.3[CFG]
FDP_IFF.1[MODULAR-DESIGN]	FDP_IFC.1 Subset information flow control, FMT_MSA.3 Static attribute initialisation	FDP_IFC.1[MODULAR-DESIGN] FMT_MSA.3[MODULAR-DESIGN]
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[GlobalArray_Refined]	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 §7.4.3.1 of [7]
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_SDI.2[SENSITIVE_RESULT]	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_AFL.1[BIO]	FIA_UAU.1 Timing of authentication	see AppNote in FIA_AFL.1[BIO]

Requirements	CC Dependencies	Satisfied Dependencies
FIA_ATD.1[AID]	No dependencies	
FIA_ATD.1[MODULAR-DESIGN]	No dependencies	
FIA_UID.1[SC]	No dependencies	
FIA_UID.1[CFG]	No dependencies	
FIA_UID.2[AID]	No dependencies	
FIA_UID.1[MODULAR-DESIGN]	No dependencies	
FIA_USB.1[AID]	FIA_ATD.1 User attribute definition	see §7.4.3.1 of [7]
FIA_USB.1[MODULAR-DESIGN]	FIA_ATD.1 User attribute definition	FIA_ATD.1[MODULAR-DESIGN]
FIA_UAU.1[SC]	FIA_UID.1 Timing of identification	FIA_UID.1[SC]
FIA_UAU.4[SC]	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 §7.4.3.1 of [7]
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 §7.4.3.1 of [7]
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 §7.4.3.1 of [7]
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[SecureBox]	[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[SecureBox] FMT_SMR.1 FMT_SMF.1[SecureBox]
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]

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[MODULAR-DESIGN]	[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.1[MODULAR-DESIGN] FMT_SMR.1[MODULAR-DESIGN] FMT_SMF.1[MODULAR-
FMT_MSA.2[FIREWALL-JVCM	[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 §7.4.3.1 of [7]
FMT_MSA.3[FIREWALL]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.4.3.1 of [7]
FMT_MSA.3[JCVM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.4.3.1 of [7]
FMT_MSA.3[ADEL]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.4.3.1 of [7]
FMT_MSA.3[SecureBox]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SecureBox] FMT_SMR.1
FMT_MSA.3[CFG]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[CFG] FMT_SMR.1[CFG]
FMT_MSA.3[SD]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SD] FMT_SMR.1[SD]
FMT_MSA.3[SC]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SC] FMT_SMR.1[SD]
FMT_MSA.3[MODULAR-DESIGN]	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles	FMT_MSA.1[MODULAR-DESIGN] FMT_SMR.1[MODULAR-DESIGN]
FMT_MTD.1[JCRE]	FMT_SMR.1 Security roles FMT_ SMF.1 Specification of Management Functions	see §7.4.3.1 of [7]
FMT_MTD.3[JCRE]	FMT_MTD.1 Management of TSF data	see §7.4.3.1 of [7]
FMT_SMF.1	No dependencies	
FMT_SMF.1[ADEL]	No dependencies	
FMT_SMF.1[SecureBox]	No dependencies	

Requirements	CC Dependencies	Satisfied Dependencies
FMT_SMF.1[CFG]	No dependencies	
FMT_SMF.1[SD]	No dependencies	
FMT_SMF.1[SC]	No dependencies	
FMT_SMF.1[MODULAR-DESIGN]	No dependencies	
FMT_SMR.1	FIA_UID.1 Timing of identification	see §7.4.3.1 of [7]
FMT_SMR.1[INSTALLER]	FIA_UID.1 Timing of identification	see §7.4.3.1 of [7]
FMT_SMR.1[ADEL]	FIA_UID.1 Timing of identification	see §7.4.3.1 of [7]
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[MODULAR-DESIGN]	FIA_UID.1 Timing of identification	FIA_UID.1[MODULAR-DESIGN]
FPR_UNO.1	No dependencies	
FPT_EMSEC.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[CCM]	No dependencies	
FPT_FLS.1[MODULAR-DESIGN]	No dependencies	
FPT_TDC.1	No dependencies	
FPT_RCV.3[INSTALLER]	AGD_OPE.1 Operational user guidance	see §7.4.3.1 of [7]
FPT_PHP.3	No dependencies	
FTP_ITC.1[SC]	No dependencies	
ADV_SPM.1	ADV_FSP.4 Complete functional specification	ADV_FSP.4

7.5.1 OS Update SFR Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FDP_IFC.2[OSU]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[OSU]
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]
FIA_UID.1[OSU]	No dependencies	
FIA_UAU.1[OSU]	FIA_UID.1 Timing of identification	FIA_UID.1[OSU]
FIA_UAU.4[OSU]	No dependencies	

Requirements	CC Dependencies	Satisfied Dependencies
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.3[OSU]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[OSU] FMT_SMR.1[OSU]
FMT_SMF.1[OSU]	No dependencies	
FMT_SMR.1[OSU]	FIA_UID.1 Timing of identification	FIA_UID.1[OSU]
FPT_FLS.1[OSU]	No dependencies	

7.5.2 MDEL SFR Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FDP_ACC.2[MDEL]	FDP_ACF.1	FDP_ACF.1[MDEL]
FDP_ACF.1[MDEL]	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1[MDEL] FMT_MSA.3[MDEL]
FDP_RIP.1[MDEL]	No dependencies	
FMT_MSA.1[MDEL]	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	FDP_ACC.1[MDEL] FMT_SMR.1[MDEL] FMT_SMF.1[MDEL]
FMT_MSA.3[MDEL]	No dependencies	
FMT_SMF.1[MDEL]	No dependencies	
FMT_SMR.1[MDEL]	FIA_UID.1	unsupported
FPT_FLS.1[MDEL]	No dependencies	

7.5.3 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 "applet deletion manager" since it can be considered as part of the TSF.

The dependency FIA_UID.1 of FMT_SMR.1[MDEL] is unsupported. This ST does not require the identification of the "module deletion manager" since it can be considered as part of the TSF.

The dependency FIA_UID.1 of FMT_SMR.1[MODULAR-DESIGN] is unsupported. This ST does not require the identification of the "Module Invoker" 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.

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

The dependencies FMT_SMR.1 of FMT_MSA.1[SecureBox] and FMT_MSA.3[SecureBox] are unsupported. Only S.JCRE is allowed to modify security attributes for the Secure Box before S.SBNativeCode is executed. Furthermore the TOE does not allow to specify alternative initial values for the security attributes of the Secure Box.

7.6 Security Assurance Requirements Rationale

The selection of assurance components is based on the underlying PP [7]. The Security Target uses the augmentations from the PP, chooses EAL6 and adds the components ASE_TSS.2 and ALC_FLR.1.

The rationale for the augmentations is the same as in the PP.

The assurance level EAL6 is an elaborated pre-defined level of the CC, Part 3 [3]. 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 EAL6. Therefore, the components ASE_TSS.2 and ALC_FLR.1 add additional assurance to EAL6, but the mutual support of the requirements is still guaranteed.

8 TOE summary specification (ASE_TSS)

8.1 Introduction

The Security Functions (SF) introduced in this section realize the SFRs of the TOE. See Table 25 "Overview of Security Functions" for list of all Security Functions. Each SF consists of components spread over several TOE components to provide a security functionality and fulfill SFRs.

