JCOP 3 EMV P60

Security Target Lite

Rev. 3.1 – 2016-05-10 — Evaluation documentation

Final — Public

NSCIB-CC-15-67351

Document Information

Info	Content
Keywords	ASE, JCOP, Common Criteria, EAL5 augmented
Abstract	This document contains information to fulfill the requirements of the Common Criteria component ASE for the Evaluation of the JCOP 3 EMV P60 developed and provided by NXP Semiconductors, Business Unit Security & Connectivity, according to the Common Criteria for Information Technology Security Evaluation Version 3.1 at EAL5 augmented

Rev	Date	Description
0.1	2015-01-25	Initial Version - Extracted from Security Target v0.6
3.0	2016-04-20	Final Version - Extracted from Security Target v3.0
3.1	2016-05-10	References updated

1 ST Introduction (ASE_INT)

1.1 ST Reference and TOE Reference

Title	JCOP 3 EMV P60 Security Target
Version	Revision 3.1
Date	2016-05-10
Product Type	Java Card
TOE name	JCOP 3 EMV P60
Certification ID	NSCIB-CC-15-67351
CC version	Common Criteria for Information Technology Security Evaluation Version 3.1, Revision 4, September 2012 (Part 1 [2], Part 2 [3] and Part 3 [4])

Tab. 1.1: ST Reference and TOE reference

1.2 TOE Overview

The TOE consists of the Micro Controller and a software stack which is stored on the Micro Controller and which can be executed by the Micro Controller. The software stack can be further split into the following components:

- Firmware for booting and low level functionality of the Micro Controller, called MC FW including MIFARE Implementations.
- Software for implementing cryptographic operations on the Micro Controller, called Crypto Lib.
- Software for implementing a Java Card Virtual Machine [26], a Java Card Runtime Environment [25] and a Java Card Application Programming Interface [24], called JCVM, JCRE and JCAPI.
- Software for implementing content management according to GlobalPlatform [20], called GP Framework.

The TOE is also referred to as JCOP 3. Whereas the JCOP 3 OS consists of the software stack without the Crypto Lib, the MC FW and the MIFARE Implementations. The TOE uses one or more communication interfaces to communicate with its environment.
The complete TOE is depicted in Figure 1.1. The elements are described in more detail in Section 1.3.

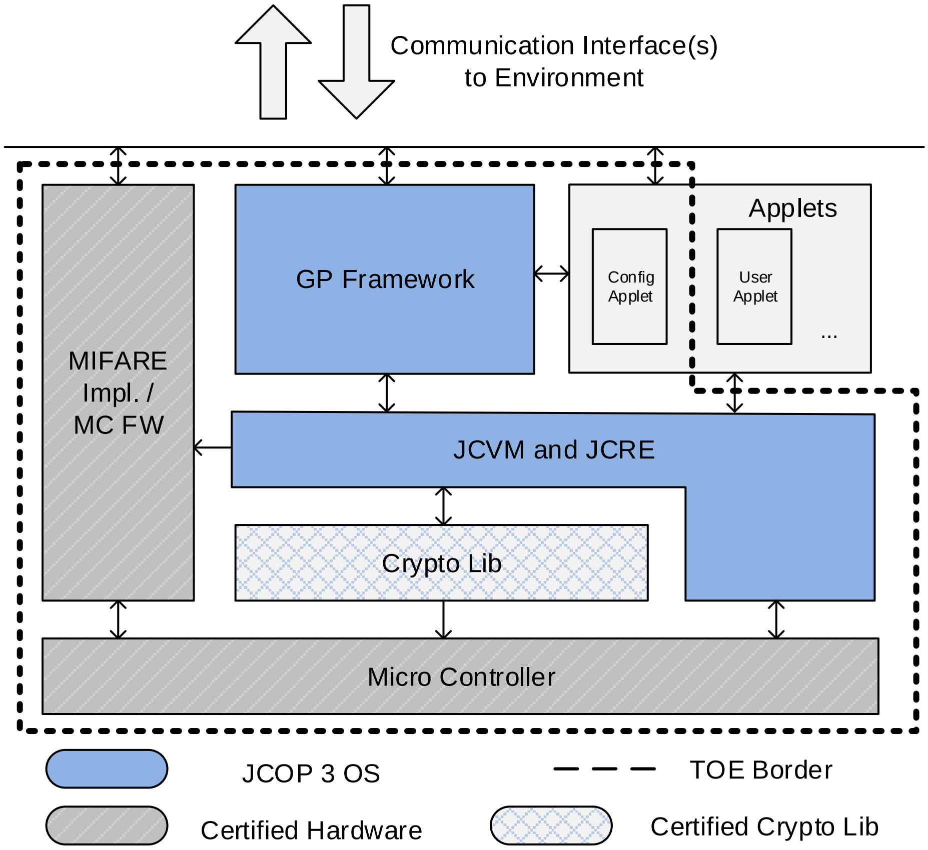

Fig. 1.1: Components of the TOE

Figure 1.1 also shows applets. The applets are small programs in Java language which can be executed by the TOE, but are not part of the TOE. The Config Applet has special privileges and is used to personalise the TOE. Customer applets are not part of the TOE. The Config Applet is part of the TOE.

1.2.1 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 many 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:

- 3 different communication protocols:
  1. ISO 7816 T=1
  2. ISO 7816 T=0
  3. ISO 14443 T=CL (contact-less)
- Cryptographic algorithms and functionality:
  1. 3DES for en-/decryption (CBC and ECB) and MAC generation and verification (Retail-MAC, CMAC and CBC-MAC).
  2. AES (Advanced Encryption Standard) for en-/decryption (CBC and ECB) and MAC generation and verification (CMAC, CBC-MAC).
  3. RSA and RSA CRT for en-/decryption and signature generation and verification.
  4. SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 hash algorithm.
  5. Random number generation according to class DRG.3 of AIS 20 [27]
- Java Card 3.0.4 functionality:
  1. Executing the Java byte codes which get generated from the Java compiler when Java source code is compiled.
  2. Managing memory allocation of code and data of applets.
  3. Enforcing access rules between applets and the JCRE.
  4. Mapping of Java method calls to native implementations of e.g. cryptographic operation.
  5. Garbage Collection fully implemented with complete memory reclamation incl. compactification.
  6. Support for Extended Length APDUs.
- GlobalPlatform 2.2.1 functionality:
  1. Loading of Java packages
  2. Instantiating applet instances
  3. Removing of Java packages
  4. Removing of applet instances
  5. Creating Supplementary Security Domains

6. Associating applets to Security Domains
7. Installation of keys
8. Verification of signatures of signed applets
9. Verification of signatures for commands
10. CVM Management (Global PIN) fully implemented.
11. Secure Channel Protocol (SCP01, SCP02 and SCP03) is supported.
12. ISD, SSD.
13. Delegated Management, DAP.
14. Post-issuance installation and deletion of applets, packages and objects.
15. Compliance to Mapping Guidelines and ID Configuration.

- NXP Proprietary Functionality

  1. MIFARE Implementation accessible via contactless interface and via Java Card API.
  2. Pre-Personalisation mechanism: JCOP3 includes a Config Applet that can be used for pre-personalisation and configuration of the TOE. At the end of Phase 6 ”Personalisation” the Config Applet is removed.

1.2.2 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.

1.2.3 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 [18] (T=1, T=0) or ISO14443 [7] (T=CL).
The second group of users are **administrators of cards**. They want to configure the card by using the Config Applet, 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 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. This means that for the certification of this TOE other certifications of components which are part of this TOE are re-used. In the following sections more detailed descriptions of the components of Figure 1.1 are provided. In the description it is also made clear whether a component is covered by a previous certification or whether it is covered in the certification of this TOE.

1.3.1.1 Micro Controller

The Micro Controller is a secure smart card controller from NXP from the SmartMX2 family. The Micro Controller contains a co-processor for symmetric cipher, supporting DES operations and AES, as well as well as an accelerator for asymmetric algorithms. It contains volatile (RAM) and non-volatile (ROM and EEPROM) 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 1.2:

Name	NXP Secure Smart Card Controller P6021J VB
Certification ID	BSI-DSZ-CC-0955
Reference	[10]

Tab. 1.2: Reference to Certified Micro Controller

1.3.1.2 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 non-volatile memory.

The MC FW has been certified in a previous certification. It has been certified together with the Micro Controller and the same references ([10]) as given for the Micro Controller also apply for the MC FW.

1.3.1.2.2 MIFARE Implementation

The NXP Secure Smart Card Controller P6021J VB hardware of this TOE contains the following MIFARE Implementation: MIFARE Plus MF1PLUSx0 and MIFARE DESFire EV1.

The MIFARE Implementation has been certified in a previous certification. It has been certified together with the Micro Controller and the same references ([10]) as given for the Micro Controller also apply for the MIFARE Implementation.

1.3.1.2.3 Crypto Lib

The Crypto Lib is certified in a previous certification and the results are re-used for this certification. The exact reference to the certification is given in the following Table:

Name	Crypto Library V3.1.x on P6021J VB
Minor Version	V3.1.1
Certification ID	NSCIB-CC-15-66030
Reference	[6]

Tab. 1.3: Reference to the Crypto Lib

1.3.1.3 IC Embedded Software

1.3.1.3.1 JCOP3 OS

JCOP3 OS consists of JCVM, JCRE, JCAPI and GP framework. It is implemented according to the Java Card Specification and GlobalPlatform version listed below. Additionally it consists of a proprietary API, which is described in the UGM [9].

JCVM, JCRE, and JCAPI version implemented in the TOE	Version 3.0.4 Classic [26] [25] [24]

Tab. 1.4: Java Card Specification Version

ID Configuration	Version 1.0 [21]
Mapping Guidelines	Version 1.0.1 [19]

Tab. 1.5: GlobalPlatform Version

The JCOP3 OS component can be identified by using the IDENTIFY APDU command (see UGM [9]). 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, EEPROM and loaded patches (if any). The Platform ID is a data string that allows to identify the JCOP3 OS component. Table 1.8 in section 1.3.3 lists all possible values for the Platform ID that are valid for this TOE.

The TOE also includes a Config Applet that is stored in the EEPROM and can be used for personalisation and configuration of the TOE. It must be deleted after the finalisation of the personalisation (end of Phase 6 ”Personalisation”) by using the DELETE APDU command. Once the Config Applet is deleted, it is no longer possible to access the configuration area in the EEPROM.

The Config Applet can be identified by using the SELECT APDU command that selects the Config Applet (see UGM [9]). The response to this APDU command contains two fields called Version Info that allow to identify the version of the Config Applet. Table 1.9 in section 1.3.3 lists all possible versions of the Config Applet that are in the scope of this TOE.

1.3.2 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 [8], see Figure 1.2.

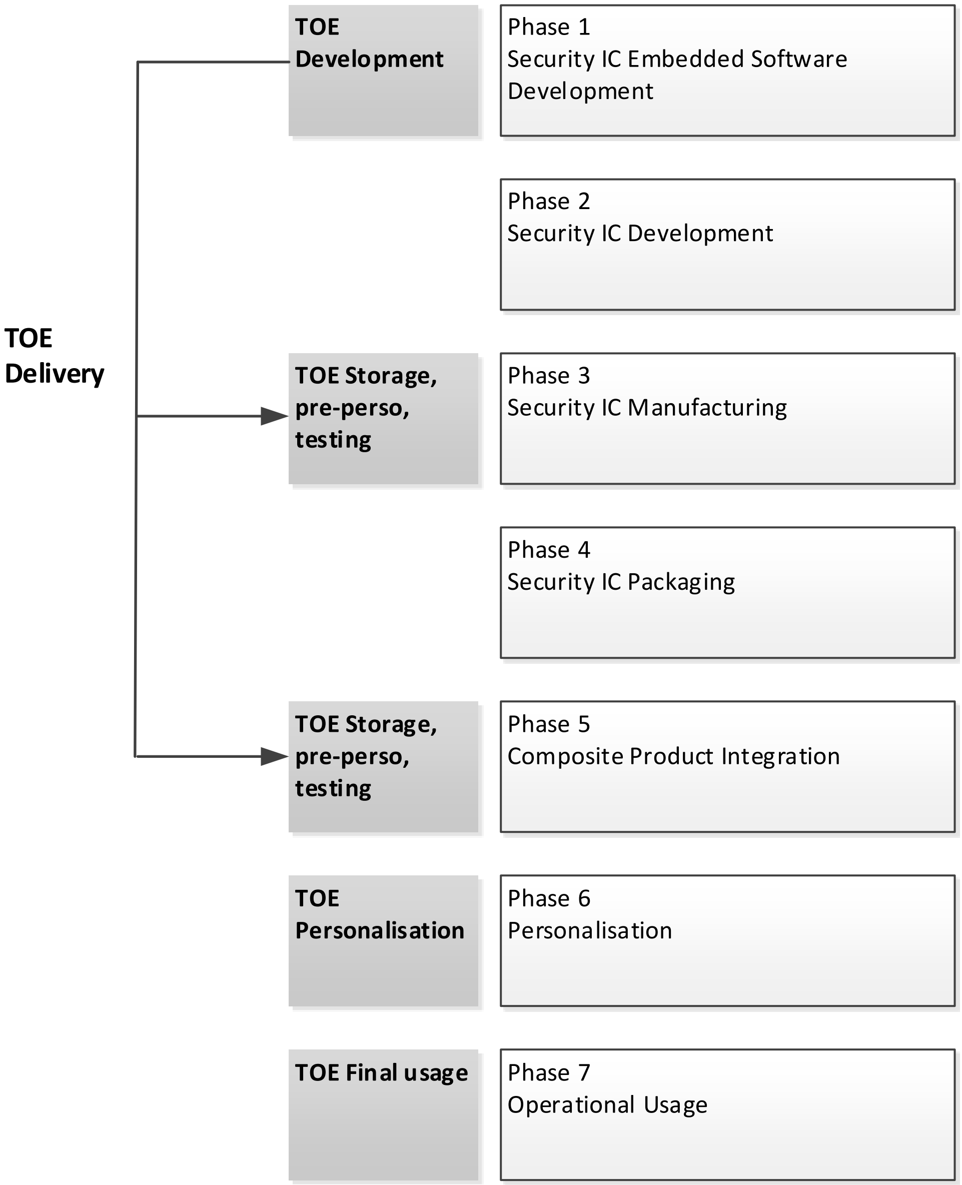

Fig. 1.2: TOE Life Cycle within Product Life Cycle

Phase	Name	Description
1	Security IC Embedded Software Development	The IC Embedded Software Developer is in charge of • smartcard embedded software development including the development of Java Card applets and • specification of IC pre-personalization requirements, though the actual data for IC pre-personalization come from phase 4, 5, or 6.
2	Security IC Development	The IC Developer • designs the IC, • develops IC Dedicated Software, • provides information, software or tools to the IC Embedded Software Developer, and • receives the embedded software from the developer, through trusted delivery and verification procedures. From the IC design, IC Dedicated Software and Smartcard Embedded Software, the IC Developer • constructs the smartcard IC database, necessary for the IC photomask fabrication.
3	Security IC Manufacturing	The IC Manufacturer is responsible for • producing the IC through three main steps: IC manufacturing, IC testing, and IC pre-personalization. The IC Mask Manufacturer • generates the masks for the IC manufacturing based upon an output from the smarcard IC database. Configuration items may be changed/deleted.
4	Security IC Packaging	The IC Packaging Manufacturer is responsible for • IC packaging and testing.

Phase	Name	Description
5	Composite Product Integration	The Composite Product Manufacturer is responsible for • smarcard product finishing process including applet loading and testing. Configuration items may be changed/deleted by using the Config Applet.
6	Personalization	The Personalizer is responsible for • smartcard (including applet) personalization and final tests. User Applets may be loaded onto the chip at the personalization process and configuration items may be changed/deleted by using the Config Applet. The Config Applet must be deleted at the end of this cycle by using the DELETE APDU command.
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.

Tab. 1.6: Life-cycle

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 ROM or EEPROM. Applet loading into ROM can only be done in phase 3. Applet loading into EEPROM can be done in phases 3, 4, 5, and 6. Applet loading in phase 7 is also allowed. This means post-issuance loading of applets can be done for a certified TOE. The Config Applet is loaded into EEPROM and has special privileges to personalize and configure the TOE. Before life cycle Phase 7 "Operational Use" the Config Applet is deleted and hence ensured that its functionality cannot be used afterwards. It is possible to load patch code into EEPROM 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 1.7. The delivery process from NXP to their customers (to phase 4 or phase 5 of the life cycle) guarantees, that the customer is aware of the exact versions of the different parts of the TOE as outlined above. TOE documentation is delivered in electronic form (encrypted according to defined mailing procedures).

