STSafe VJ200
Security Target

Version: E
Date: 2024-08-20
STMicroelectronics

Document history

Version	Date	Comment	Author
A	March 07, 2024	First release	STMicroelectronics
B	March 25, 2024	Implemented comments from SGS Action Items List v1.0	STMicroelectronics
C	April 08, 2024	Implemented comments from SGS Action Items List v2.0: A.ASE.2, A.ASE.5 and A.ASE.6	STMicroelectronics
D	August 01, 2024	Implemented comments from SGS Action Items List v4.0: A.ASE.1, A.ASE.7, updated references to AGDs	STMicroelectronics
E	August 20, 2024	Updated references to AGDs	STMicroelectronics

Distribution list

- Public

Contents

1 ST Introduction

This section provides information about the TOE, which enables a potential user of the TOE to determine, whether the TOE implements the functionality required by the user.

1.1 ST Reference

Title	STSafe VJ200 Security Target
Version	See Document History
Date	See Document History
Author	STMicroelectronics

Table 1 Security Target reference

1.2 TOE Reference

TOE Name	STSafe VJ200	
TOE Version	1.4.1	
TOE Identification	**IC**	IC Name: ST33K1M5A/ ST33K1M5M IC Maskset name: K4A0 Version: B02 Master product identification number: 0x0260/0x024B Firmware version: 3.1.4
	Java Card OS	OS_IDENTIFIER: 0x0000 OS_RELEASE_DATE: 0x4177 OS_RELEASE_LEVEL: 0x0007 OS_VERSION: 01040100
	Neslib	Neslib crypto library version: v6.8.2
	Store keeper	v4.1.2
	Weaver applet	1.9
TOE Type	Embedded secure element (eSE) with a Java Card System	

Table 2 TOE reference

1.2.1 Other certifications

The ST33K1M5 Secure IC has been already certified:

- IC name: ST33K1M5A/ ST33K1M5M
- CC certificate reference [CERT-IC]].

1.3 TOE Overview

STSafe VJ200 system-on-chip is an embedded secure element (eSE) with a Java Card System compliant with Java Card specifications version 3.0.5 with all the mandatory features, plus the following additions:

- support for the *int* type (including the *intx* package) and object deletion.
- support for Sensitive Results augmentation package.

The TOE can host and manage Java Card applets from different stakeholders (user, original equipment manufacturer (OEM), hardware integrator, service provider).

The TOE offers the following capabilities:

- Java Card functionality is included in the TOE; more specifically, the product includes a fully functional Java Card Virtual Machine [JCVM], a Java Card Runtime Environment [JCRE] and Java Card API [JCAPI] compliant to Java Card 3.0.5 specification.
- Support of Logical Secure Element (LSE), a proprietary ST feature based on ETSI TS 102 221 [TS102221].
- GlobalPlatform card specification v.2.3, including the Contactless Services according to Amendment C, the SCP03 protocol according to Amendment D, the Security Upgrade according to Amendment E, the elliptic curve-based secure channel protocol (SCP11a/SCP11b/SCP11c) according to Amendment F, and ELF upgrade according to Amendment H.
- Cryptographic functionality provided by NesLib crypto library.
- Physical Protection against physical tampering and leakage;
- Secure storage over the Weaver and OEM Secure Storage applications;
- “OS Upgrade” feature that allows Operational OS firmware update.

1.4 TOE Description

The TOE is a composition of a Java Card OS with an open configuration with the ST33K1M5 IC platform and includes Neslib and Storekeeper. It can also host and manage Java Card applets from different stakeholders (user, original equipment manufacturer (OEM), hardware integrator, service provider). The TOE also includes additional functionality to support the secure storage (OEM Secure Storage, Weaver and Storekeeper).