8.2 Security Functionality

Table 25. Overview of Security Functions

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.PUF	User Data Protection using PUF
SF.OM	Java Object Management
SF.MM	Memory Management
SF.PIN	PIN Management
SF.BIO	Biometric Template Management
SF.PERS_MEM	Persistent Memory Management
SF.EDC	Error Detection Code API
SF.HW_EXC	Hardware Exception Handling
SF.PID	Platform Identification
SF.SMG_NSC	No Side-Channel
SF.ACC_SBX	Secure Box
SF.MOD_INVOC	Module Invocation
SF.SENS_RES	Sensitive Result
SF.OSU	OS Update
SF.MOD_DEL	Module Deletion

8.2.1 SF.JVCM: 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 [18], [17]. This fulfills the SFRs FDP_IFC.1[JCVM], FDP_IFF.1[JCVM], FMT_SMF.1, FMT_SMR.1, FDP_ROL.1[FIREWALL], FDP_ACF.1[FIREWALL], FDP_ACC.2[FIREWALL] and FIA_UID.2[AID]. SF.JCVM supports FAU_ARP.1 and FPT_FLS.1 by throwing Java Exceptions according to these specifications. Additionally it supports these SFRs by verification of the integrity of used Java object headers.

Security attributes in SF.JCVM are separated from user data and not accessible by applets to fulfill FMT_MSA.1[JCRE] and FMT_MSA.1[JCVM]. All values for security attributes are initialized and assigned by the system itself which fulfills FMT_MSA.2[FIREWALL-JCVM], FMT_MSA.3[FIREWALL] , and FMT_MSA.3[JCVM].

SF.JCVM ensures together with SF.PERS_MEM that the system is halted in case non existing Java objects could be referenced after an aborted transaction to fulfill FDP_RIP.1[ABORT].

8.2.2 SF.CONFIG: Configuration Management

SF.CONFIG provides means to store Initialization Data and Pre-personalization Data before TOE delivery FAU_SAS.1[SCP].

SF.CONFIG provides means to change configuration items of the card. Some configuration items can be changed by the customer and some can only be changed by NXP (FDP_IFC.2[CFG], FDP_IFF.1[CFG] , FMT_MSA.3[CFG], FMT_MSA.1[CFG], FMT_SMR.1[CFG], FMT_SMF.1[CFG], FIA_UID.1[CFG]). SF.CONFIG supports FCS_COP.1 by configuring the behavior of cryptographic operations.

8.2.3 SF.OPEN: Card Content Management

SF.OPEN provides the card content management functionality according the GlobalPlatform Specification [20]. This supports FCO_NRO.2[SC], FDP_ACC.1[SD], FDP_ACF.1[SD], FDP_UIT.1[CCM], FDP_IFC.2[SC], FDP_IFF.1[SC], FDP_IFC.2[SC], FIA_UID.1[SC], FIA_UID.2[AID], FIA_USB.1[AID], FMT_MSA.1[SC], FMT_MSA.1[SD], FMT_MSA.3[SC], FMT_MSA.3[SD], FMT_SMF.1[ADEL] , FMT_SMR.1[SD], FMT_SMF.1[SC], FMT_SMF.1[SD], FTP_ITC.1[SC], FMT_MSA.3[ADEL], FMT_SMR.1[INSTALLER], FMT_SMR.1[ADEL], FDP_ITC.2[CCM], FDP_ROL.1[CCM], FIA_UAU.1[SC], FIA_UAU.4[SC], and FTP_ITC.1[SC]. In addition to the GP specification, the Java Card Runtime Environment specification [18] is followed to support FDP_ACC.2[ADEL], FDP_ACF.1[ADEL], FMT_MSA.3[SC], FMT_MSA.3[SD], FMT_MTD.1[JCRE], FMT_MTD.3[JCRE], FPT_FLS.1[INSTALLER], FDP_RIP.1[bArray], FDP_RIP.1[ADEL], FPT_TDC.1, FPT_FLS.1[ADEL], and FPT_FLS.1[CCM] for application loading, installation, and deletion.

AID management is provided by SF.OPEN according to the GlobalPlatform Specification [20], the Java Card Runtime Environment Specification [18], and the Java Card API Specification [16] to support FIA_ATD.1[AID].

SF.OPEN is part of the TOE runtime environment and thus separated from other applications to fulfill FMT_MSA.1[ADEL]. It supports FAU_ARP.1 and FPT_FLS.1 by responding with error messages according to the GlobalPlatform mapping guidelines [24] and fulfills FPT_RCV.3[INSTALLER] by inherent memory cleanup in case of aborted loading and installation.

8.2.4 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 [16] to fulfill FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4, and FCS_COP.1. 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 to support FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4, FDP_RIP.1[KEYS], and

FDP_SDI.2[DATA]. The Crypto Lib certified with the TOE hardware supports FCS_COP.1 and FPR_UNO.1.

8.2.5 SF.RNG: Random Number Generator

SF.RNG provides secure random number generation to fulfill FCS_CKM.1 and FCS_RNG.1. Random numbers are generated by the Crypto Lib certified with the TOE hardware. SF.RNG provides an API according to the Java Card API Specification [16] to generate random numbers according to FCS_RNG.1.

8.2.6 SF.DATA_STORAGE: Secure Data Storage

SF.DATA_STORAGE provides a secure data storage for confidential data. It is used to store cryptographic keys (supports FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, and FCS_CKM.4) and to store PINs (supports FIA_AFL.1[PIN]). Supports storage of Biometric templates, supporting FIA_AFL.1[BIO] All data stored by SF.DATA_STORAGE is CRC32 integrity protected to fulfill FDP_SDI.2[DATA], FAU_ARP.1, and FPT_FLS.1. The stored data is AES encrypted to fulfill FPR_UNO.1.

8.2.7 SF.PUF: User Data Protection using PUF

SF.PUF implements a mechanism to seal/unseal the user data stored in shared memory against unintended disclosure. SF.PUF encrypts/decrypts the user data with a cryptographic key which is derived from the PUF data and stored directly in the hardware. SF.PUF calculates a MAC as a PUF authentication value. SF.PUF serves to seal/unseal the user data stored in the memory. The user data stored in the memory can be encrypted/decrypted using the PUF block. A MAC (message authentication code) can be calculated as a PUF authentication value. Hence, the user data can be sealed within the TOE and can be solely unsealed by the TOE. The cryptographic key for sealing/unsealing of the user data is generated with the help of a key derivation function based on the PUF block and the Random Number Generator (RNG). The PUF block provides the PUF data to the key derivation function and thereby the cryptographic key is derived. If the TOE is powered off, the PUF data is not available from the PUF block. Therefore SF.PUF is suitable to meet FCS_CKM.1.1[PUF] and FCS_CKM.4.1[PUF]. The encryption/decryption of user data and the calculation of a MAC as a PUF authentication value are performed within the AES coprocessor. Therefore SF.PUF is suitable to meet FCS_COP.1.1[PUF_AES] and FCS_COP.1.1[PUF_MAC].

8.2.8 SF.OM: Java Object Management

SF.OM provides the object management for Java objects which are processed by SF.JCVM. It provides object creation (FDP_RIP.1[OBJECTS]) and garbage collection according to the Java Card Runtime Environment Specification [18] to fulfill FDP_RIP.1[ODEL] and FPT_FLS.1[ODEL]. SF.OM throws an Java Exception in case an object cannot be created as requested due to too less available memory. This fulfills FAU_ARP.1 and FPT_FLS.1.

8.2.9 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 [18]. Thus, it fulfills FDP_RIP.1[TRANSIENT] by granting access to and erasing of CLEAR_ON_RESET and CLEAR_ ON_DESELECT transient arrays. It supports

FIA_ATD.1[AID] when using logical channels and it fulfills FDP_RIP.1[APDU], FDP_RIP.1[bArray] and FDP_RIP.1[GlobalArray_Refined] by clearing the APDU buffers for new incoming data, by clearing the bArray during application installation and preventing applications to keep a pointer to global arrays.

8.2.10 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 [16] and the GlobalPlatform Specification [21]. Thus, it fulfills FDP_SDI.2[DATA], FIA_AFL.1[PIN], and FPR_UNO.1.