Note: Phases 1 to 3 are under the TOE developer scope of control. Therefore, the objectives for the environment related to phase 1 to 3 are covered by Assurance measures, which are materialized by documents, process and procedures evaluated through the TOE evaluation process. During phases 4 to 7 the TOE is no more under the *developer control. In this environment, the TOE protects itself with its own Security functions. But some additional usage recommendation must also be followed in order to ensure that the TOE is correctly and securely handled, and that shall be 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).*

1.3.3 TOE Identification

The delivery comprises the following items:

Type	Name	Version	Date
Hardware	NXP Secure Smart Card Controller P6021J VB ROM Code (Platform ID) Patch Code (Patch ID)	see Table 1.8	
Document	User Guidance Manual [9]	Rev. 3.1	2015-12-04
Document	HW Objective Data Sheet [12]	Rev. 3.1	2016-01-15

Tab. 1.7: Delivery Items

The TOE can be identified by using the Platform ID, the Patch ID and the version of the Config Applet. Table 1.8 lists identifiers for all products that are in the scope of this TOE. The Platform ID and the Patch ID can be obtained by using the IDENTIFY APDU command (see UGM [9]).

Product Short Name	Platform ID	Patch ID
J3H016P J2H016P J3H041P J2H041P J3H081P	JxHyyy00E4D80300	0xE4D8000000000004

Tab. 1.8: Product Identification

Table 1.9 lists the versions of all Config Applets that are in the scope of this TOE. The version of the Config Applet can be obtained by using the SELECT APDU command, which is used to select the Config Applet. Version Info represents the content of the identically named field (2 bytes) of the SELECT APDU command response (see UGM [9]).

Config Applet Version	Version Info (Major)	Version Info (Minor)
v1.2	’01’	’02’

Tab. 1.9: Config Applet Identification

The commercial product names of JCOP products have the following form:

Jxyeeecapp(p)/mvrrff(o)

Additionally the products can also be referenced by its short name, which contains only a part of the commercial product name and has the following form:

Jxyeeec

The **"J"** is constant, the other letters are variables. For a detailed description of these variables, please see Table 1.11. For the certified products some variables need to have defined settings.
These settings are given in Table 1.10.

Variable	Must have one of these values
x	2, 3
y	H
eee	016, 041, 081 (016 and 041 represent the JxH016 and JxH041 platforms which are derivates of the JxH081 platform but with limited EEPROM sizes).
ca	c=P and a∈{0=No Applet in ROM;1-9;A-Z identifies Applets in ROM}
ppp	These letters indicate the package (alpha numeric, third character optional). All package types which are covered by the certification of the used hardware are allowed. For the list of certified packages please refer to the public security target of the corresponding hardware [10].
m	9
v	A
o	P, 4, 8, E, F for x=2: variable o is absent

Tab. 1.10: Product Commercial Names

The values for ‘rr‘, ‘ff‘ are customer dependent. The following table explains the naming conventions of the commercial product name of the JCOP products. Every JCOP product gets assigned such a commercial name, which includes also customer and application specific data. This table does not give any information about which commercial products are Common Criteria certified.

Variable	Meaning	Example Values	Parameter Settings
x	Interface/Crypto Support	2	Contact Interface Only / Symmetric & Asymmetric Crypto Support
		3	Dual Interface Support / Symmetric & Asymmetric Crypto Support
y	JCOP OS Version	H	JCOP3 P60-2
eee	Indication of non-volatile memory size and specification variations (alpha numeric)	16	16 KB EEPROM[1]
		41	40 KB EEPROM[1]
		81	80 KB EEPROM[1]
		82	80 KB EEPROM[1]
		145	144 KB EEPROM[1]
ca	Customization/System (Applet combinations in ROM)	c=G	General
		c=S	Sec ID
		c=P	Payment
		c=C	Convergence
		a∈{0-9,A-Z}	0=No applet in ROM; 1-9,A-Z defines pre-loaded applets in ROM
ppp	Package delivery type (alpha numeric, last character optional)	Ux(x)	Wafer not thinner than 50μm (The letter ”x(x)” in ”Ux(x)” stands for a capital letter or a number, which identifies the wafer type) e.g.: U15=150μm wafer sawn on FFC.
		Xn(n)	Dual Interface or Contact Module (The letter ”n(n)” in ”Xn(n)” stands for a capital letter or a number, which identifies the module type) e.g.: X30=multisource PDM1.1
		Ax	Contactless Module (The letter ”x” in ”Ax” stands for a capital letter or a number, which identifies the module type) e.g.: A4=MOB4 module
m	Manufacturer Identifier (alpha numeric)	9	
v	Silicon Version	A	
rr	Rom Code number (alpha numeric)	The values for ‘rr‘ and ‘ff‘ are customer dependent.	
ff	FabKey number (alpha numeric)		

Variable	Meaning	Example Values	Parameter Settings
o	Mifare Option	P	Disabled MIFARE DESFire and MIFARE Plus
		4	MIFARE Plus 4K & MIFARE DESFire 4K
		8	MIFARE Plus 4K & MIFARE DESFire 8K
		E[2]	Device with any other MIFARE Classic and MIFARE DESFire EV1 configuration (also if MIFARE DESFire is disabled)
		F[2]	Device with any other MIFARE Plus and MIFARE DESFire EV1 configuration (also if MIFARE Plus is disabled)

Tab. 1.11: Product Commercial Name Format

1.3.4 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 [10]), are also allowed to be used in combination with each product of this TOE. The commercial product names contain information about the used package type. The format of the commercial product names is defined in section 1.3.3.
For a detailed description of the package type names please refer to [13].
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.

[1] Actual available EEPROM depends on MIFARE and system memory.

[2] These configurations are only available on special request.

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 4, September 2012" [2]
- "Common Criteria for Information Technology Security Evaluation, Part 2, Version 3.1, Revision 4, September 2012" [3]
- "Common Criteria for Information Technology Security Evaluation, Part 3, Version 3.1, Revision 4, September 2012" [4]

The following methodology will be used for the evaluation:

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

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in Chapter 6.

2.2 Package Claim

This Security Target claims conformance to the assurance package EAL5 augmented. The augmentation to EAL5 is AVA_VAN.5 “Advanced methodical vulnerability analysis”, ALC_DVS.2 “Sufficiency of security measures”, 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 Protection Profile - Open Configuration, Version 3.0, Certified by ANSSI, the French Certification Body May, 2012 [8]. The Java Card Protection Profile makes the use of Java Card RMI optional. The TOE does not support Java Card RMI. This ST is more restrictive than the PP [8] which chapter 2.4 provides a rational for. The TOE implements the feature "Management of External Memory (EXT-MEM)" from the group EMG that the Java Card Protection Profile makes optional. The Security IC Security Target [10] also claims strict conformance to the Packages for Cryptographic Services including Package "TDES" and Package "AES", which are defined in the Security IC PP [11].

2.4 Conformance Claim Rationale

2.4.1 TOE Type

The TOE type as stated in Section 1.2 of this ST corresponds to the TOE type of the PP as stated in Section 1.2 of [8] namely a Java Card platform, implementing the Java Card Specification Version 3.0.4 [26, 25, 24].

2.4.2 SPD Statement

The SPD statement that is presented in Chapter 4 includes the threats as presented in the PP [8], but also includes additional threats. These threats are:

- T.OS_OPERATE
- T.RND
- T.CONFIG
- T.COM_EXPLOIT
- T.LIFE_CYCLE
- T.UNAUTHORIZED_CARD_MNGT
- T.INTEG-APPLI-DATA[REFINED]

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 [8].
The threat T.RND is taken from the Security IC PP [11].
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 [8].
The threat T.COM_EXPLOIT is included to cover communication channels attacks and it is an addition to the threats in the PP [8].
The threat T.LIFE_CYCLE is included to cover content management attacks and it is an addition to the threats in the PP [8].
The threat T.UNAUTHORIZED_CARD_MNGT refines the threats T.INSTALL and T.DELETION from the Security IC PP [11].
The threat T.INTEG-APPLI-DATA[REFINED] refines the threat T.INTEG-APPLI-DATA in the Security IC PP [11].
Note that the threat T.EXE-CODE-REMOTE is not included, since the TOE does not support Java Card RMI. The Java Card Protection Profile [8] makes the use of Java Card RMI optional.
The SPD statement presented in Chapter 4, copies the OSP from the PP [8], and adds three additional OSPs:

- OSP.PROCESS-TOE

- OSP.KEY-CHANGE
- OSP.SECURITY-DOMAINS

The OSP OSP.PROCESS-TOE is introduced for the pre-personalisation feature of the TOE and is an addition to the OSPs in PP [8]. This OSP is copied from the Security IC PP [11]. The OSP OSP.KEY-CHANGE is introduced for the SD feature of the TOE and is an addition to the OSPs in PP [8]. The OSP OSP.SECURITY-DOMAINS is introduced for the SD feature of the TOE and is an addition to the OSPs in PP [8].

The SPD statement includes two of the three assumptions from the PP [8]. 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 [8]. 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 [8], five 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 [10], which is compliant to the Security IC PP [11].

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

The assumptions A.APPS-PROVIDER and A.VERIFICATION-AUTHORITY are added because Security Domains from the GlobalPlatform Specification are introduced. All the applets and packages are signed by the APSD and the correctness is verified on the TOE by VASD before the package or applet is installed or loaded. A.APPS-PROVIDER and A.VERIFICATION-AUTHORITY are additions to PP [8] for card content management environment.

2.4.3 Security Objectives Statement

The statement of security objectives in the ST presented in Chapter 5 includes all security objectives as presented in the PP [8], but also includes a number of additional security objectives. These security objectives are:

- OT.IDENTIFICATION
- OT.RND

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

The security objectives OT.IDENTIFICATION, OT.RND are part of the security objectives of the certified Micro Controller [10](see also Section 1.3.1.1) and Crypto Lib [6](see also Section 1.3.1.2.3), which are also components of this composite certification. Therefore the security objective statement is equivalent to the PP [8] for these two security objectives. OT.IDENTIFICATION is also included for the pre-personalisation feature of the TOE, which is additional functionality the PP allows.

The security objective OT.CONFIG-LIMIT is related to the configuration of the TOE via the Config Applet, which is additional functionality the PP [8] allows.

The security objectives OT.DOMAIN-RIGHTS, OT.APPLI-AUTH, OT.COMM_AUTH, OT.COMM_INTEGRITY, OT.COMM_CONFIDENTIALITY are objectives for the TOE as the GlobalPlatform API and the definitions for Secure Channel, Security Domains and Card Content Management are used from it.

The ST contains OE.APPLET, OE.VERIFICATION and OE.CODE-EVIDENCE from Security Objectives for the Operational Environment from [8]. Additionally, some of the Security Objectives for the Operational Environment from [8] 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 [8] to which conformance is claimed.

The security objective O.EXT-MEM from the optional EMG group of the PP [8] is included.

The security objectives O.INSTALL, O.LOAD, and O.DELETION from the PP [8] 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 [8] has been included in this ST. The other part of security objectives for the environment, which is present in the PP [8], is used as part of the security objectives for the TOE in this ST. The ST also introduces eight additional security objectives for the environment. The additional objectives for the environment are:

- OE.USE_DIAG
- OE.USE_KEYS
- OE.PROCESS_SEC_IC
- OE.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 [10]see also Section 1.3.1.1) that is part of this composite product evaluation. Therefore the statement of security objectives for the environment is equivalent to the statement in the Security IC PP [11].
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 [8].
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 [8] to which conformance is claimed.

2.4.4 Security Functional Requirements Statement

The statement of security functional requirements copies most SFRs as defined in the PP [8], 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 [8]. Moreover as requested by the PP [8] the ST adds additional threats, objectives and SFRs to fully cover and describe additional security functionality implemented in the TOE.
The TOE restricts remote access from the CAD to the services implemented by the applets on the card to none, and as a result the SFRs concerning Java Card RMI (FDP_ACF.1[JCRMI], SFRs FDP_IFC.1/JCRMI, FDP_IFF.1/JCRMI, FMT_MSA.1/EXPORT, FMT_MSA.1/REM_REFS, FMT_MSA.3/JCRMI, FMT_SMF.1/JCRMI, FMT_REV.1/JCRMI, and FMT_SMR.1/JCRMI) are not included in the ST. In the PP [8] the use of the Java Card RMI is optional. The TOE does not implement Java Card RMI.

The SFR FDP_ITC.2/INSTALLER from the PP [8] is replaced by FDP_ITC.2[CCM] which enforces the Firewall 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 [8].
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 [8].
The table below lists the SFRs from CarG of PP [8] and their corresponding refinements in this ST.

SFR from PP [8]	Refinement
FCO_NRO.2/CM	FCO_NRO.2[SC]
FDP_IFC.2/CM	FDP_IFC.2[SC]
FDP_IFF.1/CM	FDP_IFF.1[SC]
FDP_UIT.1/CM	FDP_UIT.1[CCM]
FIA_UID.1/CM	FIA_UID.1[SC]
FMT_MSA.1/CM	FMT_MSA.1[SC]
FMT_MSA.3/CM	FMT_MSA.3[SC]
FMT_SMF.1/CM	FMT_SMF.1[SC]
FMT_SMR.1/CM	FMT_SMR.1[SD]
FTP_ITC.1/CM	FTP_ITC.1[SC]

Tab. 2.1: CarG SFRs refinements

The following SFRs realize refinements of SFRs from PP [8] and add functionality to the TOE making the statement of security requirements more restrictive than the PP [8]:
FDP_ROL.1[CCM] and FPT_FLS.1[CCM] realize additional security functionality for the card manager which is allowed by the PP [8].
The set of SFRs that define the security domains mechanism as specified by GlobalPlatform, realize refinements of SFRs from PP [8] (see above table 2.1) and additional security functionality which is allowed by the PP [8]. 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], and FMT_SMR.1[SD].
The set of SFRs that define the secure channel mechanism as specified by GlobalPlatform, realize refinements of SFRs from PP [8] (see above table 2.1) and additional security functionality which is allowed by the PP [8]. 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], and FTP_ITC.1[SC].

The SFRs FAU_SAS.1[SCP], FIA_AFL.1[PIN] and FCS_RNG.1 realize additional security functionality which is allowed by the PP [8].
The set of SFRs that define the Config Applet realize additional security functionality, which is allowed by the PP [8]. This set of SFRs comprise FDP_ACC.1[CONFIG-APPLET], FDP_ACF.1[CONFIG-APPLET], FMT_MSA.1[CONFIG-APPLET], FMT_MSA.3[CONFIG-APPLET], FMT_SMF.1[CONFIG-APPLET], FMT_SMR.1[CONFIG-APPLET], FIA_UID.1[CONFIG-APPLET] and FIA_UAU.1[CONFIG-APPLET].

3 Security Aspects

This chapter describes the main security issues of the Java Card System and its environment addressed in this ST, called "security aspects", in a CC-independent way. In addition to this, they also give a semi-formal framework to express the CC security environment and objectives of the TOE. They can be instantiated as assumptions, threats, objectives (for the TOE and the environment) or organizational security policies. The description is based on [8].

3.1 Confidentiality

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

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.

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-DATA

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 Executions

SA.EXE-APPLI-CODE

Execution of Application Code
Application (byte)code must be protected against unauthorized execution. This concerns:

1. invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([14])
2. jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code
3. unauthorized execution of a remote method from the CAD (if the TOE provides JCRMI functionality).

SA.EXE-JCS-CODE

Execution of Java Card System Code
Java Card System bytecode must be protected against unauthorized execution. Java Card System bytecode includes any code of the Java Card RE or API. This concerns:

1. invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([14])
2. jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code. Note that execute access to native code of the Java Card System and applications is the concern of SA.NATIVE.

SA.FIREWALL

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

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

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

SA.VERIFICATION

Bytecode Verification

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

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

3.5 Card Management

SA.CARD-MANAGEMENT

Card Management

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

SA.INSTALL

Installation

1. The TOE must be able to return to a safe and consistent state when the installation of a package or an applet fails or be cancelled (whatever the reasons).
2. Installing an applet must have no effect on the code and data of already installed applets. The installation procedure should not be used to bypass the TSFs. In short, it is an atomic operation, free of harmful effects on the state of the other applets.