Figure 1 shows the high level architecture of the TOE. In pink the underlying platform, in blue the software components and interfaces and in green the out of scope applications

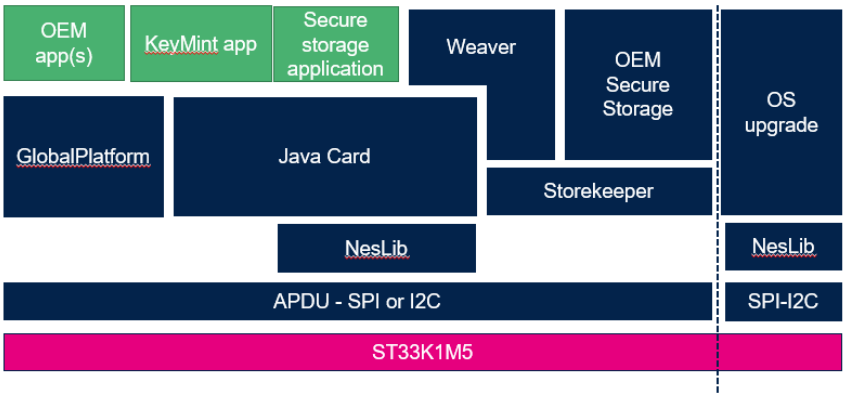

Figure 1 TOE Components

Hardware description

The hardware is the ST33K1M5A secure microcontroller, specific versions of the hardware parts are described in Section 1.2.

Note that the NESlib version is present twice, once for the Operational OS and once for the OS upgrade. The Storekeeper on the other side is present only in the Operational OS.

Physical / Communication Protocol

The device communicates over the SPI/I2C interface according to [GP_I2CSPI]. ISO7816 protocol is not supported (it is supported only in test mode).

The SPI/I2C library, implementing the Physical / communication protocol, is present twice (once for the Operational OS and once for the OS upgrade). The library is evaluated as part of the TOE evaluation.

Java Card

Java Card functionality is included in the TOE; more specifically, the product includes a fully functional Java Card Virtual Machine [JCVM], a Java Card Runtime Environment [JCRE] and Java Card API [JCAPI] compliant to Java Card 3.0.5 specification.

Logical Secure Element

To allow multiple TOE users to access different instances of the same applications, it has been introduced the concept of Logical Secure Element (LSE); an LSE is a group of package, applications and Security Domains including a specific Security Domain (called LSE-SD). To communicate to a specific LSE, a Logical Secure element Interface (LSI) needs to be established.

A specific LSE is privileged (LSE 0) and it is used to instantiate and configure the other LSE-SDs. The feature is based on ETSI TS 102 221 [TS102221] with adaptations for the specific TOE scenario.

Applications in different LSEs can have the same AID, the same GlobalPlatform services, etc.; before operating any command that may affect the applet selection, an LSI to the proper target LSE must be explicitly selected through the MANAGE LSI command [TS102221]. Only Security domains belonging to a specific LSEs can perform Card Content Management operations on the applications and packages belonging to the LSE; an exception and a difference with respect to ETSI specification is that a package belonging to the LSE 0 is visible in all the LSEs, allowing the sharing of the same applicative code among the various LSEs, while application instances are visible only in the corresponding LSE.

In the context of the TOE, this sharing allows the instantiation of multiple applications in different LSE from a single package loaded in the LSE 0, optimizing secure element memory resources.

Operational OS

The Operational OS offers secure storage capability and it is composed by several elements:

- The Weaver application is a Java Card based application with most of the functionalities developed in native; the OEM Secure Storage is fully native and it is the default application. Both applications use a specific wear-levelling library, named Storekeeper to improve reliability and performances.
- Supported communication protocols are SPI and I2C according to [GP_I2CSPI], co-existing and selected through a PIN modulation.
- The product also supports a Java Card Virtual machine and GlobalPlatform functionalities.

1.4.1 Physical Scope

The TOE is a composite TOE comprising hardware and software. The physical scope is defined as:

- the STMicroelectronics IC ST33K1M5A Security Integrated Circuit with dedicated software and embedded cryptographic library. Common Criteria certified by NSCIB with assurance level EAL6+ [CERT-IC]].
- An encrypted image of the STSafe VJ200 Operating system, including:
  - the Java Card Operating System version 3.0.5.
  - system applications with their configuration data.
- the associated guidance documentation in printed copy delivered in .pdf format delivered encrypted by e-mail:
  - Operational User Guidance [AGD_OPE]
  - Preparative Procedure [AGD_PRE]

The encrypted image of the STSafe VJ200 OS is transferred to STMicroelectronics engineering department encrypted via PGP by using shared repositories.

The TOE will be delivered by a trusted courier at the end of the phase B (see Section 1.4.5) in the format “Wafer level chip scale package” (WLCSP), with OS, personalization keys and data preloaded, in operational mode.