8.2.11 SF.BIO: Biometric Template Management

SF.BIO provides secure Biometric Template management by using SF.DATA_STORAGE for Biometric Template objects specified in the Java Card API Specification [16] and the GlobalPlatform Specification [21]. Thus, it fulfills FDP_SDI.2[DATA], FIA_AFL.1[BIO], and FPR_UNO.1.

8.2.12 SF.PERS_MEM: Persistent Memory Management

SF.PERS_MEM provides atomic write operations and transaction management according to the Java Card Runtime Environment Specification [18]. This supports FAU_ARP.1, FPT_FLS.1, and FDP_ROL.1[FIREWALL].

SF.PERS_MEM supports FDP_RIP.1[ABORT] together with 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 to fulfill FAU_ARP.1 and FPT_FLS.1.

8.2.13 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 [12]. The API throws a Java Exception in case the checksum in invalid. This supports FAU_ARP.1 and FPT_FLS.1.

8.2.14 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 to fulfill FAU_ARP.1 and FPT_FLS.1, as well as to increase the attack counter in order to resist physical manipulation and probing to fulfill FPT_PHP.3.

8.2.15 SF.PID: Platform Identification

SF.PID provides a platform identifier. This platform identifier is generated during the card image generation. The platform identifier contains IDs for:

- NVM content (stored during romizing)
- Patch Level (stored during romizing, can be changed during personalization if patch is loaded)
- ROM code (stored during romizing)
- ROM code checksum (stored during romizing or during first TOE boot).

It identifies unambiguously the NVM and ROM part of the TOE. This feature supports FAU_SAS.1[SCP] by using initialization data that is used for platform identification.

8.2.16 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) (see FPR_UNO.1 and FPT_EMSEC.1).

8.2.17 SF.ACC_SBX: Secure Box

SF.ACC_SBX provides an environment to securely execute non-certified native code from third parties. SF.ACC_SBX ensures that only program code and data contained in the secure box can be accessed from within this secure box and therefore cannot harm, manipulate, or influence other parts of the TOE. This fulfills the SFRs FDP_ACC.2[SecureBox], FDP_ACF.1[SecureBox] and FMT_MSA.1[SecureBox].

Native code executed in the Secure Box is executed in User Mode. Access to the CPU mode, memory outside the Secure Box, the MMU segment table, and Special Function Registers which allow configuration of the MMU and allow System Management is prohibited for code executed in the Secure Box to fulfill FDP_ACF.1[SecureBox].

The MMU segment table to configure the MMU is part of the Secure Box which fulfils FMT_MSA.3[SecureBox]. This MMU segment table can be modified during the prepersonalization in accordance with FMT_MSA.3[SecureBox] to specify alternative settings for initially restrictive values for the MMU segment table. This supports FMT_SMF.1[SecureBox].

8.2.18 SF.MOD_INVOC: Module Invocation

SF.MOD_INVOC limits the invocation of code inside a Module to such Modules whose security attribute Module Presence has the restrictive default value ”present”. This fulfils the FMT_SMF.1[MODULAR-DESIGN], FMT_SMR.1[MODULAR-DESIGN], FMT_MSA.3[MODULAR-DESIGN] and FIA_UID.1[MODULAR-DESIGN]. Limiting the invocation to defined subjects S.APPLET, S.SD and S.JCRE fulfils the FDP_IFC.1[MODULAR-DESIGN] and FDP_IFF.1[MODULAR-DESIGN].

Throwing an exception in cases where the security attribute Module Presence has the value ”not present” fulfils FPT_FLS.1[MODULAR-DESIGN]. Deletion of a module may only be performed by S.ADEL which fulfils FMT_MSA.1[MODULAR-DESIGN]. The Modules are identified by their associated unique AIDs, which fulfils FIA_ATD.1[MODULAR-DESIGN] and FIA_USB.1[MODULAR-DESIGN] .

8.2.19 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. This fulfills FDP_SDI.2[SENSITIVE_RESULT].

8.2.20 SF.OSU: OS Update

SF.OSU provides secure functionality to update the JCOP 4.5 OS or UpdaterOS itself with an image created by a trusted off-card entity (FMT_SMR.1[OSU], FMT_SMF.1[OSU]). SF.OSU allows an authenticated OSU command (FIA_UAU.4[OSU]) to upload an integrity and confidentiality protected update image to update to another operating system version(FDP_IFC.2[OSU], FDP_IFF.1[OSU]). User authentication is based on the verification of signed OSU commands to fulfill FIA_UAU.1[OSU] and FIA_UID.1[OSU]. Integrity protection of OSU commands uses ECDSA, SHA-256 and CRC verification to fulfill FDP_IFF.1[OSU]. Confidentiality of the update image is ensured by ECDH and AES encryption to fulfill FDP_IFF.1[OSU].

SF.OSU ensures that the system stays in a secure state in case of invalid or aborted update procedures to fulfill FPT_FLS.1[OSU] and ensures that the information identifying the currently running OS is modified and the updated code is activated only after successfull OS Update procedure FMT_MSA.3[OSU], FMT_MSA.1[OSU].

8.2.21 SF.MOD_DEL: Module Deletion

Deletion of a module may only be performed by S.MDEL which fulfils FMT_MSA.1[MODULAR-DESIGN].

The Module Deletion access control policy is realized by the SFRs: FDP_ACC.2[MDEL], FDP_ACF.2[MDEL], FDP_RIP.1[MDEL], FMT_MSA.1[MDEL], FMT_MSA.3[MDEL], FMT_SMF.1[MDEL], FMT_SMR.1[MDEL] and FPT_FLS.1[MDEL]

8.3 Protection against Interference and Logical Tampering

The protection of the TOE against Interference and Logical Tampering is implemented in software within the TOE and supported by the hardware of the micro controller.

The software protection of the TOE makes use of software security services which allow to detect and react on manipulation of the TOE. Two types of reactions are used: If invalid data from outside the TOE is detected then it is assumed that the TOE was used in a wrong way. This is indicated by an appropriate Status Word or Exception. Detected deviations from the physical operating conditions and inconsistencies of internal states and program flow however are considered to be an attack to the TOE. In such cases an internal Attack Counter is increased.

Typical software security mechanisms implemented in the TOE are e.g.:

- Complex patterned values are used instead of boolean values which are sensible to tampering (only one bit needs to be changed to manipulate a *false* into a *true*.
- Small random delays are inserted in the program flow to make successful physical interfering more difficult.
- Secret information like Keys or PINs are stored encrypted in the TOE. The Masterkey to decrypt these is not accessible during normal operation.
- Critical data is read after it has been written to non volatile memory.
- Enhanced cryptographic support is based on the certified Crypto Lib for DES, AES, RSA, ECC and random number generation.
- Critical values (like PINs) are compared timing-invariant. This prevents from side channel attacks.

Further protection against Tampering and Logical Interference is realized by the MMU implemented in hardware. The MMU is able to perform access control to all types of memory and the special functions registers depending on the current operation.

The TOE defines several MMU contexts which restrict access to card internal resources. The standard context used for normal operation has no access to the cryptographic coprocessor. The context for cryptographic operation has no access to the communication interfaces. One special context has write access to the Master Key in the TOE. Afterwards the Master Keys can only be read, but only from a dedicated context which is used to decrypt keys stored in the secure data store. In all other contexts the Master Key is not accessible.

Additionally Interference and Logical Tampering is prevented by hardware security services. JCOP 4 OS runs on a certified smart card HW platform which protects against bypass by physical and logical means such as:

- cryptographic coprocessors (for symmetric and asymmetric cryptography) protected against DPA and Differential Fault Analysis (DFA),
- enhanced security sensors for clock frequency range, low and high temperature sensor, supply voltage sensors Single Fault Injection (SFI) attack detection, light sensors, and
- encryption of data stored in persistent and transient memory.

8.4 Protection against Bypass of Security Related Actions

The TOE prevents bypassing security related actions by several software counter measures. Different mechanism are used depending on the software environment.