3. The procedure of loading and installing a package shall ensure its integrity and authenticity.

SA.SID **Subject Identification**

1. Users and subjects of the TOE must be identified.
2. The identity of sensitive users and subjects associated with administrative and privileged roles must be particularly protected; this concerns the Java Card RE, the applets registered on the card, and especially the default applet and the currently selected applet (and all other active applets in Java Card System). A change of identity, especially standing for an administrative role (like an applet impersonating the Java Card RE), is a severe violation of the SFR. Selection controls the access to any data exchange between the TOE and the CAD and therefore, must be protected as well. The loading of a 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**

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

SA.DELETION **Deletion**

1. Deletion of installed applets (or 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.
2. Erasure, if deemed successful, shall ensure that any data owned by the deleted applet is no longer accessible (shared objects shall either prevent deletion or be made inaccessible). A deleted applet cannot be selected or receive APDU commands. Package deletion shall make the code of the package no longer available for execution.

3. Power failure or other failures during the process shall be taken into account in the implementation so as to preserve the SFRs. This does not mandate, however, the process to be atomic. For instance, an interrupted deletion may result in the loss of user data, as long as it does not violate the SFRs.

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

SA.ALARM

Alarm
The TOE shall provide appropriate feedback upon detection of a potential security violation. This particularly concerns the type errors detected by the bytecode verifier, the security exceptions thrown by the Java Card VM, or any other security-related event occurring during the execution of a TSF.

SA.OPERATE

Operate

1. The TOE must ensure continued correct operation of its security functions.
2. In case of failure during its operation, the TOE must also return to a well-defined valid state before the next service request.

SA.RESOURCES

Resources
The TOE controls the availability of resources for the applications and enforces quotas and limitations in order to prevent unauthorized denial of service or malfunction of the TSFs. This concerns both execution (dynamic memory allocation) and installation (static memory allocation) of applications and packages.

SA.CIPHER

Cipher
The TOE shall provide a means to the applications for ciphering sensitive data, for instance, through a programming interface to low-level, highly secure cryptographic services. In particular, those services must support cryptographic algorithms consistent with cryptographic usage policies and standards.

SA.KEY-MNGT

Key Management

The TOE shall provide a means to securely manage cryptographic keys. This includes:

1. Keys shall be generated in accordance with specified cryptographic key generation algorithms and specified cryptographic key sizes,
2. Keys must be distributed in accordance with specified cryptographic key distribution methods,
3. Keys must be initialized before being used,
4. Keys shall be destroyed in accordance with specified cryptographic key destruction methods.

SA.PIN-MNGT

PIN Management

The TOE shall provide a means to securely manage PIN objects. This includes:

1. Atomic update of PIN value and try counter,
2. No rollback on the PIN-checking function,
3. Keeping the PIN value (once initialized) secret (for instance, no clear-PIN-reading function),
4. Enhanced protection of PIN's security attributes (state, try counter ...) in confidentiality and integrity.

SA.SCP

Smart Card Platform

The smart card platform must be secure with respect to the SFRs. Then:

1. After a power loss, RF signal loss or sudden card removal prior to completion of some communication protocol, the SCP will allow the TOE on the next power up to either complete the interrupted operation or revert to a secure state.
2. It does not allow the SFRs to be bypassed or altered and does not allow access to other low-level functions than those made available by 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.
3. It provides secure low-level cryptographic processing to the Java Card System.
4. It supports the needs for any update to a single persistent object or class field to be atomic, and possibly a low-level transaction mechanism.
5. It allows the Java Card System to store data in a ”persistent technology memory” or in volatile memory, depending on its needs (for instance, transient objects must not be stored in non-volatile memory). The memory model is structured and allows for low-level control accesses (segmentation fault detection).

6. It safely transmits low-level exceptions to the TOE (arithmetic exceptions, checksum errors), when applicable.
7. Finally, it is required that the IC is designed in accordance with a well-defined set of policies and standards (for instance, those specified in [11]), 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 External Memory

SA.EXT-MNGT

External Memory

The Extended Memory feature is an API-based mechanism to access the external memory outside the addressable Java Card VM space.

3.8 Config Applet

SA.CONFIG-APPLET

Config Applet

The Config Applet is a JavaCard applet in EEPROM which allows to:

1. Read and modify entries in the configuration area of the TOE,
2. Write default keys for the ISD into the Config Applet,
3. Perform a Bulk Update to write a new EEPROm image,
4. Create the Card Manager,
5. Identify the TOE before the Card Manager is created.

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 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 [8].

4.1.1 User Data

D.APP_CODE	The code of the applets and libraries loaded on the card. To be protected from unauthorized modification.
D.APP_C_DATA	Confidential sensitive data of the applications, like the data contained in an object, 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, 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 modification.
D.APP_KEYS	Cryptographic keys owned by the applets. To be protected from unauthorized disclosure and modification.
D.APSD_KEYS	Refinement of D.APP_KEYS of [8]. Application Provider Security Domains cryptographic keys needed to establish secure channels with the AP. These keys can be used to load and install applications on the card if the Security Domain has the appropriate privileges. To be protected from unauthorized disclosure and modification.
D.ISD_KEYS	Refinement of D.APP_KEYS of [8]. Issuer Security Domain cryptographic keys needed to perform card management operations on the card. To be protected from unauthorized disclosure and modification.

D.VASD_KEYS	Refinement of D.APP_KEYS of [8]. Verification Authority Security Domain cryptographic keys needed to verify applications Mandated DAP signature. To be protected from unauthorized disclosure and modification.
D.GP_CODE	The code of the GlobalPlatform framework on the card. To be protected from unauthorized modification.
D.CARD_MNGT_DATA	The data of the card management environment, like for instance, the identifiers, the privileges, life cycle states, the memory resource quotas of applets and security domains. To be protected from unauthorized modification.
D.PIN	Any end-user's PIN. To be protected from unauthorized disclosure and modification.

Tab. 4.1: User Data Assets

4.1.2 TSF Data

D.API_DATA	Private data of the API, like the contents of its private fields. To be protected from unauthorized disclosure and modification.
D.CRYPTO	Cryptographic data used in runtime cryptographic computations, like a seed used to generate a key. To be protected from unauthorized disclosure and modification.
D.JCS_CODE	The code of the Java Card System. To be protected from unauthorized disclosure and modification.
D.JCS_DATA	The internal runtime data areas necessary for the execution of the JCVM, such as, for instance, the frame stack, the program counter, the class of an object, the length allocated for an array, any pointer used to chain data-structures. To be protected from unauthorized disclosure or modification.
D.SEC_DATA	The runtime security data of the JCRE, like, for instance, the AIDs used to identify the installed applets, the currently selected applet, the current context of execution and the owner of each object. To be protected from unauthorized disclosure and modification.
D.TSF_KEYs	Cryptographic keys owned by the TOE (e.g. keys for secure communication).

D.ADMIN_CONF_DATA	Private data stores in the configuration area of the TOE. Accessible only via the Config Applet using a NXP owned key. To be protected from unauthorized disclosure and modification.
D.PERSO_CONF_DATA	Private data stored in the configuration area of the TOE. Accessible only via the Config Applet using a customer owned key or a NXP owned key. To be protected from unauthorized disclosure and modification.

Tab. 4.2: TSF Data Assets

4.2 Threats

4.2.1 Confidentiality

T.CONFID-APPLI-DATA

Confidentiality of Application Data
The attacker executes an application to disclose data belonging to another application. See SA.CONFID-APPLI-DATA for details. Directly threatened asset(s): D.APP_C_DATA, D.PIN and D.APP_KEYs.

T.CONFID-JCS-CODE

Confidentiality of Java Card System Code
The attacker executes an application to disclose the Java Card System code. See SA.CONFID-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.

T.CONFID-JCS-DATA

Confidentiality of Java Card System Data
The attacker executes an application to disclose data belonging to the Java Card System. See SA.CONFID-JCS-DATA for details. Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.CRYPTO, D.JCS_DATA and D.TSF_KEYs.

4.2.2 Integrity

T.INTEG-APPLI-CODE

Integrity of Application Code
The attacker executes an application to alter (part of) its own code or another application's code. See SA.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

T.INTEG-APPLI-CODE.LOAD

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

T.INTEG-APPLI-DATA[REFINED]

Integrity of Application Data

The attacker executes an application to alter (part of) another application's data. See SA.INTEG-APPLI-DATA for details. Directly threatened asset(s): D.APP_I_DATA, D.PIN, D.APP_KEYs, D.ISD_KEYs, D.VASD_KEYs and S.APSD_KEYs.

This threat is a refinement of the Threat T.INTEG-APPLI-DATA from [8].

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.

T.INTEG-JCS-CODE

Integrity of of Java Card System Code

The attacker executes an application to alter (part of) the Java Card System code. See SA.INTEG-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.

T.INTEG-JCS-DATA

Integrity of 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.TSF_KEYs.

4.2.3 Identity Usurpation

T.SID.1

Subject Identification 1

An applet impersonates another application, or even the Java Card RE, in order to gain illegal access to some resources of the card or with respect to the end user or the terminal. See SA.SID for details. Directly threatened asset(s): D.SEC_DATA (other assets may be jeopardized should this attack succeed, for instance, if the identity of the JCRE is usurped), D.PIN and D.APP_KEYs.

T.SID.2

Subject Identification 2

The attacker modifies the TOE's attribution of a privileged role (e.g. default applet and currently selected applet), which allows illegal impersonation of this role. See SA.SID for further details. Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on whose identity was forged).

4.2.4 Unauthorized Execution

T.EXE-CODE.1

Code Execution 1

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

T.EXE-CODE.2

Code Execution 2

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

T.NATIVE

Native Code Execution

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

4.2.5 Denial of Service

T.RESOURCES

Consumption of Resources

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

4.2.6 Card Management

T.UNAUTHORIZED_CARD_MNGT

Unauthorized Card Management

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

- load of a 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 objective is a refinement of the Threats T.INSTALL and T.DELETION from [8].

T.COM_EXPLOIT

Communication Channel Remote Exploit

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

All assets are threatened.

T.LIFE_CYCLE

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

4.2.7 Services

T.OBJ-DELETION

Object Deletion
The attacker keeps a reference to a garbage collected object in order to force the TOE to execute an unavailable method, to make it to crash, or to gain access to a memory containing data that is now being used by another application. See SA.OBJ-DELETION for further details. Directly threatened asset(s): D.APP_C_DATA, D.APP_I_DATA and D.APP_KEYs.

4.2.8 Miscellaneous

T.PHYSICAL

Physical Tampering
The attacker discloses or modifies the design of the TOE, its sensitive data or application code by physical (opposed to logical) tampering means. This threat includes IC failure analysis, electrical probing, unexpected tearing, and DPA. That also includes the modification of the runtime execution of Java Card System or SCP software through alteration of the intended execution order of (set of) instructions through physical tampering techniques. This threatens all the identified assets. This threat refers to the point (7) of the security aspect SA.SCP, and all aspects related to confidentiality and integrity of code and data.

4.2.9 Operating System

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) bypass 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 Random Numbers

T.RND

Deficiency of Random Numbers

An attacker may predict or obtain information about random numbers generated by the TOE for instance because of a lack of entropy of the random numbers provided. An attacker may gather information about the produced random numbers which might be a problem because they may be used for instance to generate cryptographic keys. Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the TOE without specific knowledge about the TOE's generator. Malfunctions or premature ageing are also considered which may assist in getting information about random numbers.

4.2.11 Config Applet

T.CONFIG

Unauthorized configuration

The attacker tries to change configuration items, ISD default keys, EEPROM image or tries to create the Card Manager without authorization.

4.3 Organisational Security Policies

OSP.VERIFICATION

File Verification

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

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

OSP.PROCESS-TOE **Identification of the TOE**

An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this identification.

OSP.KEY-CHANGE **Security Domain Keys Change**

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

OSP.SECURITY-DOMAINS **Security Domains**

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

4.4 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.

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” ([26]) outside the API.

A.VERIFICATION **Bytecode Verification**

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

A.USE_DIAG **Usage of TOE’s Secure Communication Protocol by OE**

It is assumed that the operational environment supports and uses the secure communication protocols offered by the TOE.

A.USE_KEYS **Protected Storage of Keys Outside of TOE**

It is assumed that the keys which are stored outside the TOE and which are used for secure communication and authentication between Smart Card and terminals are protected for confidentiality and integrity in their own storage environment. This is especially true for D.APSD_KEYS, D.ISD_KEYS, and D.VASD_KEYS.

Info: This is to assume that the keys used in terminals or systems are correctly protected for confidentiality and integrity in their own environment, as the disclosure of such information which is shared with the TOE but is not under the TOE control, may compromise the security of the TOE.

A.PROCESS-SEC-IC

Protection during Packaging, Finishing and Personalisation

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that the Phases after TOE Delivery are assumed to be protected appropriately. The assets to be protected are: The information and material produced and/or processed by the Security IC Embedded Software Developer in Phase 1 and by the Composite Product Manufacturer can be grouped as follows:

1. the Security IC Embedded Software including specifications, implementation and related documentation,
2. pre-personalisation and personalisation data including specifications of formats and memory areas, test related data,
3. the User Data and related documentation, and
4. material for software development support

as long as they are not under the control of the TOE Manufacturer.

A.APPS-PROVIDER

Application Provider

The AP is a trusted actor that provides basic or secure applications. He is responsible for his security domain keys (APSD keys).

Info: An AP generally refers to the entity that issues the application. For instance it can be a financial institution for a payment application such as EMV or a transport operator for a transport application.

A.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

OT.SID

Subject Identification

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

5.1.2 Execution

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.

OT.GLOBAL_ARRAYS_CONFID

Confidentiality of Global Arrays

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

OT.GLOBAL_ARRAYS_INTEG

Integrity of Global Arrays

The TOE shall ensure that only the currently selected applications may have a write access to the APDU buffer and the global byte array used for the invocation of the install method of the selected applet.

OT.NATIVE

Native Code

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

OT.OPERATE

Correct Operation

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

OT.REALLOCATION

Secure Re-Allocation

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

OT.RESOURCES

Resources availability

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

5.1.3 Services

OT.ALARM

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

OT.CIPHER

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

OT.KEY-MNGT

Key Management
The TOE shall provide a means to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys. See SA.KEY-MNGT.

OT.PIN-MNGT

Pin Management
The TOE shall provide a means to securely manage PIN objects. See SA.PIN-MNGT for details.

AppNote: PIN objects may play key roles in the security architecture of client applications. The way they are stored and managed in the memory of the smart card must be carefully considered, and this applies to the whole object rather than the sole value of the PIN.

OT.TRANSACTION

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

5.1.4 Object Deletion

OT.OBJ-DELETION

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

5.1.5 Applet Management

OT.APPLI-AUTH

Application Authentication
The card manager shall enforce the application security policies established by the card issuer by requiring application authentication during application loading on the card. This security objective is a refinement of the Security Objective O.LOAD from [8].

AppNote: Each application loaded onto the TOE has been signed by a VA. The VA will guarantee that the security policies established by the card issuer on applications are enforced. For example this authority (DAP) or a third party (Mandated DAP) can be present on the TOE as a Security Domain whose role is to verify each signature at application loading.

OT.DOMAIN-RIGHTS

Domain Rights
The Card issuer shall not get access or change personalized AP security domain keys which belong to the AP. Modification of a security domain keyset is restricted to the AP who owns the security domain.
AppNote: APs have a set of keys that allows them to establish a secure channel between them and the platform. These keys sets are not known by the TOE issuer. The security domain initial keys are changed before any operation on the SD (OE.KEY-CHANGE).

OT.COMM_AUTH

Communication Mutual Authentication
The TOE shall authenticate the origin of the card management requests that the card receives, and authenticate itself to the remote actor.

OT.COMM_INTEGRITY

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

OT.COMM_CONFIDENTIALITY

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

5.1.6 External Memory

OT.EXT-MEM

External Memory
The TOE shall provide controlled access means to the external memory and ensure that the external memory does not address Java Card System memory (containing User Data and TSF Data).

5.1.7 Card Management

OT.CARD-MANAGEMENT

Card Management
The TOE shall provide card management functionalities (loading, installation, extradition, deletion of applications and GP registry updates) in charge of the life cycle of the whole device and installed applications (applets). The card manager, the application with specific rights responsible for the administration of the smart card, shall control the access to card management functions. It shall also implement the card issuer's policy on card management.

The Security Objective from [8] 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 [8]. 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 safe.
- 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 [8] 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 [8]. 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.

5.1.8 Smart Card Platform

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 [8] for the environment OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are listed as TOE Security Objectives (OT.SCP.RECOVERY, OT.SCP.SUPPORT, and OT.SCP.IC) for the TOE in this section as the Smart Card Platform belongs to the TOE for this evaluation.

OT.SCP.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 [8] for the environment OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are listed as TOE Security Objectives (OT.SCP.RECOVERY, OT.SCP.SUPPORT, and OT.SCP.IC) for the TOE in this section as the Smart Card Platform belongs to the TOE for this evaluation.