1.4.2 Logical Scope

The main security functions of the TOE are:

- **Firewall**. The TOE implements an applet firewall according to [JCRE].

- **Sensitive data confidentiality**. The TOE ensures that sensitive information is made unavailable after deletion.

- **Rollback protection**. The TOE implements atomicity and rollback mechanism for Java Card runtime environment [JCRE]

- **Secure Communications**. The TOE implements secure channel protocols according to [GP], chapter 10, GP Amendment D and GP Amendment F.

- **Card Management**. The TOE supports the GlobalPlatform Card Specifications v.2.3 and related amendments:
  - GlobalPlatform Amendment C – Contactless Services v1.3 (support of the "Cumulative Granted Memory" and "Cumulative Delete" sections)
  - GlobalPlatform Amendment D – Secure Channel Protocol SCP03 v1.1.1
  - GlobalPlatform Amendment E – Security Upgrade for Card Content Management v1.1
  - GlobalPlatform Amendment F – Secure Channel Protocol ‘11’ v1.2.1
  - GlobalPlatform Amendment H – Executable Load File Upgrade v1.1
  - GlobalPlatform Access Control v1.1
  - GlobalPlatform APDU communication over I²C/SPI based on the GlobalPlatform® “APDU Transport over I2C/SPI” specification v1.0
  - GlobalPlatform® SE Configuration v2.0

  A Card Manager (Issuer Security Domain) is present on the product.

- **Physical Protection**. The TOE provides means to protect itself against physical tampering and leakage.

- **Cryptographic Support**. The TOE provides key creation, key management, key deletion and cryptographic functionality. It provides the API in accordance to the Java Card API Specification [JCAPI].

- **PIN**. The TOE implements secure PIN compare functions and PIN integrity protection;

- **Firmware upgrade**. The product also contains an additional operating system, called OS Upgrade that is used to patch the operational OS firmware at OEM factory or on the field. The mechanism is based on AES encryption to ensure the confidentiality of software images. It can be configured to request user authentication to initiate the update procedure.

  It communicates over the SPI / I2C protocols according to [GP_I2CSPI] and relies on NESlib cryptographic services;

- **Secure Storage**. The TOE provides two different applications for accessing two separate secure memory areas. The OEM Secure Storage is the default application; it is operational only on logical channel 0, while Weaver application is operational only on logical channel 1.
  - The life cycle of the applications has the following steps:
  - FMS (flash manufacturing state)
  - CMS (Car manufacturing state)
  - UDS (user debug state)
  - URS (user release state)
  - TDS (temporary disabled state)

  The TOE is delivered to the car manufacturer in CMS state. Before the TOE is state is set to the final usage phase URS, the manufacturer replaces the pre-defined keys loaded into the TOE by the final keys to be used in normal operation, including the firmware loading keys and the OEM Secure Storage /Weaver binding keys.

.

1.4.3 TOE composition and identification

The TOE is the composition of a Java Card OS (including OEM Secure Storage and Weaver applications) over the Storekeeper library and the NESlib library, based on ST33K1M5 chip. The chip has been certified Common Criteria

As defined in [ADG_OPE], as the TOE is made by two Operating systems (Operational OS and OS upgrade), the two OSs have independent GET DATA commands that return the two OS versions.

As the Operational OS uses the NESlib and the Storekeeper, the Operational OS GET DATA command returns also NESlib and Storekeeper versions.

As the OS Upgrade uses the NESlib, the GET DATA command returns also NESlib version.

Both GET DATA commands allow also to identify the Hardware.

The TOE certification applies to the versions defined in Section 1.2

For further details, refer to [AGD_OPE].