Generally all input parameter are validated and in case of incorrect parameters the program flow is interrupted. Such event is indicated by an appropriate Status Word or Exception. This prevents the TOE from being attacked by undefined or unauthorized commands or data.

Basic protection is contributed by implementation of following standards within the TOE:

- Java Applets are separated from each other as defined in the Java Card specifications [16], [18], [17]. The separation is achieved by implementation of the firewall which prevents Applets to access data belonging to a different Java Card context. Information sharing between different contexts is possible by supervision of the well defined Java Card Firewall mechanism implemented in the TOE.
- Access to security relevant Applications in the TOE (like Security Domains) is protected by the Secure Channel mechanism defined by GlobalPlatform [21]. The secure channel allows access to Applications only if the secret keys are known. Further protection implemented in the TOE prevent brute force attacks on the secret keys of the Secure Channel.

The following mechanisms ensure that it is not possible to access information from the Java Layer without being authorized to do so.

- Status informations like Life Cycle of Applets or the Authentication State of a Secure Channel are stored in complex patterned values which protects them from manipulation.
- Correct order of Java Card Byte Code execution is ensured by the Virtual Machine which detects if Byte Code of a wrong context is executed.
- Correct processing of Byte Codes is ensured by checking at the beginning and end of Byte Code execution that the same Byte Code is executed.

Execution of native code in the TOE is protected by following mechanisms:

- Critical execution paths of the TOE functionality are protected by program flow and call tree protection. This ensures that it is not possible to bypass security relevant checks and verifications.
- Critical conditions are evaluated twice. This ensures that physical attacks on the compared values are detected during security relevant checks and verifications.
- The true case in if-conditions leads to the less critical program flow or to an error case. This prevents attacks on the program flow during security relevant checks and verifications.
- At the exit of critical loops it is checked that the whole loop was processed. This prevents from manipulation of the program flow and jumping out of the loop.
- Critical parameters are checked for consistency. This prevents from attacks with manipulated parameters.

9 Glossary

AID Application Identifier, an ISO-7816 data format used for unique identification of Java Card applications (and certain kinds of files in card file systems). The Java Card platform uses the AID data format to identify applets and packages. AIDs are administered by the International Standards Organization (ISO), so they can be used as unique identifiers.

AIDs are also used in the security policies (see 'Context' below): applets' AIDs are related to the selection mechanisms, packages' AIDs are used in the enforcement of the firewall. Note: although they serve different purposes, they share the same name space.

APDU buffer The APDU buffer is the buffer where the messages sent (received) by the card depart from (arrive to). The JCRE owns an APDU object (which is a JCRE Entry Point and an instance of the javacard.framework.APDU class) that encapsulates APDU messages in an internal byte array, called the APDU buffer. This object is made accessible to the currently selected applet when needed, but any permanent access (out-of selection-scope) is strictly prohibited for security reasons.

applet The name is given to a Java Card technology-based user application. An applet is the basic piece of code that can be selected for execution from outside the card. Each applet on the card is uniquely identified by its AID.

applet deletion manager The on-card component that embodies the mechanisms necessary to delete an applet or library and its associated data on smart cards using Java Card technology.

context A context is an object-space partition associated to a package. Applets within the same Java technology-based package belong to the same context. The firewall is the boundary between contexts (see "current context").

current context The JCRE keeps track of the current Java Card System context (also called "the active context"). When a virtual method is invoked on an object, and a context switch is required and permitted, the current context is changed to correspond to the context of the applet that owns the object. When that method returns, the previous context is restored. Invocations of static methods have no effect on the current context. The current context and sharing status of an object together determine if access to an object is permissible.

currently selected applet The applet has been selected for execution in the current session. The Java Card RE keeps track of the currently selected Java Card applet. Upon receiving a SELECT command from the CAD or PCD with this applet's AID, the Java Card RE makes this applet the currently selected applet over the I/O interface that received the command. The Java Card RE sends all further APDU commands received over each interface to the currently selected applet on this interface ([18], Glossary).

default applet The applet that is selected after a card reset ([18], §4.1).

installer The installer is the on-card application responsible for the installation of applets on the card. It may perform (or delegate) mandatory security checks according to the card issuer policy (for bytecode-verification, for instance), loads and link packages (CAP file(s)) on the card to a suitable form for the Java Card VM to execute the code they contain. It is a subsystem of what is usually called "card manager"; as such, it can be seen as the portion of the card manager that belongs to the TOE.

The installer has an AID that uniquely identifies him, and may be implemented as a Java Card applet. However, it is granted specific privileges on an implementation-specific manner ([18], §10).

interface A special kind of Java programming language class, which declares methods, but provides no implementation for them. A class may be declared as being the implementation of an interface, and in this case must contain an implementation for each of the methods declared by the interface (See also shareable interface).

Java Card RE The runtime environment under which Java programs in a smart card are executed. It is in charge of all the management features such as applet lifetime, applet isolation, object sharing, applet loading, applet initializing, transient objects, the transaction mechanism and so on.

Java Card RE Entry Point An object owned by the Java Card RE context but accessible by any application. These methods are the gateways through which applets request privileged Java Card RE services: the instance methods associated to those objects may be invoked from any context, and when that occurs, a context switch to the Java Card RE context is performed.

There are two categories of Java Card RE Entry Point Objects: Temporary ones and Permanent ones. As part of the firewall functionality, the Java Card RE detects and restricts attempts to store references to these objects.

Java Card RMI Java Card Remote Method Invocation is the Java Card System version 2.2 and 3 Classic Edition mechanism enabling a client application running on the CAD platform to invoke a method on a remote object on the card. Notice that in Java Card System, version 2.1.1, the only method that may be invoked from the CAD is the process method of the applet class and that in Java Card System, version 3 Classic Edition, this functionality is optional.

Java Card System Java Card System includes the Java Card RE, the Java Card VM, the Java Card API and the installer.

Java Card VM The embedded interpreter of bytecodes. The Java Card VM is the component that enforces separation between applications (firewall) and enables secure data sharing.

logical channel A logical link to an application on the card. A new feature of the Java Card System, version 2.2 and 3 Classic Edition, that enables the opening of simultaneous sessions with the card, one per logical channel. Commands issued to a specific logical channel are forwarded to the active applet on that logical channel. Java Card platform, version 2.2.2 and 3 Classic Edition, enables opening up to twenty logical channels over each I/O interface (contacted or contactless).

NVRAM Non-Volatile Random Access Memory, a type of memory that retains its contents when power is turned off.

object deletion The Java Card System version 2.2 and 3 Classic Edition mechanism ensures that any unreferenced persistent (transient) object owned by the current context is deleted. The associated memory space is recovered for reuse prior to the next card reset.

PCD Proximity Coupling Device. The PCD is a contactless card reader device.

PICC Proximity Card. The PICC is a card with contactless capabilities.

Secure Box The Secure Box is a construct which allows to run non certified third party native code and ensures that this code cannot harm, influence or manipulate the JCOP operating system or any of the applets executed by the operating system.

Secure Box Native Library The Secure Box Native Library is non certified third party native code running in the Secure Box.

shareable interface An interface declaring a collection of methods that an applet accepts to share with other applets. These interface methods can be invoked from an applet in a context different from the context of the object implementing the methods, thus “traversing” the firewall.

SIO An object of a class implementing a shareable interface.

subject An active entity within the TOE that causes information to flow among objects or change the system’s status. It usually acts on the behalf of a user. Objects can be active and thus are also subjects of the TOE.

transient object An object whose contents are not preserved across CAD sessions. The contents of these objects are cleared at the end of the current CAD session or when a card reset is performed. Writes to the fields of a transient object are not affected by transactions.

user Any application interpretable by the Java Card RE. That also covers the packages. The associated subject(s), if applicable, is (are) an object(s) belonging to the javacard.framework.applet class.