OT.SCP.SUPPORT

SCP Support

The SCP shall support the TSFs of the TOE. This security objective for the environment refers to the security aspects 2, 3, 4 and 5 of SA.SCP

AppNote: The Security Objectives from [8] for the environment OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC are listed as TOE Security Objectives (OT.SCP.RECOVERY, OT.SCP.SUPPORT, and OT.SCP.IC) for the TOE in this section as the Smart Card Platform belongs to the TOE for this evaluation.

OT.IDENTIFICATION

TOE identification

The TOE must provide means to store Initialization Data and Pre-personalization Data in its non-volatile memory. The Initialization Data (or parts of them) are used for TOE identification.

5.1.9 Random Numbers

OT.RND

Quality of random numbers

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

5.1.10 Config Applet

OT.CONFIG-LIMIT

Limitation of the Configuration

The TOE shall not permit for unauthorized users to read and modify configuration items, ISD default keys, the EEPROM image or create the Card Manager.

5.2 Security Objectives for the Operational Environment

OE.APPLET

Applet

No applet loaded post-issuance shall contain native methods.

OE.VERIFICATION

Bytecode Verification

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

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

Application Note:

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

OE.CODE-EVIDENCE

Code Evidence

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

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

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

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

OE.APPS-PROVIDER

Application Provider

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

OE.VERIFICATION-AUTHORITY

Verification Authority

The VA should be a trusted actor who is able to guarantee and check the digital signature attached to an application.

OE.KEY-CHANGE

Security Domain Key Change

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

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

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

OE.USE_KEYS **Protection of OPE keys**
During the TOE usage, the terminal or system in interaction with the TOE, shall ensure the protection (integrity and confidentiality) of their own keys by operational means and/or procedures.

OE.PROCESS_SEC_IC **Protection during composite product manufacturing**
Security procedures shall be used after TOE Delivery up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that Phases after TOE Delivery up to the end of Phase 6 must be protected appropriately.

5.3 Security Objectives Rationale

In this section it is proven that the security objectives described in Chapter 4 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.EXT-MEM OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT

Security Problem Definition	Security Objective
T.CONFID-JCS-CODE	OT.NATIVE OE.VERIFICATION OT.EXT-MEM OT.CARD-MANAGEMENT
T.CONFID-JCS-DATA	OT.SID OT.FIREWALL OT.OPERATE OT.ALARM OE.VERIFICATION OT.EXT-MEM OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT
T.INTEG-APPLI-CODE	OT.NATIVE OE.VERIFICATION OT.EXT-MEM OT.CARD-MANAGEMENT OE.CODE-EVIDENCE
T.INTEG-APPLI-CODE.LOAD	OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.APPLI-AUTH
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
T.INTEG-APPLI-DATA.LOAD	OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.APPLI-AUTH

Security Problem Definition	Security Objective
T.INTEG-JCS-CODE	OT.NATIVE OE.VERIFICATION OT.EXT-MEM OT.CARD-MANAGEMENT OE.CODE-EVIDENCE
T.INTEG-JCS-DATA	OT.SID OT.FIREWALL OT.OPERATE OT.ALARM OE.VERIFICATION OT.EXT-MEM OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT OE.CODE-EVIDENCE
T.SID.1	OT.SID OT.FIREWALL OT.GLOBAL_ARRAYS_CONFID OT.GLOBAL_ARRAYS_INTEG OT.CARD-MANAGEMENT
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.RESOURCES	OT.OPERATE OT.RESOURCES OT.CARD-MANAGEMENT OT.SCP.RECOVERY OT.SCP.SUPPORT
T.UNAUTHORIZED_CARD_MNGT	OT.CARD-MANAGEMENT OT.DOMAIN-RIGHTS OT.COMM_AUTH

Security Problem Definition	Security Objective
	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.CONFIG-LIMIT
T.PHYSICAL	OT.SCP.IC
T.OS_OPERATE	OT.OPERATE
T.RND	OT.RND
OSP.VERIFICATION	OE.VERIFICATION OT.CARD-MANAGEMENT OE.CODE-EVIDENCE OT.APPLI-AUTH
OSP.PROCESS-TOE	OT.IDENTIFICATION
OSP.KEY-CHANGE	OE.KEY-CHANGE
OSP.SECURITY-DOMAINS	OE.SECURITY-DOMAINS
A.APPLET	OE.APPLET
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

Tab. 5.1: SPDs of the TOE vs. Objectives

5.3.1 Threats

5.3.1.1 Confidentiality

T.CONFID-APPLI-DATA

Objective	Rationale
OT.SID	Counters this threat by providing correct identification of applets.
OT.FIREWALL	Counters this thread by providing the Java Card Virtual Machine Firewall as specified in [25].

Objective	Rationale
OT.GLOBAL_ARRAYS_CONFID	Counters this threat by preventing the disclosure 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 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.
OT.EXT-MEM	Contributes to counter this threat by controlling the access to external memory areas.
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.

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.EXT-MEM	Contributes to counter this threat by controlling the access to external memory areas.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.

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.EXT-MEM	Contributes to counter this threat by controlling the access to external memory areas.
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.

5.3.1.2 Integrity

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.

Objective	Rationale
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.EXT-MEM	Contributes to counter this threat by controlling the access to external memory areas.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OE.CODE-EVIDENCE	The objective OE.CODE-EVIDENCE contributes to counter this threat by ensuring that integrity and authenticity evidences exist for the application code loaded into the platform.

T.INTEG-APPLI-CODE.LOAD

Objective	Rationale
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions such as the installation, update or deletion of applets.
OE.CODE-EVIDENCE	Contributes to counter this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.
OT.APPLI-AUTH	Counters this threat by ensuring that the loading of packages is done securely and thus preserves the integrity of packages code.

T.INTEG-APPLI-DATA[REFINED]

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

Objective	Rationale
OT.REALLOCATION	Counters the threat by preventing any attempt to read a piece of information that was previously used by an application but has been logically deleted. It states that any information that was formerly stored in a memory block shall be cleared before the block is reused.
OT.ALARM	Contributes to counter this threat by obtaining clear warning and error messages from the firewall, which is a software tool automating critical controls, so that the appropriate countermeasure can be taken.
OT.CIPHER	Contributes to counter this threat by protecting the data shared or information communicated between applets and the CAD using cryptographic functions.
OT.KEY-MNGT	Counters this threat by providing appropriate management of keys, PINs which are particular cases of an application's sensitive data.
OT.PIN-MNGT	Counters this threat by providing appropriate management of keys, PINs which are particular cases of an application's sensitive data.
OT.TRANSACTION	Counters this threat by providing appropriate management of keys, PINs which are particular cases of an application's sensitive data.
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.

T.INTEG-APPLI-DATA.LOAD

Objective	Rationale
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions such as the installation, update or deletion of applets.
OE.CODE-EVIDENCE	Contributes to counter this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.
OT.APPLI-AUTH	Counters this threat by ensuring that the loading of packages is done securely and thus preserves the integrity of packages code.

T.INTEG-JCS-CODE

Objective	Rationale
OT.NATIVE	Counters this threat by ensuring that no native code can be run to modify a piece of code.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecode. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code.
OT.EXT-MEM	Contributes to counter this threat by controlling the access to external memory areas.
OT.CARD-MANAGEMENT	Contributes to counter this threat by controlling the access to card management functions.
OE.CODE-EVIDENCE	Contributes to counter this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.

T.INTEG-JCS-DATA

Objective	Rationale
OT.SID	Counters this threat by providing correct identification of applets.
OT.FIREWALL	Contributes to counter this threat by providing means of separation.
OT.OPERATE	Counters the threat by ensuring that the firewall shall never stop operating.
OT.ALARM	Contributes to counter this threat by obtaining clear warning and error messages from the firewall so that the appropriate countermeasure can be taken.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes.

Objective	Rationale
OT.EXT-MEM	Contributes to counter this threat by controlling the access to external memory areas.
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.

5.3.1.3 Identity Usurpation

T.SID.1

Objective	Rationale
OT.SID	Counters this threat by providing unique subject identification.
OT.FIREWALL	Counters the threat by providing separation of application data (like PINs).
OT.GLOBAL_ARRAYS_CONFID	Counters this threat by preventing the disclosure of the installation parameters of an applet (like its name). These parameters are loaded into a global array that is also shared by all the applications. The disclosure of those parameters could be used to impersonate the applet.
OT.GLOBAL_ARRAYS_INTEG	Counters this threat by preventing the disclosure of the installation parameters of an applet (like its name). These parameters are loaded into a global array that is also shared by all the applications. The disclosure of those parameters could be used to impersonate the applet.
OT.CARD-MANAGEMENT	Contributes to counter this threat by preventing usurpation of identity resulting from a malicious installation of an applet on the card.

T.SID.2

Objective	Rationale
OT.SID	Counters this threat by providing unique subject identification.

Objective	Rationale
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

T.EXE-CODE.1

Objective	Rationale
OT.FIREWALL	Counters the threat by preventing the execution of non-shareable methods of a class instance by any subject apart from the class instance owner.
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code.

T.EXE-CODE.2

Objective	Rationale
OE.VERIFICATION	Contributes to counter the threat by checking the bytecodes. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. Especially the control flow confinement and the validity of the method references used in the bytecodes are guaranteed.

T.NATIVE

Objective	Rationale
OT.NATIVE	Counters this threat by ensuring that a Java Card applet can only access native methods indirectly that is, through an API.
OE.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.

5.3.1.5 Denial of Service

T.RESOURCES

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

5.3.1.6 Card Management

T.UNAUTHORIZED_CARD_MNGT

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

T.COM_EXPLOIT

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

T.LIFE_CYCLE

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

5.3.1.7 Services

T.OBJ-DELETION

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

5.3.1.8 Miscellaneous

T.PHYSICAL

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

5.3.1.9 Operating System

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.

5.3.1.10 Random Numbers

T.RND

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

5.3.1.11 Config Applet

T.CONFIG

Objective	Rationale
OT.CONFIG-LIMIT	Counters the threat by allowing only authorized users to read and modify configuration items, ISD default keys, the EEPROM image or create the Card Manager.

5.3.2 Organisational Security Policies

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.

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

OSP.PROCESS-TOE

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

OSP.KEY-CHANGE

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

OSP.SECURITY-DOMAINS

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

5.3.3 Assumptions

A.APPLET

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

A.VERIFICATION

Objective	Rationale
OE.VERIFICATION	Upholds the assumption by guaranteeing that all the bytecodes shall be verified at least once, before the loading, before the installation or before the execution in order to ensure that each bytecode is valid at execution time.
OE.CODE-EVIDENCE	This assumption is also upheld by the security objective of the environment OE.CODE-EVIDENCE which ensures that evidences exist that the application code has been verified and not changed after verification.

A.USE_DIAG

Objective	Rationale
OE.USE_DIAG	Directly upholds this assumption.

A.USE_KEYS

Objective	Rationale
OE.USE_KEYS	Directly upholds this assumption.

A.PROCESS-SEC-IC

Objective	Rationale
OE.PROCESS_SEC_IC	Directly upholds this assumption.

A.APPS-PROVIDER

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

A.VERIFICATION-AUTHORITY

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

6 Extended Components Definition (ASE_ECD)

6.1 Definition of Family ”Generation of random numbers (FCS_RNG)”

This section has been taken over from the certified (BSI-PP-0084) Smartcard IC Platform Protection profile [11].

6.1.1 Family behavior

This family defines quality requirements for the generation of random numbers which are intended to be use for cryptographic purposes.
Component leveling:

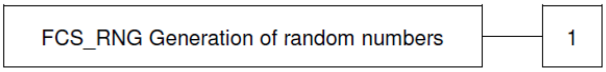

Fig. 6.1: Random Number Generation

FCS_RNG Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RNG.1
There are no management activities foreseen.

Audit: FCS_RNG.1
There are no actions defined to be auditable.

FCS_RNG.1 Random Number Generation.

Hierarchical to: No other components.

Dependencies No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic] random number generator that implements: [assignment: list of security capabilities].

FCS_RNG.1.2 The TSF shall provide [selection: bits, octets of bits, numbers [assignment: format of the numbers]] that meet [assignment: a defined quality metric].

Application Note: A physical random number generator (RNG) produces the random number by a noise source based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes like human interaction (key strokes, mouse movement). A deterministic RNG uses an random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs where a hybrid physical RNG produces at least the amount of entropy the RNG output may contain and the internal state of a hybrid deterministic RNG output contains fresh entropy but less than the output of RNG may contain.

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

This section has been taken over from the certified (BSI-PP-0084-2014) Smartcard IC Platform Protection profile [11]. To define the security functional requirements of the TOE an additional family (”Audit Data Storage (FAU_SAS)”) of the Class ”Security audit (FAU)” is defined here. This family describes the functional requirements for the storage of audit data. It has a more general approach than FAU_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

6.2.1 Family behavior

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

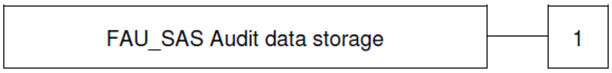

Fig. 6.2: SAS Component

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

Management: FAU_SAS.1
There are no management activities foreseen.

Audit: FAU_SAS.1
There are no actions defined to be auditable.

FAU_SAS.1 Audit storage.

Hierarchical to: No other components.

Dependencies No dependencies.

FAU_SAS.1.1 The TSF shall provide [assignment: list of subjects] with the capability to store [assignment: list of audit information] in the [assignment: type of persistent memory].

7 Security Requirements (ASE_REQ)

This section states the security functional requirements for the TOE. For readability requirements are arranged into groups taken from [8]. The permitted operations (assignment, iteration, selection and refinement) of the SFRs taken from Common Criteria [3] are printed in bold. Completed operations related to the PP are additionally marked within [ ] where assignments are additionally marked with the keyword "assignment".

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 [16] (cf. Java Card System Protection Profile Collection [17]).
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.
External Memory (EMG)	The EMG group contains security requirements for the management of external memory.

Tab. 7.1: Requirement Groups

Subjects are active components of the TOE that (essentially) act on the behalf of users. The users of the TOE include people or institutions (like the applet developer, the card issuer, the verification authority), hardware (like the CAD where the card is inserted or the PCD) and software components (like the application packages installed on the card). Some of the users may just be aliases for other users. For instance, the verification authority in charge of the bytecode verification of the applications may be just an alias for the card issuer. Subjects (prefixed with an "S") are described in the following table:

Subject	Description
S.ADEL	The applet deletion manager which also acts on behalf of the card issuer. It may be an applet ([25], §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 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 card are executed.
S.JCVM	The bytecode interpreter that enforces the firewall at runtime.
S.LOCAL	Operand stack of a JCVM frame, or local variable of a JCVM frame containing an object or an array of references.
S.SD	A GlobalPlatform Security Domain representing on the card a off-card entity. This entity can be the Issuer, an Application Provider, the Controlling Authority or the Verification Authority.
S.MEMBER	Any object's field, static field or array position.
S.PACKAGE	A package is a namespace within the Java programming language that may contain classes and interfaces, and in the context of Java Card technology, it defines either a user library, or one or several applets.
S.CONFIG-APPLET	The applet which is used to read and modify items in the configuration area of the TOE, write default ISD keys, write a new EEPROM image and created the Card Manager. The Config Applet further allows to identify the TOE before the Card Manager is created.

Tab. 7.2: Subject Descriptions

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

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.CONFIG-ITEM	Information which is accessed by S.CONFIG-APPLET. It can be one of the following: 1. a Config Item in the configuration area of the TOE, consisting of a tag, length and the value, 2. keys used as default keys for the SCP protocol of the ISD, 3. a whole EEPROM image.
O.CONFIG-APPLET	The applet, which is used for pre-personalisation.
O.EXT_MEM_INSTANCE	Any External Memory Instance created from the MemoryAccess Interface of the **external** package from the Java Card API [24].

Tab. 7.3: Object Groups

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

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

Tab. 7.4: Information Groups

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

Security attributes	Description
Active Applets	The set of the active applets' AIDs. An active applet is an applet that is selected on at least one of the logical channels.
Applet Selection Status	”Selected” or ”Deselected”.
Applet's Version Number	The version number of an applet (package) indicated in the export file.
Context	Package AID or ”Java Card RE ”.
Currently Active Context	Package AID or ”Java Card RE”.

Security attributes	Description
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.
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.
Key Set	Key Set for Secure Channel
Security Level	Secure Communication Security Level defined in Section 10.6 of [20]
Security Channel Protocol	Security Channel Protocol version used
Session Key	Security Channel's session key
Sequence Counter	Security Channel Session's Sequence Counter
ICV	Security Channel Session's ICV
Card Life Cycle	defined in Section 5.1.1 of [20]
Privileges	defined in Section 6.6.1 of [20]
Life-cycle Status	defined in Section 5.3.2 of [20]
Config Item Tag	The tag which uniquely identifies a Config Item.
Config Item Content	The data content of a Config Item.
ISD Default Keys	Initial Keys used for the ISD SCP protocol.
EEPROM Image	Content of the EEPROM of the TOE. Contains the Configuration Area, the Config Applet and its keys for authentication and token verification, and the default keys for the ISD.

Tab. 7.5: Security attribute description

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

[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.

seen as security attributes that are under the control of the subject performing the operation.

Operations	Description
OP.ARRAY_ACCESS(O.JAVAOBJECT, field)	Read/Write an array component.
OP.CREATE(Sharing, LifeTime)(*) [2]	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 [25], §6.2.8. It stands for one of the operations OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW, OP.TYPE_ACCESS.
OP.PUT(S1,S2,I)	Transfer a piece of information I from S1 to S2.
OP.THROW(O.JAVAOBJECT)	Throwing of an object (athrow, see [25], §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.CREATE_EXT_MEM_INSTANCE	Creation of an instance supporting the MemoryAccess Interface.
OP.READ_EXT_MEM(O.EXT_MEM_IN STANCE, address)	Reading the external memory represented by O.EXT_MEM_INSTANCE.

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

Operations	Description
OP.WRITE_EXT_MEM(O.EXT_MEM_IN STANCE, address)	Writing the external memory represented by O.EXT_MEM_INSTANCE.
OP.READ_CONFIG_ITEM	Reading a Config Item from the configuration area.
OP.MODIFY_CONFIG_ITEM	Writing of a Config Item, an ISD default key or the EEPROM image.
OP.USE_CONFIG_ITEM	Operational usage of Config Items or ISD default keys by subjects inside the TOE.
OP.IDENTIFY_TOE	Identification of the TOE by using the according information stored in O.CONFIG-ITEM.
OP.CREATE_CARD_MANAGER	Creation of the Card Manager.

Tab. 7.6: Operation Description

7.1 Security Functional Requirements

7.1.1 COREG_LC Security Functional Requirements

The list of SFRs of this category are taken from [8].

7.1.1.1 Firewall Policy

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

Hierarchical-To FDP_ACC.1 Subset access control

Dependencies FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1[FIREWALL] The TSF shall enforce the **[assignment: FIREWALL access control SFP]** on **[assignment: S.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)

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.

AppNote

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

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

Hierarchical-To

No other components.

Dependencies

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

FDP_ACF.1.1[FIREWALL]

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

Subject/Object	Security attributes
S.PACKAGE	**LC Selection Status**
S.JCVM	**Active Applets, Currently Active Context**
S.JCRE	**Selected Applet Context**
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 ([25], §6.2.8): S.PACKAGE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...), OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...), OP.THROW(O.JAVAOBJECT) or OP.TYPE_ACCESS(O.JAVAOBJECT, class) upon any O.JAVAOBJECT whose Sharing attribute has value "JCRE entry point" or "global array".
- R.JAVA.2 ([25], §6.2.8): S.PACKAGE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...), OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...) or 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 ([25], §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 ([25], §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:

a) The value of the attribute LC Selection Status of the package whose AID is ”Package AID” is ”Multiselectable”,

b) 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)(*) only if the value of the Sharing parameter is ”Standard”.

]

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, ...) or OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...)).

]

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

- Any subject with OP.JAVA(...) upon an O.JAVAOBJECT whose LifeTime attribute has value ”CLEAR_ON_DESELECT” if O.JAVAOBJECT’s Context attribute is not the same as the Selected Applet Context.
- Any subject attempting to create an object by the means of OP.CREATE(Sharing, LifeTime)(*) and a ”CLEAR_ON_DESELECT” LifeTime parameter if the active context is not the same as the Selected Applet Context.

]

AppNote FDP_ACF.1.4[FIREWALL]:

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

In the case of an array type, fields are components of the array ([15], §2.14, §2.7.7), as well as the length; the only methods of an array object are those inherited from the Object class.

The Sharing attribute defines four categories of objects:

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

When a new object is created, it is associated with the Currently Active Context. But the object is owned by the applet instance within the Currently Active Context when the object is instantiated ([25], §6.1.3). An object is owned by an applet instance, by the JCRE or by the package library where it has been defined (these latter objects can only be arrays that initialize static fields of packages).

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

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

([25], §6.1.2.1) At any point in time, there is only one active context within the Java Card VM (this is called the Currently Active Context).

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

It should be noticed that the Java Card platform, version 2.2.x and version 3 Classic Edition, introduces the possibility for an applet instance to be selected on multiple logical channels at the same time, or accepting other applets belonging to the same package being selected simultaneously. These applets are referred to as multiselectable applets. Applets that belong to a same package are either all multiselectable or not ([26], §2.2.5). Therefore, the selection mode can be regarded as an attribute of packages. No selection mode is defined for a library package.

An applet instance will be considered an active applet instance if it is currently selected in at least one logical channel. An applet instance is the currently selected applet instance only if it is processing the current command. There can only be one currently selected applet instance at a given time. ([25], §4).

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

Hierarchical-To No other components.

Dependencies FDP_IFF.1 Simple security attributes

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

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

FDP_IFF.1[JCVM] **Simple security attributes (JCVM)**

Hierarchical-To No other components.

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

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

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]**.

AppNote The storage of temporary Java Card RE-owned objects references is runtime-enforced ([25], §6.2.8.1-3).

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

FDP_RIP.1[OBJECTS] **Subset residual information protection (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]**.

AppNote The semantics of the Java programming language requires for any object field and array position to be initialized with default values when the resource is allocated [15], §2.5.1.

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

Hierarchical-To No other components.

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

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

AppNote The modification of the Selected Applet Context should be performed in accordance with the rules given in [25], §4 and [26], §3.4.

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

Hierarchical-To No other components.

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

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

AppNote The modification of the Currently Active Context should be performed in accordance with the rules given in [25], §4 and [26], §3.4.

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

Hierarchical-To No other components.

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

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

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

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

Hierarchical-To No other components.

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

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

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

AppNote FMT_MSA.3.1[FIREWALL]

- Objects' security attributes of the access control policy are created and initialized at the creation of the object or the subject. Afterwards, these attributes are no longer mutable (FMT_MSA.1[JCRE]). At the creation of an object (OP.CREATE), the newly created object, assuming that the FIREWALL access control SFP permits the operation, gets its Lifetime and Sharing attributes from the parameters of the operation; on the contrary, its Context attribute has a default value, which is its creator's Context attribute and AID respectively ([25], §6.1.3). There is one default value for the Selected Applet Context that is the default applet identifier's Context, and one default value for the Currently Active Context that is ”Java Card RE”.
- The knowledge of which reference corresponds to a temporary entry point object or a global array and which does not is solely available to the Java Card RE (and the Java Card virtual machine).

FMT_MSA.3.2[FIREWALL Editorially Refined]

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

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

Hierarchical-To No other components.

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

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

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

FMT_SMF.1	**Specification of Management Functions**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions: **[assignment:** • **modify the Currently Active Context, the Selected Applet Context and the Active Applets** **]**

FMT_SMR.1	**Security roles**
Hierarchical-To	No other components.
Dependencies	FIA_UID.1 Timing of identification
FMT_SMR.1.1	The TSF shall maintain the roles: **[assignment:** • **Java Card RE (JCRE),** • **Java Card VM (JCVM).** **].**
FMT_SMR.1.2	The TSF shall be able to associate users with roles.

7.1.1.2 Application Programming Interface

The following SFRs are related to the Java Card API. The whole set of cryptographic algorithms is generally not implemented because of limited memory resources and/or limitations due to exportation. Therefore, the following requirements only apply to the implemented subset. It should be noticed that the execution of the additional native code is not within the TSF. Nevertheless, access to API native methods from the Java Card System is controlled by TSF because there is no difference between native and interpreted methods in their interface or invocation mechanism.

FCS_CKM.1	**Cryptographic key generation**
Hierarchical-To	No other components.
Dependencies	[FCS_CKM.2 Cryptographic key distribution, or FCS_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 RSA: LENGTH_RSA_2048 and from 2000 bit to 2048 bit in one bit steps]** that meet the following: **[assignment: [1]]**.

AppNote

- The keys can be generated and diversified in accordance with [24] specification in classe KeyBuilder.
- This component shall be instantiated according to the version of the Java Card API applying to the security target and the implemented algorithms ([24]).

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 and secure messaging]** that meets the following: **[assignment: [24], [9]]**.

AppNote

- The keys can be accessed as specified in [24] Key class and [9] 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 [24] and [9] for proprietary classes.

FCS_CKM.3 **Cryptographic key access**

Hierarchical-To

No other components.

Dependencies

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

FCS_CKM.3.1

The TSF shall perform **[assignment: management of DES, AES, RSA, RSA-CRT]** in accordance with a specified cryptographic key access method **[assignment: methods/commands defined in packages javacard.security of [24] and [9] for proprietary classes]** that meets the following: **[assignment:** [24], [9]**]**.

AppNote

- The keys can be accessed as specified in [24] Key class and [9] 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 ([24]) and [9] for proprietary classes.

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]**.

AppNote

- The keys are reset as specified in [24] Key class, with the method clearKey(). Any access to a cleared key for ciphering or signing shall throw an exception.
- This component shall be instantiated according to the version of the Java Card API applicable to the security target and the implemented algorithms ([24]).

FCS_COP.1 **Cryptographic operation**

Hierarchical-To No other components.

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

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 [24]]**.

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_ECB_NOPAD, ALG_AES_CBC_ISO9797_M1, ALG_AES_CBC_ISO9797_M2, ALG_AES_ECB_ISO9797_M1, ALG_AES_ECB_ISO9797_M2]** 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** [24]**]**.

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 and from 2000 bit to 2048 bit in one bit steps]** that meet the following: **[assignment: Java Card API Spec** [24] **and for the one bit step range see API specified in JCOPX** [9]**]**.

FCS_COP.1.1[DESMAC] The TSF shall perform **[assignment: 8 byte MAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: Triple-DES in outer CBC for Mode ALG_DES_MAC8_ISO9797_1_M1_ALG3, ALG_DES_MAC8_ISO9797_1_M2_ALG3, ALG_DES_MAC8_ISO9797_M1, ALG_DES_MAC8_ISO9797_M2, ALG_DES_MAC8_NOPAD, ALG_DES_MAC4_ISO9797_1_M2_ALG3, ALG_DES_MAC4_ISO9797_M2]** and cryptographic key sizes **[assignment: LENGTH_DES3_2KEY, LENGTH_DES3_3KEY]** that meet the following: **[assignment: Java Card API Spec** [24]**]**.

FCS_COP.1.1[AESMAC] The TSF shall perform **[assignment: 16 byte MAC generation and verification]** in accordance with a specified cryptographic algorithm **[assignment: AES in CBC Mode ALG_AES_MAC_128_NOPAD]** and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_AES_256 bit]** that meet the following: **[assignment: Java Card API Spec** [24]**]**.

FCS_COP.1.1[RSASignaturePKCS1] The TSF shall perform **[assignment: digital signature generation and verification]** in accordance with a specified cryptographic algorithm **[assignment:ALG_RSA_SHA_224_PKCS1, ALG_RSA_SHA_224_PKCS1_PSS, ALG_RSA_SHA_256_PKCS1, ALG_RSA_SHA_256_PKCS1_PSS, ALG_RSA_SHA_384_PKCS1, ALG_RSA_SHA_384_PKCS1_PSS, ALG_RSA_SHA_512_PKCS1, ALG_RSA_SHA_512_PKCS1_PSS 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_OAEP]** and cryptographic key sizes **[assignment: LENGTH_RSA_2048 and from 2000 bit to 2048 bit in one bit steps]** that meet the following: **[assignment: Java Card API Spec** [24] **and for the one bit step range see API specified in JCOPX** [9]**]**.

FCS_COP.1.1[SHA] The TSF shall perform **[assignment: secure hash computation]** in accordance with a specified cryptographic algorithm **[assignment: ALG_SHA_224, ALG_SHA_256, ALG_SHA_384, ALG_SHA_512]** and cryptographic key sizes **[assignment: LENGTH_SHA_224, LENGTH_SHA_256, LENGTH_SHA_384, LENGTH_SHA_512]** that meet the following: **[assignment: Java Card API Spec** [24] **and JCOPX** [9]**]**.

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_CMAC16, SIG_CIPHER_AES_CMAC16]** and cryptographic key sizes **[assignment: LENGTH_AES_128, LENGTH_AES_192 and LENGTH_AES_256 bit]** that meet the following: **[assignment: see API specified in JCOPX [9]]**.

FCS_COP.1.1[DAP] The TSF shall perform **[assignment: verification of the DAP signature attached to**

Executable Load Applications] in accordance with a specified cryptographic algorithm **[assignment: ALG_RSA_SHA_PKCS1]** and cryptographic key sizes **[assignment: LENGTH_ RSA_1024]** that meet the following: **[assignment: GP Spec [22]]**.

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]**.

AppNote The events that provoke the de-allocation of a transient object are described in [25], §5.1.

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

Hierarchical-To No other components.

Dependencies No dependencies.

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

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

FDP_RIP.1[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]**.

AppNote

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

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

Hierarchical-To

No other components.

Dependencies

No dependencies.

FDP_RIP.1.1[KEYS]

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

AppNote

- The javacard.security and javacardx.crypto packages do provide secure interfaces to the cryptographic buffer in a transparent way. See javacard.security.KeyBuilder and Key interface of [24].

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]**.

AppNote

- The events that provoke the de-allocation of any transient object are described in [25], §5.1.
- The clearing of CLEAR_ON_DESELECT objects is not necessarily performed when the owner of the objects is deselected. In the presence of multiselectable applet instances, CLEAR_ON_DESELECT memory segments may be attached to applets that are active in different logical channels. Multiselectable applet instances within a same package must share the transient memory segment if they are concurrently active ([25], §4.2.)

FDP_ROL.1[FIREWALL] Basic rollback (FIREWALL)

Hierarchical-To No other components.

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

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

FDP_ROL.1.2[FIREWALL] The TSF shall permit operations to be rolled back within the **[assignment: scope of a select(), deselect(), process(), install() or uninstall() call, notwithstanding the restrictions given in** [25], §7.7, **within the bounds of the Commit Capacity** ([25], §7.8), **and those described in** [24]**]**.

AppNote Transactions are a service offered by the APIs to applets. It is also used by some APIs to guarantee the atomicity of some operation. This mechanism is either implemented in Java Card platform or relies on the transaction mechanism offered by the underlying platform. Some operations of the API are not conditionally updated, as documented in [24] (see for instance, PIN-blocking, PIN-checking, update of Transient objects).

7.1.1.3 Card Security Management

FAU_ARP.1 Security alarms

Hierarchical-To No other components.

Dependencies FAU_SAA.1 Potential violation analysis

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

- **throw an exception,**
- **reset the card session (after a predefined number of resetted sessions the card gets terminated),**
- **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: CAP file inconsistency **(response with error code to S.CAD)**,
- typing error in the operands of a bytecode **(Only possible in BCV)**,
- LFC: applet life cycle inconsistency **(reset the card session)**,
- CHP: card tearing (unexpected removal of the Card out of the CAD) and power failure **(reset the card session)**,
- ABT: abort of a transaction in an unexpected context **(throw an exception)**,

- FWL: violation of the Firewall or JCVM SFPs **(throw an exception)**,
- RSC: unavailability of resources **(throw an exception)**,
- OFL: array overflow **(throw an exception)**,
- EDC: checksum mismatch of EDC arrays **(throw an exception)**,
- assignment:
  - CHP: Abnormal environmental condition (Frequency, Voltage, Temperature) **(reset the card session)**,
  - Physical Tampering
    - CLC: Card Manager Life Cycle inconsistency **(reset the card session)**,
    - CHP: General Fault Injection Detection **(reset the card session)**
  - CHP: EEPROM defects **(reset the card session)**,
  - CHP: Integrity protected persistent data inconsistency **(reset the card session)**,
  - CHP: Integrity protected transient data inconsistency **(reset the card session)**,
  - Memory Access Violation
    - CHP: Others **(reset the card session)**

.

AppNote

- 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 ”resetting 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 resetting should occur when ”clean” re-initialization seems to be impossible.
- The resetting may be implemented at the level of the Java Card System as a denial of service (through some systematic ”fatal error” message or return value) that lasts

up to the next ”RESET” event, without affecting other components of the card (such as the card manager). Finally, because the installation of applets is a sensitive process, security alerts in this case should also be carefully considered herein.

FDP_SDI.2 **Stored data integrity monitoring and action**

Hierarchical-To FDP_SDI.1 Stored data integrity monitoring

Dependencies No dependencies.