10 Acronyms

3DES Data Encryption Standard with 3 keys.

AES Advanced Encryption Standard.

AES CCM AES in Counter with CBC-MAC mode.

AP Application Provider.

APSD Application Provider Security Domain.

CAD Card Acceptance Device.

CRT Chinese Remainder Theorem.

Crypto Lib Crypto Library.

DAP Data Authentication Pattern.

DFA Differential Fault Analysis.

DPA Differential Power Analysis.

ECC Elliptic Curve Cryptography.

ECDAA Elliptic Curve Direct Anonymous Attestation.

GP GlobalPlatform.

HKDF HMAC based Key Derivation Function.

HMAC Keyed-Hash Message Authentication Code.

ICV Initial Chaining Vector.

ISD Issuer Security Domain.

MC FW Micro Controller Firmware.

MMU Memory Management Unit.

OS Operating System.

OSP Organizational Security Policy.

PKCC Public Key Crypto Coprocessor.

PP Protection Profile.

RAM Random Access Memory.

RF Radio Frequency.

ROM Read Only Memory.

RSA Rivest Shamir Adleman asymmetric algorithm.

SCP Smart Card Platform. It is comprised of the integrated circuit, the operating system and the dedicated software of the smart card.

SD Security Domain.

SFR Security Functional Requirement.

SPD Security Problem Definition.

SSD Supplementary Security Domain.

VASD Verification Authority Security Domain.

11 Bibliography

11 . 1 Evaluation documents

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, CCMB-2017-04-001, April 2017.

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, CCMB-2017-04-002, April 2017.

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

[4] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, CCMB-2017-04-004, April 2017.

[5] 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.

[6] (U)SIM Java Card Platform Protection Profile ,Basic and SCWS Configurations 17 June 2010, Version 2.0.2 .

[7] Java Card System - Open Configuration Protection Profile, December 2017, Version 3.0.5, published by Oracle, Inc. (BSI-CC-PP-0099-2017).

[8] Java Card Protection Profile Collection, Version 1.0b, August 2003, registered and certified by the French certification body (ANSSI) under the following references: [PP/0303] “Minimal Configuration”, [PP/0304] “Standard 2.1.1 Configuration”, [PP/0305] “Standard 2.2 Configuration” and [PP/0306] “Defensive Configuration”.

[9] ICAO. Common criteria protection profile, machine readable travel document with ICAO application, basic access control, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference bsi-cc-pp-0055, rev 1.10, 25 march 2009.

[10] Joint Interpretation Library. Joint Interpretation Library, Security requirements for post-delivery code loading, Draft Version 1.0, February 2016.

[11] ANSSI. Application Note. Security Requirements for Post-Delivery Code Loading, Version 2.0, 23 January 2015, ANSSI-CC-NOTE-06/2.0.

11 . 2 Developer documents

[12] JCOP 4.5 P71, User manual for JCOP 4.5 P71, User Guidance and Administrator Manual, NXP Semiconductors, Rev. 1.8 – 2023-04-03.

[13] NXP Secure Smart Card Controller N7122 with IC Dedicated Software and Crypto Library (R1/R2/R3), Security Target, NXP Semiconductors, Release 1.8, 1 December 2023.

[14] NXP. PUF Key derivation function specification, NXP Semiconductors, BUID, 2014.

[15] NXP Semiconductors, https://www.docstore.nxp.com.

11 . 3 Standards

[16] Oracle. Java Card 3 Platform, Application Programming Interface, Classic Edition, Version 3.0.5.

[17] Oracle. Java Card 3 Platform, Virtual Machine Specification, Classic Edition, Version 3.0.5.

[18] Oracle. Java Card 3 Platform, Runtime Environment Specification, Classic Edition, Version 3.0.5.

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

[20] GlobalPlatform. GlobalPlatform Card Specification 2.3.0, GPC_SPE_034, GlobalPlatform Inc., Oct 2015.

[21] GlobalPlatform. GlobalPlatform Card Specification 2.3.1, GPC_SPE_034, GlobalPlatform Inc., Mar 2018.

[22] GlobalPlatform. GlobalPlatform Technology Executable Load File Upgrade - Version 1.1, March 2018.

[23] GlobalPlatform. GlobalPlatform Technology Secure Element Management Service - Version 1.0, March 2018.

[24] GlobalPlatform Card Mapping Guidelines of Existing GP v2.1.1 Implementation on v2.2.1, January 2011.

[25] GlobalPlatform. GlobalPlatform Technology APDU Transport over SPI/I2C, Version 1.0, January 2020

[26] Bundesamt fuer Sicherheit in der Informationstechnik. AIS20/31: A proposal for: Functionality classes for random number generators, Bundesamt für Sicherheit in der Informationstechnik (BSI), Version 2.1, 2. December 2011.

[27] FIPS PUB 140-2: Security Requirements for Cryptographic Modules, Federal Information Processing Standards Publication 140-2, US Department of Commerce/ National Institute of Standards and Technology, 25. May 2001.

[28] FIPS PUB 186-4: Digital Signature Standard (DSS), US Department of Commerce/ National Institute of Standards and Technology, July 2013.

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

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

[31] NIST SP 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, National Institute of Standards and Technology, May 2013.

[32] NIST SP 800-73-4: Interfaces for Personal Identity Verification - Part 2: PIV Card Application Card Command Interface, National Institute of Standards and Technology, May 2015.

[33] ISO/IEC 14888-3: IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms, International Organization for Standardization, March 2016.

[34] ISO/IEC 9797-1: IT Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher, International Organization for Standardization, 1999.

[35] RFC 5869: HMAC-based Extract-and-Expand Key Derivation Function (HKDF), Request For Comments, May 2010.

[36] ANSI X9.62: Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA), American Standard for Financial Services, November 2005.

[37] ANSI/INCITS 504-1: Information Technology - Generic Identity Command Set - Part 1: Card Application Command Set, American National Standards Institute, April 2013.

[38] TPM Rev. 2.0: Trusted Platform Module Library Specification, Family "2.0", Level 00, Revision 01.07 , March 2014.

[39] ISO 7816-3: Part 3: Cards with contacts - Electrical interface and transmission protocols, November 2006.

[40] ISO/IEC 14443-4 Cards and security devices for personal identification - Contactless proximity objects - Part 4: Transmission protocol, July 2008.

[41] NXP. UM10204, I2C-bus specification and user manual, Rev. 6, 4 April 2014.

[42] NXP Semiconductors. NXP SE05x T=1 Over SPI/I2C Specification, doc. no. um11225, rev. 1.1, January 8 2020.

12 Legal information

12.1 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.

12.2 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.

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.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

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.

12.3 Licenses

ICs with DPA Countermeasures functionality

NXP ICs containing functionality implementing countermeasures to Differential Power Analysis and Simple Power Analysis are produced and sold under applicable license from Cryptography Research, Inc.

12.4 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.

MIFARE — is a trademark of NXP B.V.