FDP_SDI.2.1 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 Upon detection of a data integrity error, the TSF shall **[assignment: reset the card session for integrity errors]**.

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

- **D.APP_KEYs**
- **D.PIN**
- **D.TSF_KEYs**

AppNote

- Although no such requirement is mandatory in the Java Card specification, at least an exception shall be raised upon integrity errors detection on cryptographic keys, PIN values and their associated security attributes. Even if all the objects cannot be monitored, cryptographic keys and PIN objects shall be considered with particular attention by ST authors as they play a key role in the overall security.
- It is also recommended to monitor integrity errors in the code of the native applications and Java Card applets.
- For integrity sensitive application, their data shall be monitored (D.APP_I_DATA): applications may need to protect information against unexpected modifications, and explicitly control whether a piece of information has been changed between two accesses. For example, maintaining the integrity of an electronic purse’s balance is extremely important because this value represents real money. Its modification must be controlled, for illegal ones would denote an important failure of the payment system.
- A dedicated library could be implemented and made available to developers to achieve better security for specific objects, following the same pattern that already exists in cryptographic APIs, for instance.

FPR_UNO.1 **Unobservability**

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.TSF_KEYs, D.Crypto]** by **[assignment: another user]**.

AppNote Although it is not required in [25] specifications, the non-observability of operations on sensitive information such as keys appears as impossible to circumvent in the smart card world. The precise list of operations and objects is left unspecified, but should at least concern secret keys and PIN codes when they exists on the card, as well as the cryptographic operations and comparisons performed on them.

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]**.

AppNote The Java Card RE Context is the Current context when the Java Card VM begins running after a card reset ([25], §6.2.3) or after a proximity card (PICC) activation sequence ([25]). Behavior of the TOE on power loss and reset is described in [25], §3.6 and §7.1. Behavior of the TOE on RF signal loss is described in [25], §3.6.1.

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 [26] specification**
- **the API tokens defined in the export files of reference implementation**

- **The ISO 7816-6 rules**
- **The EMV specification**

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

AppNote

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

7.1.1.4 AID Management

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

Hierarchical-To

No other components.

Dependencies

No dependencies.

FIA_ATD.1.1[AID]

The TSF shall maintain the following list of security attributes belonging to individual users: **[assignment:**

- **Package AID,**
- **Applet's Version Number,**
- **Registered Applets,**
- **Applet Selection Status ([25], §4.6).**

].

Refinement

"Individual users" stands for applets.

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

Hierarchical-To

FIA_UID.1 Timing of identification

Dependencies

No dependencies.

FIA_UID.2.1[AID]

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

AppNote

- By users here it must be understood the ones associated to the packages (or applets) that act as subjects of policies. In the Java Card System, every action is always performed by an identified user interpreted here as the currently selected applet or the package that is the subject's owner. Means of identification are provided during the loading procedure of the package and the registration of applet instances.

- The role Java Card RE defined in FMT_SMR.1 is attached to an IT security function rather than to a ”use” of the CC terminology. The Java Card RE does not ”identify” itself to the TOE, but it is part of it.

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

Hierarchical-To No other components.

Dependencies FIA_ATD.1 User attribute definition

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

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

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

Hierarchical-To No other components.

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

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

AppNote

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

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

Hierarchical-To — No other components.

Dependencies — FMT_MTD.1 Management of TSF data

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

7.1.2 INSTG Security Functional Requirements

The list of SFRs of this category are take from [8]. The SFR FDP_ITC.2[INSTALLER] has been refined and is now part of the card management SFRs (FDP_ITC.2[CCM]) in section 7.1.6.

FMT_SMR.1[INSTALLER] — **Security roles (INSTALLER)**

Hierarchical-To — No other components.

Dependencies — FIA_UID.1 Timing of identification

FMT_SMR.1.1[INSTALLER] — The TSF shall maintain the roles: **[assignment: Installer]**.
FMT_SMR.1.2[INSTALLER] — The TSF shall be able to associate users with roles.

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

Hierarchical-To — No other components.

Dependencies — No dependencies.

FPT_FLS.1.1[INSTALLER] — The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the installer fails to load/install a package/applet as described in [25], §11.1.5]**.

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

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

Hierarchical-To — FPT_RCV.2 Automated recovery

Dependencies — AGD_OPE.1 Operational user guidance

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

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

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

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

AppNote FPT_RCV.3.1[Installer]:

- This element is not within the scope of the Java Card specification, which only mandates the behavior of the Java Card System in good working order. Further details on the ”maintenance mode” shall be provided in specific implementations. The following is an excerpt from [3], p298: In this maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only authorised users should be allowed access to this mode but the real details of who can access this mode is a function of FMT: Security management. If FMT: Security management does not put any controls on who can access this mode, then it may be acceptable to allow any user to restore the system if the TOE enters such a state. However, in practice, this is probably not desirable as the user restoring the system has an opportunity to configure the TOE in such a way as to violate the SFRs.

FPT_RCV.3.2[Installer]:

- Should the installer fail during loading/installation of a package/applet, it has to revert to a ”consistent and secure state”. The Java Card RE has some clean up duties as well; see [25], §11.1.5 for possible scenarios. Precise behavior is left to implementers. This component shall include among the listed failures the deletion of a package/applet. See ([25], §11.3.4) for possible scenarios. Precise behavior is left to implementers.
- Other events such as the unexpected tearing of the card, power loss, and so on, are partially handled by the underlying hardware platform (see [11]) and, from the TOE's side, by events ”that clear transient objects” and transactional features. See FPT_FLS.1.1, FDP_RIP.1[TRANSIENT], FDP_RIP.1[ABORT] and FDP_ROL.1[FIREWALL].

FPT_RCV.3.3[Installer]:

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

7.1.3 ADELG Security Functional Requirements

The list of SFRs of this category are taken from [8].

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,
- OP.DELETE_PCKG_APPLET.

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

Hierarchical-To No other components.

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

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

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) 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 ([25], §11.3.4.2, 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 ([25], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.15 ([25], §11.3.4.2.1, 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 ([25], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.16 ([25], §11.3.4.3, Applet/Library Package Deletion): S.ADEL may perform OP.DELETE_PCKG 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 ([25], §11.3.4.4, Applet Package and Contained Instances Deletion): S.ADEL may perform OP.DELETE_PCKG_APPLET 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 ([25], §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-EditoriallyRefined] The TSF shall explicitly deny access of **[assignment: any subject but S.ADEL to O.CODE_PKG or O.APPLET for the purpose of deleting them from the card]**.

AppNote FDP_ACF.1.2[ADEL]:

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

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

Hierarchical-To No other components.

Dependencies No dependencies.

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

AppNote Deleted freed resources (both code and data) may be reused, depending on the way they were deleted (logically or physically). Requirements on de-allocation during applet/package deletion are described in [25], §11.3.4.1, §11.3.4.2 and §11.3.4.3.

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

Hierarchical-To No other components.

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

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

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

Hierarchical-To No other components.

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

FMT_MSA.3.1[ADEL] The TSF shall enforce the**[assignment: ADEL access control SFP]** to provide **[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.

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]**.

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

Hierarchical-To No other components.

Dependencies FIA_UID.1 Timing of identification

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

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

FPT_FLS.1[ADEL]	**Failure with preservation of secure state (ADEL)**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FPT_FLS.1.1[ADEL]	The TSF shall preserve a secure state when the following types of failures occur: **[assignment: the applet deletion manager fails to delete a package/applet as described in [25], §11.3.4.]**
AppNote	• The TOE may provide additional feedback information to the card manager in case of a potential security violation (see FAU_ARP.1). • The Package/applet instance deletion must be atomic. The ”secure state” referred to in the requirement must comply with Java Card specification ([25], §11.3.4.)

7.1.4 RMIG Security Functional Requirements

Not used in this ST because RMI is optional in PP [8] and the TOE does not support RMI.

7.1.5 ODELG Security Functional Requirements

The list of SFRs of this category are take from [8].

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()]**.
AppNote	• Freed data resources resulting from the invocation of the method javacard.framework.JCSystem.requestObjectDeletion() may be reused. Requirements on de-allocation after the invocation of the method are described in [24]. • There is no conflict with FDP_ROL.1 here because of the bounds on the rollback mechanism: the execution of requestObjectDeletion() is not in the scope of the rollback because it must be performed in between APDU command processing, and therefore no transaction can be in progress.

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

Hierarchical-To No other components.

Dependencies No dependencies.

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

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

7.1.6 CarG Security Functional Requirements

The card management SFRs from the PP [8] are refined and replaced by the following SFRs.

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

Hierarchical-To No other components.

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

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

FDP_UIT.1.2[CCM] The TSF shall be able to determine on receipt of user data, whether **[selection: modification, deletion, insertion, replay]** has occurred.

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

Hierarchical-To No other components.

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

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

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

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

Hierarchical-To No other components.

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

FDP_ITC.2.1[CCM] The TSF shall enforce the **[assignment: Firewall access control SFP 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 ([26], §4.5.2). **]**

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 [25], Section 11.1.5]**

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

Hierarchical-To No other components.

Dependencies FDP_ACF.1 Security attribute based access control

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

- **Subjects: S.INSTALLER, S.ADEL, S.CAD (from [8]) and S.SD**

- **Objects: Delegation Token, DAP Block and Load File**
- **Operations: GlobalPlatform's card content management APDU commands and API methods**]

FDP_ACF.1[SD]	**Security attribute based access control (SD)**
Hierarchical-To	No other components.
Dependencies	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1[SD]	The TSF shall enforce the **[assignment: Security Domain access control policy]** to objects based on the following: **[assignment:**

- **Subjects:**
  - **S.INSTALLER, defined in [8] and represented by the GlobalPlatform Environment (OPEN) on the card, the Card Life Cycle attributes (defined in Section 5.1.1 of [20])**
  - **S.ADEL, also defined in [8] 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 [20]), a Life-cycle Status (defined in Section 5.3.2 of [20]) and a Secure Communication Security Level (defined in Section 10.6 of [20])**
  - **S.CAD, defined in [8], 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 [20])**
- **installation (Section 9.3.6 of [20])**
- **extradition (Section 9.4.1 of [20])**
- **registry update (Section 9.4.2 of [20])**
- **content removal (Section 9.5 of [20]).]**

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

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

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

Hierarchical-To No other components.

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

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

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].

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 [20])**
  - **application locking and unlocking (Section 9.6.2 of [20])**
  - **card termination (Section 9.6.4 of [20])**
  - **card status interrogation (Section 9.6.6 of [20])**
  - **application status interrogation (Section 9.6.5 of [20]).**

FMT_SMR.1[SD] Security roles (SD)

Hierarchical-To No other components.

Dependencies FIA_UID.1 Timing of identification

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

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

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

Hierarchical-To FCO_NRO.1 Selective proof of origin.

Dependencies FIA_UID.1 Timing of identification.

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

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

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

AppNote FCO_NRO.2.1[SC]:

- Upon reception of a new application package for installation, the card manager shall first check that it actually comes from the verification authority. The verification authority is the entity responsible for bytecode verification.

FCO_NRO.2.3[SC]:

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

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

Hierarchical-To FDP_IFC.1 Subset information flow control

Dependencies FDP_IFF.1 Simple security attributes

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

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

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

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

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

Hierarchical-To No other components.

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

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

- **Subjects:**
  - **S.SD receiving the Card Content Management commands (through APDUs or APIs).**
  - S.CAD **the off-card entity that communicates with the S.SD.**
- **Information:**
  - **executable load file, in case of application loading;**
  - **applications or SD privileges, in case of application installation or registry update;**

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 [20]);**
  - **installation (Section 9.3.6 of [20]);**
  - **extradition (Section 9.4.1 of [20]);**
  - **registry update (Section 9.4.2 of [20]);**
  - **content removal (Section 9.5 of [20]).**

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.

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,**
- **Security 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.]**

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]** to specify alternative initial values to override the default values when an object or information is created.

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

Hierarchical-To No other components.

Dependencies No dependencies.

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

- **Management functions specified in GlobalPlatform specifications [GP]:**
  - **loading (Section 9.3.5 of [20])**
  - **installation (Section 9.3.6 of [20])**
  - **extradition (Section 9.4.1 of [20])**
  - **registry update (Section 9.4.2 of [20])**
  - **content removal (Section 9.5 of [20]).**

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

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

Hierarchical-To No other components.

Dependencies No dependencies.

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

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

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

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

AppNote The GlobalPlatform TSF mediated actions listed in [GP] such as selecting an application, requesting data, initializing, etc.

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

Hierarchical-To No other components.

Dependencies FIA_UID.1 Timing of identification

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

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

FIA_UAU.4[SC]	**Single-use authentication mechanisms**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FIA_UAU.4.1[SC]	The TSF shall prevent reuse of authentication data related to **[assignment: the authentication mechanism used to open a secure communication channel with the card.]**

FTP_ITC.1[SC]	**Inter-TSF trusted channel (SC)**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FTP_ITC.1.1[SC]	The TSF shall provide a communication channel between itself and another trusted IT that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2[SC]	The TSF shall permit **[selection: another trusted IT product]** to initiate communication via the trusted channel.
FTP_ITC.1.3[SC]	The TSF shall initiate communication via the trusted channel for **[assignment: all card management functions:** • **loading** • **installation** • **extradition** • **registry update** • **content removal** • **changing the Application Life Cycle or card Life Cycle.]**

7.1.7 EMG Security Functional Requirements

The list of SFRs of this category are take from [8].

7.1.7.1 External Memory

FDP_ACC.1[EXT-MEM]	**Subset access control (EXT-MEM)**
Hierarchical-To	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1[EXT-MEM] The TSF shall enforce the **[assignment: EXTERNAL MEMORY access control SFP]** on **[assignment: subject S.APPLET, object O.EXT_MEM_INSTANCE, and operations OP.CREATE_EXT_MEM_INSTANCE, OP.READ_EXT_MEM and OP.WRITE_EXT_MEM]**.

FDP_ACF.1[EXT-MEM] **Security attribute based access control (EXT-MEM)**

Hierarchical-To No other components.

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

FDP_ACF.1.1[EXT-MEM] The TSF shall enforce the **[assignment: EXTERNAL MEMORY access control SFP]** to objects based on the following: **[assignment:**

Object	Security attribute
O.EXT_MEM_INSTANCE	**Address Space.**

]

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

- R.JAVA.20: Any subject S.APPLET that performs OP.CREATE_EXT_MEM_INSTANCE obtains an object O.EXT_MEM_INSTANCE that addresses a memory space different from that of the Java Card System.
- R.JAVA.21: Any subject S.APPLET may perform OP.READ_EXT_MEM (O.EXT_MEM_INSTANCE, address) provided the address belongs to the space of the O.EXT_MEM_INSTANCE.
- R.JAVA.22: Any subject S.APPLET may perform OP.WRITE_EXT_MEM (O.EXT_MEM_INSTANCE, address) provided the address belongs to the space of the O.EXT_MEM_INSTANCE.

]

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

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

AppNote The actual mechanism for creating an instance of external memory is implementation dependent. This rule only states that the accessible address space must not interfere with that of the Java Card System. The creation and the access to an external memory instance fall in the scope of the Firewall rules.

FMT_MSA.1[EXT-MEM] **Management of security attributes (EXT-MEM)**

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[EXT-MEM] The TSF shall enforce the **[assignment: EXTERNAL MEMORY access control SFP]** to restrict the ability to **[assignment: set up]** the security attributes **[assignment: Address Space]** to **[assignment: the S.JCRE]**.

FMT_MSA.3[EXT-MEM] **Static attribute initialisation (EXT-MEM)**

Hierarchical-To No other components.

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

FMT_MSA.3.1[EXT-MEM] The TSF shall enforce the **[assignment: EXTERNAL MEMORY access control SFP]** to provide **[assignment: no]** default values for security attributes that are used to enforce the SFP.

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

AppNote Upon creation of an external memory instance, the Java Card RE gets the address space value for the newly created object. This is implementation-dependent.

FMT_SMF.1[EXT-MEM] **Specification of Management Functions (EXT-MEM)**

Hierarchical-To No other components.

Dependencies No dependencies.

FMT_SMF.1.1[EXT-MEM] The TSF shall be capable of performing the following management functions: **[assignment: set up the Address Space security attribute]**.

7.1.8 Further Security Functional Requirements

The SFRs in this section provide additional proprietary features.

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

Hierarchical-To No other components.

Dependencies — No other components.

FAU_SAS.1.1[SCP] — The TSF shall provide **[assignment: 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]**.

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: hybrid 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 [27]) 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 [27]).
- (DRG.3.3) The RNG provides enhanced backward secrecy even if the current internal state is known (as defined in [27])

]

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 [27]) as random source, generates output for which in AES mode 2^{48} and in 3DES mode 2^{35} strings of bit length 128 are mutually different with probability at least $1 - 2^{-24}$ in AES mode and $1 - 2^{-17}$ in 3DES mode.
- (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 [27])

]

AppNote — This functionality is provided by the certified Crypto Lib, see [6]

AppNote — Random numbers according to FCS_RNG.1 can be created by using the RandomData class defined in the Java Card API Specification [24] in ALG_PSEUDO_RANDOM mode and ALG_SECURE_RANDOM mode.

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.

7.1.9 Configuration Security Functional Requirements

FDP_ACC.1[CONFIG-APPLET] **Complete access control (CONFIG-APPLET)**

Hierarchical-To No other components.

Dependencies FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1[CONFIG-APPLET] The TSF shall enforce the **[assignment: CONFIGURATION AREA access control SFP]** on **[assignment: S.CONFIG-APPLET, O.CONFIG-ITEM, and OP.READ_CONFIG_ITEM, OP.MODIFY_CONFIG_ITEM, OP.USE_CONFIG_ITEM, OP.CREATE_CARD_MANAGER and OP.IDENTIFY_TOE]**.

FDP_ACF.1[CONFIG-APPLET] **Security based access control (CONFIG-APPLET)**

Hierarchical-To No other components.

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

FDP_ACF.1.1[CONFIG-APPLET] The TSF shall enforce the **[assignment: CONFIGURATION AREA access control SFP]** to objects based on the following: **[assignment: S.CONFIG-APPLET, O.CONFIG-ITEM and the SFP-relevant security attributes Config Item Tag, Config Item Content, ISD default keys and EEPROM image]**.

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

- **Only S.CONFIG-APPLET shall be able to modify O.CONFIG-ITEM,**
- **A Config Item shall be accessible only via its associated Config Item Tag,**

- **The Config Item Content shall be accessible only via O.CONFIG-ITEM,**
- **OP.READ_CONFIG_ITEM shall never be possible for ISD default keys,**
- **OP.READ_CONFIG_ITEM shall never be possible for the EEPROM image,**
- **Any of the allowed operations OP.READ_CONFIG_ITEM and OP.MODIFY_CONFIG_ITEM of O.CONFIG-ITEM and OP.CREATE_CARD_MANAGER of O.CONFIG-APPLET shall be allowed only for authorized users**

].

FDP_ACF.1.3[CONFIG-APPLET] The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment:**

- **OP.USE_CONFIG_ITEM is always allowed for subjects S.JCRE, S.SD and S.CONFIG-APPLET,**
- **OP.IDENTIFY_TOE shall be allowed for all users**

].

FDP_ACF.1.4[CONFIG-APPLET] The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment: after deletion of S.CONFIG-APPLET the operation OP.IDENTIFY_TOE, OP.CREATE_CARD_MANAGER and the operations OP.READ_CONFIG_ITEM and OP.MODIFY_CONFIG_ITEM of any O.CONFIG-ITEM shall not be allowed]**.

FMT_MSA.1[CONFIG-APPLET] Management of security attributes (CONFIG-APPLET)

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[CONFIG-APPLET] The TSF shall enforce the **[assignment: CONFIGURATION AREA access control SFP]** to restrict the ability to **[selection: modify, [assignment: read]]** the security attributes **[assignment: Config Item Tag and Config Item Content, ISD Default Keys and EEPROM Image]** to **[assignment: S.CONFIG-APPLET]**.

FMT_MSA.3[CONFIG-APPLET] Static attribute initialisation (CONFIG-APPLET)

Hierarchical-To No other components.

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

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

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

FMT_SMF.1[CONFIG-APPLET] Specification of Management Functions (CONFIG-APPLET)

Hierarchical-To No other components.

Dependencies No dependencies.

FMT_SMF.1.1[CONFIG-APPLET] The TSF shall be capable of performing the following management functions: **[assignment: modify the Config Item Content, the ISD Default Keys or the EEPROM Image and create the Card Manager]**.

FMT_SMR.1[CONFIG-APPLET] Security roles (CONFIG-APPLET)

Hierarchical-To No other components.

Dependencies FIA_UID.1 Timing of identification.

FMT_SMR.1.1[CONFIG-APPLET] The TSF shall maintain the roles **[assignment: CONFIG-APPLET]**.

FMT_SMR.1.2[CONFIG-APPLET] The TSF shall be able to associate users with roles.

FIA_UID.1[CONFIG-APPLET] Timing of identification (CONFIG-APPLET)

Hierarchical-To No other components.

Dependencies No dependencies.

FIA_UID.1.1[CONFIG-APPLET] The TSF shall allow **[assignment: execution of OP.IDENTIFY_TOE]** on behalf of the user to be performed before the user is identified.

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

FIA_UAU.1[CONFIG-APPLET] Timing of authentication (CONFIG-APPLET)

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

7.2 Security Assurance Requirements

The assurance requirements of this evaluation are EAL5 augmented by AVA_VAN.5, ALC_DVS.2, ASE_TSS.2, and ALC_FLR.1. The assurance requirements ensure, among others, the security of the TOE during its development and production.

7.3 Security Requirements Rationale for the TOE

7.3.1 Identification

OT.SID

SFR	Rationale
FIA_UID.2[AID]	Subjects' identity is AID-based (applets, packages) and is met by the SFR. Installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities and is met by the SFR.
FIA_USB.1[AID]	Subjects' identity is AID-based (applets, packages) and is met by the SFR. Installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities and is met by the SFR.
FMT_MSA.1[JCRE]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.1[JCVM]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.1[ADEL]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.3[FIREWALL]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.
FMT_MSA.3[JCVM]	Subjects' identity is AID-based (applets, packages) and is met by the SFR.

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

7.3.2 Execution

OT.FIREWALL

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

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

OT.GLOBAL_ARRAYS_CONFID

SFR	Rationale
FDP_IFC.1[JCVM]	The JCVM information flow control policy meets the objective by preventing an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.
FDP_IFF.1[JCVM]	The JCVM information flow control policy meets this objective by preventing an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.
FDP_RIP.1[OBJECTS]	Contributes to meet the objective by protecting the array parameters of remotely invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[ABORT]	Contributes to meet the objective by protecting the array parameters of remotely invoked methods, which are global as well, through the general initialization of method parameters.

SFR	Rationale
FDP_RIP.1[APDU]	Only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer and the global byte array input parameter (bArray) to an applet's install method. Contributes to meet this objective by fulfilling the clearing requirement of these arrays.
FDP_RIP.1[bArray]	Only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer and the global byte array input parameter (bArray) to an applet's install method. Contributes to meet this objective by fulfilling the clearing requirement of these arrays.
FDP_RIP.1[KEYS]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[TRANSIENT]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[ADEL]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.
FDP_RIP.1[ODEL]	Contributes to meet the objective by protecting the array parameters of invoked methods, which are global as well, through the general initialization of method parameters.

OT.GLOBAL_ARRAYS_INTEG

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

OT.NATIVE

SFR	Rationale
FDP_ACF.1[FIREWALL]	Covers this objective by ensuring that the only means to execute native code is the invocation of a Java Card API method. This objective mainly relies on the environmental objective OE.APPLET, which uphold the assumption A.APPLET.

OT.OPERATE

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

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

OT.REALLOCATION

SFR	Rationale
FDP_RIP.1[OBJECTS]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[ABORT]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[APDU]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[bArray]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[KEYS]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[TRANSIENT]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[ADEL]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.
FDP_RIP.1[ODEL]	Contributes to meet the objective by imposing that the contents of the re-allocated block shall always be cleared before delivering the block.

OT.RESOURCES

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

7.3.3 Services

OT.ALARM

SFR	Rationale
FAU_ARP.1	Contributes to meet this objective by defining TSF reaction upon detection of a potential security violation

SFR	Rationale
FPT_FLS.1	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur
FPT_FLS.1[INSTALLER]	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur
FPT_FLS.1[ADEL]	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur
FPT_FLS.1[ODEL]	Contributes to meet the objective by providing the guarantee that a secure state is preserved by the TSF when failures occur

OT.CIPHER

SFR	Rationale
FCS_CKM.1	Covers the objective directly
FCS_CKM.2	Covers the objective directly
FCS_CKM.3	Covers the objective directly
FCS_CKM.4	Covers the objective directly
FCS_COP.1	Covers the objective directly
FPR_UNO.1	Contributes to meet the objective by controlling the observation of the cryptographic operations which may be used to disclose the keys

OT.KEY-MNGT

SFR	Rationale
FCS_CKM.1	Covers the objective directly
FCS_CKM.2	Covers the objective directly
FCS_CKM.3	Covers the objective directly
FCS_CKM.4	Covers the objective directly
FCS_COP.1	Covers the objective directly
FDP_RIP.1[OBJECTS]	Covers the objective directly
FDP_RIP.1[ABORT]	Covers the objective directly
FDP_RIP.1[APDU]	Covers the objective directly
FDP_RIP.1[bArray]	Covers the objective directly
FDP_RIP.1[KEYS]	Covers the objective directly
FDP_RIP.1[TRANSIENT]	Covers the objective directly
FDP_RIP.1[ADEL]	Covers the objective directly

SFR	Rationale
FDP_RIP.1[ODEL]	Covers the objective directly
FDP_SDI.2	Covers the objective directly
FPR_UNO.1	Contributes to meet objective by controlling the observation of the cryptographic operations which may be used to disclose the keys

OT.PIN-MNGT

SFR	Rationale
FDP_ACC.2[FIREWALL]	Contributes to meet the objective by protecting the access to private and internal data of the objects
FDP_ACF.1[FIREWALL]	Contributes to meet the objective by protecting the access to private and internal data of the objects
FDP_RIP.1[OBJECTS]	Contributes to meet the objective
FDP_RIP.1[ABORT]	Contributes to meet the objective
FDP_RIP.1[APDU]	Contributes to meet the objective
FDP_RIP.1[bArray]	Contributes to meet the objective
FDP_RIP.1[KEYS]	Contributes to meet the objective
FDP_RIP.1[TRANSIENT]	Contributes to meet the objective
FDP_RIP.1[ADEL]	Contributes to meet the objective
FDP_RIP.1[ODEL]	Contributes to meet the objective
FDP_ROL.1[FIREWALL]	Contributes to meet the objective
FDP_SDI.2	Contributes to meet the objective
FPR_UNO.1	Contributes to meet the objective

OT.TRANSACTION

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

7.3.4 Object Deletion

OT.OBJ-DELETION

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

7.3.5 Applet Management

OT.APPLI-AUTH

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

OT.DOMAIN-RIGHTS

SFR	Rationale
FDP_ACC.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FDP_ACF.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_MSA.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_MSA.3[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_SMF.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
FMT_SMR.1[SD]	Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.

SFR	Rationale
FTP_ITC.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FCO_NRO.2[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FDP_IFC.2[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FDP_IFF.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_MSA.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_MSA.3[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FMT_SMF.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UID.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UAU.1[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
FIA_UAU.4[SC]	Contributes to cover this security objective by enforcing Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.

OT.COMM_AUTH

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

OT.COMM_INTEGRITY

SFR	Rationale
FCS_COP.1	Contributes to meet the security objective by by specifying secure cryptographic algorithm that shall be used to ensure the integrity of the card management commands.
FMT_SMR.1[SD]	Contributes to cover this security objective by defining the roles enabling to send and authenticate the card management requests for which the integrity has to be ensured.
FTP_ITC.1[SC]	Contributes to meet the security objective by ensuring the integrity of card management commands.

SFR	Rationale
FDP_IFC.2[SC]	Contributes to cover the security objective by enforcing the Secure Channel Protocol information flow control policy to guarantee the integrity of administration requests.
FDP_IFF.1[SC]	Contributes to cover the security objective by enforcing the Secure Channel Protocol information flow control policy to guarantee the integrity of administration requests.
FMT_MSA.1[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the integrity of card management requests.
FMT_MSA.3[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the integrity of card management requests.
FMT_SMF.1[SC]	Contributes to meet the security objective by specifying the actions activating the integrity check on the card management commands.

OT.COMM_CONFIDENTIALITY

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

SFR	Rationale
FMT_MSA.1[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the confidentiality of card management requests by decrypting those requests and imposing management conditions on that attributes.
FMT_MSA.3[SC]	Contributes to cover the security objective by specifying security attributes enabling to guarantee the confidentiality of card management requests by decrypting those requests and imposing management conditions on that attributes.
FMT_SMF.1[SC]	Contributes to cover the security objective by specifying the actions ensuring the confidentiality of the card management commands.

7.3.6 External Memory

OT.EXT-MEM

SFR	Rationale
FDP_ACC.1[EXT-MEM]	Contributes to meet the objective by the EXTERNAL MEMORY access control policy which protects the Java Card system memory against applet's attempts of unauthorized access through the external memory facilities.
FDP_ACF.1[EXT-MEM]	Contributes to meet the objective by the EXTERNAL MEMORY access control policy which protects the Java Card system memory against applet's attempts of unauthorized access through the external memory facilities.
FMT_SMF.1[EXT-MEM]	Contributes to meet the objective by controlling the external memory management

7.3.7 Card Management

OT.CARD-MANAGEMENT

SFR	Rationale
FDP_ACC.2[ADEL]	Contributes to meet the objective by the ADEL access control policy which ensures the non-introduction of security holes. The integrity and confidentiality of data that does not belong to the deleted applet or package is a by-product of this policy as well

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

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

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

7.3.8 Smart Card Platform

OT.SCP.IC

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

OT.SCP.RECOVERY

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

OT.SCP.SUPPORT

SFR	Rationale
FCS_CKM.1	Contributes to meet the objective
FCS_CKM.4	Contributes to meet the objective
FCS_COP.1	Contributes to meet the objective
FDP_ROL.1[FIREWALL]	Contributes to meet the objective

OT.IDENTIFICATION

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

7.3.9 Random Numbers

OT.RND

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

7.3.10 Config Applet

OT.CONFIG-LIMIT

SFR	Rationale
FDP_ACC.1[CONFIG-APPLET]	Contributes to meet the objective.
FDP_ACF.1[CONFIG-APPLET]	Contributes to meet the objective.
FMT_MSA.1[CONFIG-APPLET]	Contributes to meet the objective.
FMT_MSA.3[CONFIG-APPLET]	Contributes to meet the objective.
FMT_SMF.1[CONFIG-APPLET]	Contributes to meet the objective.
FMT_SMR.1[CONFIG-APPLET]	Contributes to meet the objective.
FIA_UID.1[CONFIG-APPLET]	Contributes to meet the objective.
FIA_UAU.1[CONFIG-APPLET]	Contributes to meet the objective.

7.4 SFR Dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FAU_ARP.1	FAU_SAA.1 Potential violation analysis	see §7.3.3.1 of [8]
FAU_SAS.1[SCP]	No other components.	
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 §7.3.3.1 of [8]
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	see §7.3.3.1 of [8]
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	see §7.3.3.1 of [8]
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 §7.3.3.1 of [8]
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 §7.3.3.1 of [8]
FCS_RNG.1	No dependencies	

Requirements	CC Dependencies	Satisfied Dependencies
FDP_ACC.1[EXT-MEM]	FDP_ACF.1 Security attribute based access control	FDP_ACF.1[EXT-MEM]
FDP_ACC.1[SD]	FDP_ACF.1 Security attribute based access control	FDP_ACF.1[SD]
FDP_ACC.2[FIREWALL]	FDP_ACF.1 Security attribute based access control	see §7.3.3.1 of [8]
FDP_ACC.2[ADEL]	FDP_ACF.1 Security attribute based access control	see §7.3.3.1 of [8]
FDP_ACC.1[CONFIG-APPLET]	FDP_ACF.1 Security attribute based access control	see §7.3.3.1 of [8]
FDP_ACF.1[FIREWALL]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	see §7.3.3.1 of [8]
FDP_ACF.1[ADEL]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	see §7.3.3.1 of [8]
FDP_ACF.1[EXT-MEM]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1[EXT-MEM] FMT_MSA.3[EXT-MEM]
FDP_ACF.1[SD]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.1[SD] FMT_MSA.3[SD]
FDP_ACF.1[CONFIG-APPLET]	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialization	see §7.3.3.1 of [8]
FDP_IFC.1[JCVM]	FDP_IFF.1 Simple security attributes	see §7.3.3.1 of [8]
FDP_IFC.2[SC]	FDP_IFF.1 Simple security attributes	FDP_IFF.1[SC]
FDP_IFF.1[JCVM]	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation	see §7.3.3.1 of [8]

Requirements	CC Dependencies	Satisfied Dependencies
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_ITC.2[CCM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] FPT_TDC.1 Inter-TSF basic TSF data consistency	FDP_ACC.1[SD] FTP_ITC.1[SC]
FDP_RIP.1[OBJECTS]	No dependencies.	
FDP_RIP.1[ABORT]	No dependencies.	
FDP_RIP.1[APDU]	No dependencies.	
FDP_RIP.1[bArray]	No dependencies.	
FDP_RIP.1[KEYS]	No dependencies.	
FDP_RIP.1[TRANSIENT]	No dependencies.	
FDP_RIP.1[ADEL]	No dependencies.	
FDP_RIP.1[ODEL]	No dependencies.	
FDP_ROL.1[FIREWALL]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	see §7.3.3.1 of [8]
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	No dependencies.	
FDP_UIT.1[CCM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	FDP_ACC.1[SD] FTP_ITC.1[SC]
FIA_AFL.1[PIN]	FIA_UAU.1 Timing of authentication.	see AppNote in FIA_AFL.1[PIN]
FIA_ATD.1[AID]	No dependencies.	
FIA_UID.1[SC]	No dependencies.	
FIA_UID.1[CONFIG-APPLET]	No dependencies.	
FIA_UID.2[AID]	No dependencies.	