Tables

Tab. 1. ST Reference and TOE reference 3
Tab. 2. Reference to Certified Micro Controller 8
Tab. 3. TOE Life Cycle phases 11
Tab. 4. Platform ID 13
Tab. 5. IDENTIFY fields 13
Tab. 6. Certified Firmware Configurations 14
Tab. 7. Delivery Items 14
Tab. 8. CarG SFRs refinements 20
Tab. 9. 23
Tab. 10. 23
Tab. 11. 24
Tab. 12. 25
Tab. 13. 25
Tab. 14. 26
Tab. 15. 28
Tab. 16. 28
Tab. 17. SA.MODULAR-DESIGN 29
Tab. 18. SA.MODULE-INVOCATION 29
Tab. 19. SFR Groups 68
Tab. 20. TOE Subjects 69
Tab. 21. TOE Objects 70
Tab. 22. TOE Informations 70
Tab. 23. TOE Security attributes 71
Tab. 24. TOE Operations 73
Tab. 25. Overview of Security Functions 124

Figures

Fig. 1. Components of the TOE 4
Fig. 2. TOE Life Cycle within Product Life Cycle 11

Contents

1 ST Introduction (ASE_INT) 3
1.1 ST Reference and TOE Reference 3
1.2 TOE Overview 3
1.2.1 TOE Components 3
1.2.2 JCOP components 4
1.2.3 Usage and Major Security Features of the TOE 5
1.2.4 TOE Type 7
1.2.5 Required non-TOE Hardware/Software/ Firmware 7
1.3 TOE Description 8
1.3.1 TOE Components and Composite Certification 8
1.3.1.1 Micro Controller 8
1.3.1.2 Security IC Dedicated Software 8
1.3.1.3 Security IC Embedded Software 9
1.3.1.4 Excluded functionality 10
1.3.2 Optional TOE Functionality 10
1.3.3 TOE Life Cycle 10
1.3.4 TOE Identification 13
1.3.4.1 TOE Delivery 14
1.3.5 Evaluated Package Types 14
2 Conformance Claims (ASE_CCL) 15
2.1 CC Conformance Claim 15
2.2 Package Claim 15
2.3 PP Claim 15
2.4 Conformance Claim Rationale 15
2.4.1 TOE Type 15
2.4.2 SPD Statement 16
2.4.2.1 Threats 16
2.4.2.2 Organizational Security Policies 17
2.4.2.3 Assumptions 17
2.4.3 Security Objectives Statement 18
2.4.4 Security Functional Requirements Statement 20
3 Security Aspects 23
3.1 Confidentiality 23
3.2 Integrity 23
3.3 Unauthorized Execution 24
3.4 Bytecode Verification 25
3.5 Card Management 25
3.6 Services 26
3.7 Config Applet 28
3.8 OS Update 28
3.9 Modular Design 29
3.9.1 Modular Design 29
3.9.2 Module Invocation 29
4 Security Problem Definition (ASE_SPD) 30
4.1 Assets 30
4.1.1 User Data 30
4.1.2 TSF Data 31
4.1.3 Biometric Templates 32
4.2 Threats 32
4.2.1 Confidentiality 32
4.2.1.1 T.CONFID-APPLI-DATA[REFINED]: Confidentiality of Application Data 32
4.2.1.2 T.CONFID-JCS-CODE: Confidentiality of Java Card System Code 32
4.2.1.3 T.CONFID-JCS-DATA: Confidentiality of Java Card System Data 32
4.2.2 Integrity 32
4.2.2.1 T.INTEG-APPLI-CODE: Integrity of Application Code 32
4.2.2.2 T.INTEG-APPLI-CODE.LOAD: Integrity of Application Code - Load 32
4.2.2.3 T.INTEG-APPLI-DATA[REFINED]: Integrity of Application Data 32
4.2.2.4 T.INTEG-APPLI-DATA.LOAD: Integrity of Application Data - Load 33
4.2.2.5 T.INTEG-JCS-CODE: Integrity of Java Card System Code 33
4.2.2.6 T.INTEG-JCS-DATA: Integrity of Java Card System Data 33
4.2.3 Identity Usurpation 33
4.2.3.1 T.SID.1: Subject Identification 1 33
4.2.3.2 T.SID.2: Subject Identification 2 33
4.2.4 Unauthorized Execution 33
4.2.4.1 T.EXE-CODE.1: Code Execution 1 33
4.2.4.2 T.EXE-CODE.2: Code Execution 2 33
4.2.4.3 T.NATIVE: Native Code Execution 33
4.2.4.4 T.MODULE_EXEC: Code Execution of Modules 34
4.2.5 Denial of Service 34
4.2.5.1 T.RESOURCES: Consumption of Resources 34
4.2.6 Card Management 34
4.2.6.1 T.UNAUTHORIZED_CARD_MNGT: Unauthorized Card Management 34
4.2.6.2 T.COM_EXPLOIT: Communication Channel Remote Exploit 34
4.2.6.3 T.LIFE_CYCLE: Life Cycle 34
4.2.7 Services 35
4.2.7.1 T.OBJ-DELETION: Object Deletion 35
4.2.8 Miscellaneous 35
4.2.8.1 T.PHYSICAL: Physical Tampering 35
4.2.9 Operating System 35
4.2.9.1 T.OS_OPERATE: Incorrect Operating System Behavior 35
4.2.10 Configuration Module 36
4.2.10.1 T.CONFIG: Unauthorized configuration 36
4.2.11 Secure Box 36
4.2.11.1 T.SEC_BOX_BORDER: SecureBox Border Infringement 36
4.2.12 Module replacement 36
4.2.12.1 T.MODULE_REPLACEMENT: Replacement of Module 36
4.2.13 OS Update 36
4.2.13.1 T.CONFID-UPDATE-IMAGE.LOAD: Confidentiality of update Image - Load 36
4.2.13.2 T.INTEG-UPDATE-IMAGE.LOAD: Integrity of update Image -Load 36