Requirements	CC Dependencies	Satisfied Dependencies
FIA_USB.1[AID]	FIA_ATD.1 User attribute definition	see §7.3.3.1 of [8]
FIA_UAU.1[SC]	FIA_UID.1 Timing of identification	FIA_UID.1[SC]
FIA_UAU.1[CONFIG-APPLET]	FIA_UID.1 Timing of identification	see §7.3.3.1 of [8]
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 §7.3.3.1 of [8]
FMT_MSA.1[JCVM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [8]
FMT_MSA.1[ADEL]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [8]
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[EXT-MEM]	[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[EXT-MEM] FMT_SMR.1 FMT_SMF.1[EXT-MEM]

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[CONFIG-APPLET]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [8]
FMT_MSA.2[FIREWALL-JCVM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [8]
FMT_MSA.3[FIREWALL]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [8]
FMT_MSA.3[JCVM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [8]
FMT_MSA.3[ADEL]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	see §7.3.3.1 of [8]
FMT_MSA.3[EXT-MEM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[EXT-MEM] FMT_SMR.1
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[CONFIG-APPLET]	FMT_MSA.1 Management of security attributes, FMT_SMR.1 Security roles	see §7.3.3.1 of [8]

Requirements	CC Dependencies	Satisfied Dependencies
FMT_MTD.1[JCRE]	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	see §7.3.3.1 of [8]
FMT_MTD.3[JCRE]	FMT_MTD.1 Management of TSF data	see §7.3.3.1 of [8]
FMT_SMF.1	No dependencies.	
FMT_SMF.1[ADEL]	No dependencies.	
FMT_SMF.1[EXT-MEM]	No dependencies.	
FMT_SMF.1[SD]	No dependencies.	
FMT_SMF.1[SC]	No dependencies.	
FMT_SMF.1[CONFIG-APPLET]	No dependencies.	
FMT_SMR.1	FIA_UID.1 Timing of identification	see §7.3.3.1 of [8]
FMT_SMR.1[INSTALLER]	FIA_UID.1 Timing of identification	see §7.3.3.1 of [8]
FMT_SMR.1[ADEL]	FIA_UID.1 Timing of identification	see §7.3.3.1 of [8]
FMT_SMR.1[SD]	FIA_UID.1 Timing of identification	FIA_UID.1[SC]
FMT_SMR.1[CONFIG-APPLET]	FIA_UID.1 Timing of identification.	see §7.3.3.1 of [8]
FPR_UNO.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_TDC.1	No dependencies.	
FPT_RCV.3[INSTALLER]	AGD_OPE.1 Operational user guidance	see §7.3.3.1 of [8]
FTP_ITC.1[SC]	No dependencies.	

Tab. 7.34: SFRs Dependencies

7.4.1 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 notrequire the identification of the "deletion manager" since it can be considered as part of the TSF.

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

The dependency FAU_SAA.1 of FAU_ARP.1 is unsupported. The dependency of FAU_ARP.1 on FAU_SAA.1 assumes that a "potential security violation" generates an audit event. On the contrary, the events listed in FAU_ARP.1 are self-contained (arithmetic exception, ill-formed bytecodes, access failure) and ask for a straightforward reaction of the TSFs on their occurrence at runtime. The JCVM or other components of the TOE detect these events during their usual working order. Thus, there is no mandatory audit recording in this ST.

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

7.5 Security Assurance Requirements Rationale

The selection of assurance components is based on the underlying PP [8]. The Security Target uses the augmentations from the PP, chooses EAL5 and adds the components AVA_VAN.5, ALC_DVS.2, ASE_TSS.2 and ALC_FLR.1.

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

The assurance level EAL5 is an elaborated pre-defined level of the CC, part 3 [4]. The assurance components in an EAL level are chosen in a way that they build a mutually supportive and complete set of components. The additional requirements chosen for augmentation do not add any dependencies, which are not already fulfilled for the corresponding requirements contained in EAL5. Therefore, the components AVA_VAN.5, ALC_DVS.2, ASE_TSS.2 and ALC_FLR.1 add additional assurance to EAL5, but the mutual support of the requirements is still guaranteed.

The assurance level EAL5 is an elaborated pre-defined level of the CC, part 3 [4]. The assurance components in an EAL level are chosen in a way that they build a mutually supportive and complete set of components. The additional requirements chosen for augmentation do not add any dependencies, which are not already fulfilled for the corresponding requirements contained in EAL5. Therefore, the components AVA_VAN.5, ALC_DVS.2, ASE_TSS.2 and ALC_FLR.1 add additional assurance to EAL5, 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 8.1 for list of all Security Functions. Each SF consists of components spread over several TOE modules to provide a security functionality and fulfill SFRs.

8.2 Security Functionality

Name	Title
SF.JCVM	Java Card Virtual Machine
SF.OPEN	Card Content Management
SF.CRYPTO	Cryptographic Functionality
SF.RNG	Random Number Generator
SF.DATA_STORAGE	Secure Data Storage
SF.EXT_MEM	External Memory
SF.OM	Java Object Management
SF.MM	Memory Management
SF.PIN	PIN Management
SF.PERS_MEM	Persistent Memory Management
SF.EDC	Error Detection Code API
SF.HW_EXC	Hardware Exception Handling
SF.PID	Platform Identification
SF.LIMIT_CONF	Limitation of the Configuration

Tab. 8.1: Overview of Security Functionality

SF.JCVM **Java Card Virtual Machine**

SF.JCVM provides the Java Card Virtual Machine including byte code interpretation and the Java Card Firewall according to the specifications [25, 26]. 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].

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 [25] 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 [25], and the Java Card API Specification [24] 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 [19] and fulfills FPT_RCV.3[INSTALLER] by inherent memory cleanup in case of aborted loading and installation.

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

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 [24] to generate random numbers according to FCS_RNG.1.

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]). All data stored by SF.DATA_STORAGE is CRC32 integrity protected to fulfill FDP_SDI.2, FAU_ARP.1, and FPT_FLS.1. The stored data is AES encrypted to fulfill FPR_UNO.1.

SF.EXT_MEM **External Memory**

SF.EXT_MEM provides mechanisms to access memory subsystems which are not directly addressable by the Java Card runtime environment (Java Card RE) on the Java Card platform such as MIFARE. The API is according to the Java Card API Specification [24] and implements the rules given in the EXTERNAL MEMORY access control SFP and thus fulfills FDP_ACC.1[EXT-MEM], FDP_ACF.1[EXT-MEM], FMT_MSA.1[EXT-MEM], FMT_MSA.3[EXT-MEM], and FMT_SMF.1[EXT-MEM].

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 [25] 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.

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 [25]. 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] and FDP_RIP.1[bArray] by clearing the APDU buffers for new incoming data and by clearing the bArray during application installation.

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 [24] and the GlobalPlatform Specification [23]. Thus, it fulfills FDP_SDI.2, FIA_AFL.1[PIN], and FPR_UNO.1.

SF.PERS_MEM **Persistent Memory Management**

SF.PERS_MEM provides atomic write operations and transaction management according to the Java Card Runtime Environment Specification [25]. 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.

SF.EDC **Error Detection Code API**

SF.EDC provides an Java API for user applications to perform high performing integrity checks based on a checksum on Java arrays [9]. The API throws a Java Exception in case the checksum in invalid. This supports FAU_ARP.1 and FPT_FLS.1.

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 and ensures the system goes to a secure state to fulfill FAU_ARP.1 and FPT_FLS.1.

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:

- EEPROM 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 EEPROM and ROM part of the TOE. This feature supports FAU_SAS.1.1[SCP] by using initialization data that is used for platform identification.

SF.LIMIT_CONF **Limitation of the Configuration**

SF.LIMIT_CONF limits the possibility to configure the TOE to authenticated users by using S.CONFIG-APPLET. This fulfills the SFR FMT_SMR.1[CONFIG-APPLET]. Reading of Config Items, creation of the Card Manager or modification of Config Items, ISD default keys or the EEPROM image is restricted by SF.LIMIT_CONF to the Config Applet which fulfills the SFRs FDP_ACC.1[CONFIG-APPLET], FDP_ACF.1[CONFIG-APPLET], and FMT_MSA.1[CONFIG-APPLET]. Subjects S.JCRE, S.SD and S.CONFIG-APPLET of the TOE are always allowed to use items of the configuration area to ensure that the TOE can make use of the configuration settings. This fulfills the SFR FDP_ACF.1[CONFIG-APPLET]. The Configuration area is initialized with initially restrictive default values which can be modified during the personalization. This fulfills the SFRs FMT_MSA.3[CONFIG-APPLET] and FMT_SMF.1[CONFIG-APPLET]. SF.LIMIT_CONF restricts the access to the functionality of the Config Applet to identified and authenticated users which fulfills the SFRs FIA_UID.1[CONFIG-APPLET] and FIA_UAU.1[CONFIG-APPLET].

8.3 Protection against Interference and Logical Tampering

The protection of JCOP 3 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. Once the Attack Counter reaches the maximum value, the TOE will terminate itself.

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, and RSA including protection against fault injection and random number generation.
- Critical values (like PINs) are compared timing-invariant. This prevents from side channel attacks.

A full list of software countermeasures is contained in ADV_ARC.

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.

JCOP 3 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 3 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 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

JCOP 3 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 [24, 25, 26]. The separation is achieved by implementation of the firewall which prevents Applets to access data belonging to a different Java Card context. Sharing information 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 Global platform [23]. The secure channel allows access to Applications only if the secret keys are known. Further protection implemented in JCOP 3 prevents brute force attacks to 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 JCOP 3 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 check for consistency. This prevents from attacks with manipulated parameters.

Further protection is achieved by using different buffers for APDUs in case more than one physical interface is supported. This prevents bypassing the state machine on one physical interface by the other interface.

9 Contents

10 Glossary

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 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 ([25], Glossary).

default applet The applet that is selected after a card reset ([25], §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 ([25], §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).

MC FW Microcontroller Firmware. Firmware for booting and low level functionality of the Micro Controller.

NVRAM Non-Volatile Random Access Memory, a type of memory that retains its contents when power is turned off.

object deletion The Java Card System version 2.2 and 3 Classic Edition mechanism ensures that any unreferenced persistent (transient) object owned by the current context is deleted. The associated memory space is recovered for reuse prior to the next card reset.

package A package is a namespace within the Java programming language that may contain classes and interfaces. A package defines either a user library, or one or more applet definitions. A package is divided in two sets of files: export files (which exclusively contain the public interface information for an entire package of classes, for external linking purposes; export files are not used directly in a Java Card virtual machine) and CAP files.

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.

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.

11 Acronyms

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.

AP Application Provider.

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.

APSD Application Provider Security Domain.

CAD Card Acceptance Device, or card reader. The device where the card is inserted, and which is used to communicate with the card.

Crypto Lib Crypto Library certified with the TOE hardware.

DFA Differential Fault Analysis, a side channel attack that induces faults into cryptographic implementations, to reveal their internal states..

DPA Differential Power Analysis is a form of side channel attack in which an attacker studies the power consumption of a cryptographic hardware device such as a smart card.

GP GlobalPlatform.

ICV Initial Chaining Vector.

ISD Issuer Security Domain.

JCOP 3 Java Card OpenPlatform Version 3 is the TOE (short for JCOP 3 EMV P60).

JCOP 3 OS Java Card OpenPlatform Operating System is the software stack without the MC FW.

OSP Organizational Security Policy.

PCD Proximity Coupling Device. The PCD is a contactless card reader device.

PICC Proximity Card. The PICC is a card with contactless capabilities.

RAM Random Access Memory, is a type of computer memory that can be accessed randomly.

RNG Random Number Generator.

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.

SIO An object of a class implementing a shareable interface.

SPD Security Problem Definition.

SSD Supplementary Security Domain.

VASD Verification Authority Security Domain.

12 Bibliography

[1] Bundesamt für Sicherheit in der Informationstechnik, Technische Richtlinie - Kryptographische Verfahren: Empfehlungen und Schlüssellängen, 09. Januar 2013, BSI-TR02102.

[2] Common Criteria for Information Technology Security Evaluation, Part 1 - Introduction and general model - Version 3.1 CCMB-2012-09-001, Revision 4, September 2012.

[3] Common Criteria for Information Technology Security Evaluation, Part 2 - Security functional components, Version 3.1 CCMB-2012-09-002, Revision 4, September 2012.

[4] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 4, September 2012, CCMB-2012-09-003.

[5] Common Methodology for Information Technology Security Evaluation - Evaluation Methodology, Version 3.1 CCMB-2012-09-004, Revision 4, September 2012.

[6] Crypto Library V3.1.x on P6021y VB, Security Target, Rev. 1.4, 2016-03-18.

[7] ISO/IEC 14443 Proximity Cards - Part 4: Transmission protocol - ISO/IEC 14443-2:2008.

[8] Java Card Protection Profile - Open Configuration, Version 3.0 (May 2012), Published by Oracle, Inc.

[9] JCOP 3 EMV P60, User Guidance and Administrator Manual, Rev. 3.1 - 2015-12-04.

[10] NXP Secure Smart Card Controller P6021y VB Security Target, Security Target, Rev. 0.93, 2016-01-25.

[11] Security IC Platform Protection Profile, registered and certified by Bundesamt fuer Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, Rev 1.0, 13 January 2014.

[12] SmartMX2 family P6021y VB Secure high-performance smart card controller, Objective Data Sheet, Rev. 3.1 - 2016-01-15.

[13] SmartMX2 P6021y_VB, Wafer and delivery specification, NXP Semiconductors, Rev. 3.1, 2016-01-21.

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

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

[16] Java Card System MISC 2.2 Configuration Protection Profile, Version 1.0b, August 2003.

[17] Java Card System Protection Profile Collection, Version 1.0b, August 2003.

[18] ISO 7816-3: Part 3: Cards with contacts - Electrical interface and transmission protocols , Nov 2006.

[19] GlobalPlatform Card Mapping Guidelines of Existing GP v2.1.1 Implementation on v2.2.1, January 2011.

[20] GlobalPlatform Card Specification 2.2.1, GPC_SPE_034, GlobalPlatform Inc., January 2011.

[21] GlobalPlatform ID Configuration, December 2011.

[22] Security Upgrade for Card Content Management Card Specification v2.2 - Amendment E v1.0, November 2011.

[23] Contactless Services, GlobalPlatform Card Specification v 2.2 - Amendment C v1.0.1, February 2012.

[24] Published by Oracle. Java Card 3 Platform, Application Programming Interface, Classic Edition, Version 3.0.4., September 2011.

[25] Published by Oracle. Java Card 3 Platform, Runtime Environment Specification, Classic Edition, Version 3.0.4, E18985-01., September 2011.

[26] Published by Oracle. Java Card 3 Platform, Virtual Machine Specification, Classic Edition, Version 3.0.4, E25256-01., September 2011.

[27] Bundesamt fuer Sicherheit in der Informationstechnik. Anwendungshinweise und Interpretationen zum Schema, AIS 20: Funktionalitaetsklassen und Evaluationsmethodologie fuer deterministische Zufallszahlengeneratoren, Version 2.1, 2.12.2011.

13 Legal information

13.1 Definitions

Draft – The document is a draft version only. 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 herein and shall have no liability for the consequences of use of such information.

13.2 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.

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 accepts 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.

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.

13.3 Licenses

ICs with DPA Countermeasures functionality

LICENSED DPA COUNTERMEASURES™

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.

13.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

13.5 Trademarks

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

MIFARE – is a trademark of NXP B.V.

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

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

Date of release: 2016-05-10

Document identifier: NSCIB-CC-15-67351