4.2.13.3 T.UNAUTH-UPDATE-IMAGE.LOAD: Load an unauthorized update 36
4.2.13.4 T.INTERRUPT_OSU: OS Update procedure interrupted 37
4.3 Organisational Security Policies 37
4.3.1 OSP.VERIFICATION: File Verification 37
4.3.2 OSP.PROCESS-TOE: Identification of the TOE 37
4.3.3 OSP.KEY-CHANGE: Security Domain Keys Change 37
4.3.4 OSP.SECURITY-DOMAINS: Security Domains 37
4.3.5 OSP.SECURE-BOX: Secure Box Border 37
4.4 Assumptions 37
4.4.1 A.APPLET: Applets without Native Methods 37
4.4.2 A.VERIFICATION: Bytecode Verification 38
4.4.3 A.USE_DIAG: Usage of TOE's Secure Communication Protocol by OE 38
4.4.4 A.USE_KEYS: Protected Storage of Keys Outside of TOE 38
4.4.5 A.PROCESS-SEC-IC: Protection during Packaging, Finishing and Personalisation 38
4.4.6 A.APPS-PROVIDER: Application Provider 38
4.4.7 A.VERIFICATION-AUTHORITY: Verification Authority 39
5 Security Objectives 40
5.1 Security Objectives for the TOE 40
5.1.1 Identification 40
5.1.1.1 OT.SID: Subject Identification 40
5.1.1.2 OT.SID_MODULE: Subject Identification of Modules 40
5.1.2 Execution 40
5.1.2.1 OT.FIREWALL: Firewall 40
5.1.2.2 OT.GLOBAL_ARRAYS_CONFID: Confidentiality of Global Arrays 40
5.1.2.3 OT.GLOBAL_ARRAYS_INTEG: Integrity of Global Arrays 40
5.1.2.4 OT.NATIVE: Native Code 40
5.1.2.5 OT.OPERATE: Correct Operation 40
5.1.2.6 OT.REALLOCATION: Secure Re-Allocation 40
5.1.2.7 OT.RESOURCES: Resources availability 41
5.1.2.8 OT.SENSITIVE_RESULTS_INTEG: Sensitive Result 41
5.1.3 Services 41
5.1.3.1 OT.ALARM: Alarm 41
5.1.3.2 OT.CIPHER: Cipher 41
5.1.3.3 OT.RND: Random Numbers Generation 41
5.1.3.4 OT.KEY-MNGT: Key Management 41
5.1.3.5 OT.PIN-MNGT: Pin Management 41
5.1.3.6 OT.BIO-MNGT: Biometric Template Management 42
5.1.3.7 OT.TRANSACTION: Transaction 42
5.1.4 Object Deletion 42
5.1.4.1 OT.OBJ-DELETION: Object Deletion 42
5.1.5 Applet Management 42
5.1.5.1 OT.APPLI-AUTH: Application Authentication 42
5.1.5.2 OT.DOMAIN-RIGHTS: Domain Rights 42
5.1.5.3 OT.COMM_AUTH: Communication Mutual Authentication 42
5.1.5.4 OT.COMM_INTEGRITY: Communication Request Integrity 43
5.1.5.5 OT.COMM_CONFIDENTIALITY: Communication Request Confidentiality 43
5.1.6 Card Management 43
5.1.6.1 OT.CARD-MANAGEMENT: Card Management 43
5.1.7 Smart Card Platform 44
5.1.7.1 OT.SCP.IC IC: Physical Protection 44
5.1.7.2 OT.SCP.RECOVERY: SCP Recovery 44
5.1.7.3 OT.SCP.SUPPORT: SCP Support 44
5.1.7.4 OT.IDENTIFICATION: TOE identification 44
5.1.8 Secure Box 44
5.1.8.1 OT.SEC_BOX_FW: SecureBox firewall 44
5.1.9 Configuration Module 44
5.1.9.1 OT.CARD-CONFIGURATION: Card Configuration 44
5.1.10 OS Update 45
5.1.10.1 OT.CONFID-UPDATE-IMAGE.LOAD: Confidentiality of Update Image 45
5.1.10.2 OT.AUTH-LOAD-UPDATE-IMAGE: Authorization of Update Image - Load 45
5.1.10.3 OT.SECURE_LOAD_ACODE: Secure loading of Additional Code 45
5.1.10.4 OT.SECURE_ACTIVATION_ADDITIONAL_CODE: Secure activation of the Additional Code 45
5.1.10.5 OT.TOE_IDENTIFICATION: Secure identification of the TOE 45
5.2 Security Objectives for the Operational Environment 45
5.2.1 OE.VERIFICATION: Bytecode Verification 45
5.2.2 OE.CODE-EVIDENCE: Code Evidence 46
5.2.3 OE.APPS-PROVIDER: Application Provider 46
5.2.4 OE.VERIFICATION-AUTHORITY: Verification Authority 46
5.2.5 OE.KEY-CHANGE: Security Domain Key Change 46
5.2.6 OE.SECURITY-DOMAINS: Security Domains 46
5.2.7 OE.USE_DIAG: Secure TOE communication protocols 46
5.2.8 OE.USE_KEYS: Protection of OPE keys 47
5.2.9 OE.PROCESS_SEC_IC: Protection during composite product manufacturing 47
5.2.10 OE.CONFID-UPDATE-IMAGE.CREATE: Confidentiality of Update Image - CREATE 47
5.3 Security Objectives Rationale 47
5.3.1 Threats 51
5.3.1.1 Confidentiality 51
5.3.1.2 Integrity 53
5.3.1.3 Identity Usurpation 56
5.3.1.4 Unauthorized Excecution 57
5.3.1.5 Denial of Service 58
5.3.1.6 Card Management 59
5.3.1.7 Services 59
5.3.1.8 Miscellaneous 60

5.3.1.9 Operating System 60
5.3.1.10 Random Numbers 60
5.3.1.11 Configuration Module 61
5.3.1.12 Secure Box 61
5.3.1.13 Module replacement 61
5.3.1.14 OS Update 61
5.3.2 Organisational Security Policies 62
5.3.2.1 OSP.VERIFICATION 62
5.3.2.2 OSP.PROCESS-TOE 63
5.3.2.3 OSP.KEY-CHANGE 63
5.3.2.4 OSP.SECURITY-DOMAINS 63
5.3.2.5 OSP.SECURE-BOX 63
5.3.3 Assumptions 63
5.3.3.1 A.APPLET 63
5.3.3.2 A.VERIFICATION 63
5.3.3.3 A.USE_DIAG 64
5.3.3.4 A.USE_KEYS 64
5.3.3.5 A.PROCESS-SEC-IC 64
5.3.3.6 A.APPS-PROVIDER 64
5.3.3.7 A.VERIFICATION-AUTHORITY 64
6 Extended Components Definition (ASE_ECD) 65
6.1 Definition of Family ”Audit Data Storage (FAU_SAS)” 65
6.2 Definition of Family ”TOE emanation (FPT_EMSEC)” 66
7 Security Requirements (ASE_REQ) 68
7.1 Definitions 68
7.1.1 Groups 68
7.1.2 Subjects 69
7.1.3 Objects 70
7.1.4 Informations 70
7.1.5 Security Attributes 71
7.1.6 Operations 72
7.2 Security Functional Requirements 74
7.2.1 COREG_LC Security Functional Requirements 74
7.2.1.1 Firewall Policy 74
7.2.1.2 Application Programming Interface 79
7.2.1.3 Card Security Management 85
7.2.1.4 AID Management 88
7.2.2 INSTG Security Functional Requirements 89
7.2.2.1 FMT_SMR.1[INSTALLER] Security roles (INSTALLER) 89
7.2.2.2 FPT_FLS.1[INSTALLER] Failure with preservation of secure state (INSTALLER) 89
7.2.2.3 FPT_RCV.3[INSTALLER] Automated recovery without undue loss (INSTALLER) 89
7.2.3 ADELG Security Functional Requirements 90
7.2.3.1 FDP_ACC.2[ADEL] Complete access control (ADEL) 90
7.2.3.2 FDP_ACF.1[ADEL] Security attribute based access control (ADEL) 90
7.2.3.3 FDP_RIP.1[ADEL] Subset residual information protection (ADEL) 91
7.2.3.4 FMT_MSA.1[ADEL] Management of security attributes (ADEL) 92
7.2.3.5 FMT_MSA.3[ADEL] Static attribute initialisation (ADEL) 92
7.2.3.6 FMT_SMF.1[ADEL] Specification of Management Functions (ADEL) 92
7.2.3.7 FMT_SMR.1[ADEL] Security roles (ADEL) 92
7.2.3.8 FPT_FLS.1[ADEL] Failure with preservation of secure state (ADEL) 92
7.2.4 RMIG Security Functional Requirements 93
7.2.5 ODELG Security Functional Requirements 93
7.2.5.1 FDP_RIP.1[ODEL] Subset residual information protection (ODEL) 93
7.2.5.2 FPT_FLS.1[ODEL] Failure with preservation of secure state (ODEL) 93
7.2.6 CarG Security Functional Requirements 93
7.2.6.1 FDP_UIT.1[CCM] Data exchange integrity (CCM) 93
7.2.6.2 FDP_ROL.1[CCM] Basic rollback (CCM) 94
7.2.6.3 FDP_ITC.2[CCM] Import of user data with security attributes (CCM) 94
7.2.6.4 FPT_FLS.1[CCM] Failure with preservation of secure state (CCM) 94
7.2.6.5 FDP_ACC.1[SD] Subset access control (SD) 94
7.2.6.6 FDP_ACF.1[SD] Security attribute based access control (SD) 95
7.2.6.7 FMT_MSA.1[SD] Management of security attributes (SD) 96
7.2.6.8 FMT_MSA.3[SD] Static attribute initialisation (SD) 96
7.2.6.9 FMT_SMF.1[SD] Specification of Management Functions (SD) 96
7.2.6.10 FMT_SMR.1[SD] Security roles (SD) 97
7.2.6.11 FCO_NRO.2[SC] Enforced proof of origin (SC) 97
7.2.6.12 FDP_IFC.2[SC] Complete information flow control (SC) 97
7.2.6.13 FDP_IFF.1[SC] Simple security attributes (SC) 97
7.2.6.14 FMT_MSA.1[SC] Management of security attributes (SC) 98
7.2.6.15 FMT_MSA.3[SC] Static attribute initialisation (SC) 99
7.2.6.16 FMT_SMF.1[SC] Specification of Management Functions (SC) 99
7.2.6.17 FIA_UID.1[SC] Timing of identification (SC) 99
7.2.6.18 FIA_UAU.1[SC] Timing of authentication (SC) 100
7.2.6.19 FIA_UAU.4[SC] Single-use authentication mechanisms 100
7.2.6.20 FTP_ITC.1[SC] Inter-TSF trusted channel (SC) 100
7.2.7 ConfG Security Functional Requirements 100
7.2.7.1 FDP_IFC.2[CFG] Complete information flow control (CFG) 101
7.2.7.2 FDP_IFF.1[CFG] Simple security attributes (CFG) 101
7.2.7.3 FMT_MSA.1[CFG] Management of security attributes (CFG) 102
7.2.7.4 FMT_MSA.3[CFG] Static attribute initialisation (CFG) 102
7.2.7.5 FMT_SMR.1[CFG] Security roles (CFG) 102

7.2.7.6 FMT_SMF.1[CFG] Specification of Management Functions (CFG) 102
7.2.7.7 FIA_UID.1[CFG] Timing of identification (CFG) 103
7.2.8 SecBoxG Security Functional Requirements 103
7.2.8.1 FDP_ACC.2[SecureBox] Complete access control (SecureBox) 103
7.2.8.2 FDP_ACF.1[SecureBox] Security attribute based access control (SecureBox) 103
7.2.8.3 FMT_MSA.1[SecureBox] Management of security attributes (SecureBox) 104
7.2.8.4 FMT_MSA.3[SecureBox] Static attribute initialisation (SecureBox) 104
7.2.8.5 FMT_SMF.1[SecureBox] Specification of Management Functions (SecureBox) 105
7.2.9 ModDesG Security Functional Requirements 105
7.2.9.1 FDP_IFC.1[MODULAR-DESIGN] Subset information flow control (MODULAR-DESIGN) 105
7.2.9.2 FDP_IFF.1[MODULAR-DESIGN] Simple security attributes (MODULAR-DESIGN) 105
7.2.9.3 FIA_ATD.1[MODULAR-DESIGN] User attribute definition (MODULAR-DESIGN) 106
7.2.9.4 FIA_USB.1[MODULAR-DESIGN] User-subject binding (MODULAR-DESIGN) 106
7.2.9.5 FMT_MSA.1[MODULAR-DESIGN] Management of security attributes (MODULAR-DESIGN) 106
7.2.9.6 FMT_MSA.3[MODULAR-DESIGN] Static attribute initialisation (MODULAR-DESIGN) .. 106
7.2.9.7 FMT_SMF.1[MODULAR-DESIGN] Specification of Management Functions (MODULAR-DESIGN) 107
7.2.9.8 FMT_SMR.1[MODULAR-DESIGN] Security roles (MODULAR-DESIGN) 107
7.2.9.9 FPT_FLS.1[MODULAR-DESIGN] Failure with preservation of secure state (MODULAR-DESIGN) 107
7.2.9.10 FIA_UID.1[MODULAR-DESIGN] Timing of identification (MODULAR-DESIGN) 107
7.2.10 Module Deletion Security Functional Requirements 108
7.2.10.1 FDP_ACC.2[MDEL] Complete access control (MDEL) 108
7.2.10.2 FDP_ACF.1[MDEL] Security attribute based access control (MDEL) 108
7.2.10.3 FDP_RIP.1[MDEL] Subset residual information protection (ADEL) 108
7.2.10.4 FMT_MSA.1[MDEL] Management of security attributes (MDEL) 109
7.2.10.5 FMT_MSA.3[MDEL] Static attribute initialisation (MDEL) 109
7.2.10.6 FMT_SMF.1[MDEL] Specification of Management Functions (MDEL) 109
7.2.10.7 FMT_SMR.1[MDEL] Security roles (MDEL) ... 109
7.2.10.8 FPT_FLS.1[MDEL] Failure with preservation of secure state (MDEL) 109
7.2.11 OS Update Security Functional Requirements 110
7.2.11.1 FDP_IFC.2[OSU]: Complete Information flow control (OSU) 110
7.2.11.2 FDP_IFF.1[OSU]: Simple security attributes .. 110
7.2.11.3 FIA_UAU.1[OSU]: Timing of authentication (OSU) 111
7.2.11.4 FIA_UAU.4[OSU]: Single-use authentication mechanisms (OSU) 111
7.2.11.5 FIA_UID.1[OSU]: Timing of Identification (OSU) 112
7.2.11.6 FMT_MSA.1 [OSU]: Management of security attributes (OSU) 112
7.2.11.7 FMT_MSA.3[OSU]: Static attribute initialisation (OSU) 112
7.2.11.8 FMT_SMF.1[OSU]: Specification of Management Functions (OSU) 113
7.2.11.9 FMT_SMR.1[OSU]: Security roles (OSU) 113
7.2.11.10 FPT_FLS.1[OSU]: Failure with preservation of secure state (OSU) 113
7.2.12 Further Security Functional Requirements 113
7.2.12.1 FAU_SAS.1[SCP] Audit Data Storage (SCP) 114
7.2.12.2 FIA_AFL.1[PIN] Basic Authentication Failure Handling (PIN) 114
7.2.12.3 FIA_AFL.1[BIO] Basic Authentication Failure Handling (BIO) 114
7.2.12.4 FPT_EMSEC.1 TOE emanation 114
7.2.12.5 FPT_PHP.3 Resistance to physical attack 115
7.2.12.6 FCS_CKM.2 Cryptographic key distribution ...115
7.2.12.7 FCS_CKM.3 Cryptographic key access 115
7.2.12.8 FDP_SDI.2[SENSITIVE_RESULT] Stored data integrity monitoring and action (Sensitive Result) 116
7.3 Security Assurance Requirements 116
7.3.1 ADV_SPM.1 Formal TOE security policy model 116
7.4 Security Requirements Rationale for the TOE 116
7.5 SFR Dependencies 116
7.5.1 OS Update SFR Dependencies 121
7.5.2 MDEL SFR Dependencies 122
7.5.3 Rationale for Exclusion of Dependencies 122
7.6 Security Assurance Requirements Rationale 123
8 TOE summary specification (ASE_TSS) 124
8.1 Introduction 124
8.2 Security Functionality 124
8.2.1 SF.JVCM: Java Card Virtual Machine 124
8.2.2 SF.CONFIG: Configuration Management 125
8.2.3 SF.OPEN: Card Content Management 125
8.2.4 SF.CRYPTO: Cryptographic Functionality 125
8.2.5 SF.RNG: Random Number Generator 126
8.2.6 SF.DATA_STORAGE: Secure Data Storage .. 126
8.2.7 SF.PUF: User Data Protection using PUF 126
8.2.8 SF.OM: Java Object Management 126
8.2.9 SF.MM: Memory Management 126
8.2.10 SF.PIN: PIN Management 127
8.2.11 SF.BIO: Biometric Template Management 127

8.2.12 SF.PERS_MEM: Persistent Memory Management 127
8.2.13 SF.EDC: Error Detection Code API 127
8.2.14 SF.HW_EXC: Hardware Exception Handling 127
8.2.15 SF.PID: Platform Identification 127
8.2.16 SF.SMG_NSC: No Side-Channel 128
8.2.17 SF.ACC_SBX: Secure Box 128
8.2.18 SF.MOD_INVOC: Module Invocation 128
8.2.19 SF.SENS_RES: Sensitive Result 128
8.2.20 SF.OSU: OS Update 129
8.2.21 SF.MOD_DEL: Module Deletion 129
8.3 Protection against Interference and Logical Tampering 129
8.4 Protection against Bypass of Security Related Actions 130
9 Glossary 132
10 Acronyms 135
11 Bibliography 137
11.1 Evaluation documents 137
11.2 Developer documents 137
11.3 Standards 137
12 Legal information 140

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

Date of release: 11 December 2023