1.4.4 Non-TOE Hardware/Software/Firmware

Here is a description of the non-TOE components and systems:

Component	Required	Description
Bytecode verifier	Mandatory	The bytecode verifier is a program that performs static checks on the bytecodes of the methods of a CAP file prior to the execution of the file on the card. Bytecode verification is a key component of security: applet isolation, for instance, depends on the file satisfying the properties a verifier checks to hold. A method of a CAP file that has been verified shall not contain, for instance, an instruction that allows forging a memory address or an instruction that makes improper use of a return address as if it were an object reference. In other words, bytecodes are verified to hold up to the intended use to which they are defined. Bytecode verification could be performed totally or partially dynamically. No standard procedure in that concern has yet been recognized. Furthermore, different approaches have been proposed for the implementation of bytecode verifiers, most notably data flow analysis, model checking and lightweight bytecode verification, this latter being an instance of what is known as proof carrying code. The actual set of checks performed by the verifier is implementation-dependent, but it is required that it should at least enforce all the “must clauses” imposed in [JCVM] on the bytecodes and the correctness of the CAP files’ format.

Table 3 Components of the environment

1.4.5 TOE Life Cycle

The composite product life cycle is decomposed into 4 phases. Each of these phases has the very same boundaries as those defined in the claimed protection profile [PP-JC].

The life cycle phases are summarized in Table 4.

Phase	Name	Description
A	JCS Development	This phase corresponds to the first two stages of the IC development. In this phase the OS and related applications are developed according to the Phase 1 of the ST Life cycle model as reported in Operational User Guidance.

B	JCS Storage, Pre-personalization testing	This phase corresponds to phase 5 of the IC development. In this phase the encrypted image is downloaded on the hardware by using the Flash Loader according to IC procedures. Product configuration is performed, including all the applications integration, the system applications configurations and static data configuration, according to Phase 5 of ST life cycle model reported in Operational User Guidance.
C	JCS Personalization	This phase corresponds to phase 6 of the IC development. In this phase, the devices are personalized with diversified credentials, according to Phase 6 of ST life cycle model reported in Operational User Guidance.
D	JCS Final usage	This phase corresponds to phase 7 of the IC development. Such a phase represents the life cycle state of the product on the field, according to Phase 7 of ST life cycle model reported in Operational User Guidance.

Table 4 TOE life cycle phases

2 Conformance claims

2.1 CC Conformance Claims

The TOE and ST claim conformance to the CC Version 3.1 revision 5 [CC31R5P2] [CC31R5P3].

The ST claim conformance to CC Part 2 extended and CC Part 3 conformant.

2.2 Package Claims

ST claims conformance to assurance package EAL5 augmented with ALC_DVS.2 and AVA_VAN.5.

2.3 PP Claims

ST claims demonstrable conformance to:

- Java Card Protection Profile - Open Configuration Version 3.1 PP-JC.

This ST is more restrictive than the PP [PP-JC] and Section 9.1 "Conformance Claim Rationale" provides a rational for it.

2.4 Conformance Rationale

This Security Target claims demonstrable conformance to the protection profile PP-JC.

The Security Assurance Requirements statement for the TOE in this Security Target includes all the requirements for the TOE from the PP-JC.

2.4.1 Security Problem Definition Statement

All sections of this Security Target regarding the Security Problem Definition, Security Objectives Statement and Security Requirements Statement for the TOE are taken over from PP-JC with the exception described in the following table.

SPD from [PP-JC]	Description
A.DELETION	Deletion of applets is in the scope of the evaluation. As discussed in Section2.4.2, O.CARD_MANAGEMENT is now Security Objective for the TOE.

Table 5 Security Problem Definition Statement

In addition, the Sensitive Result augmentation packages from [PP-JC] is in the scope. The SPD of this optional package is taken from Appendix 2 of the Java Card PP [PP-JC].

Additional threats and OSPs have been defined for Secure Storage and Firmware Upgrade in section 3.2.

2.4.2 Security Objectives Statement

The Security Objectives for the TOE and the Operational Environment of the Java Card implementation are the same as in the Java Card PP PP-JC with the following exceptions described in the following table.

SO from [PP-JC]	Description
O.ARRAY_VIEWS_CONFID	Functionality not implemented and therefore SO is not claimed.
O.ARRAY_VIEWS_INTEG	Functionality not implemented and therefore SO is not claimed.
OE.CARD-MANAGEMENT	Request on the Security IC component. Replaced by O.CARD-MANAGEMENT.

OE.SCP.RECOVERY	Request on the Security IC component. Replaced by O.SCP.RECOVERY.
OE.SCP.SUPPORT	Request on the Security IC component. Replaced by O.SCP.SUPPORT.
OE.SCP.IC	Request on the Security IC component. Replaced by O.SCP.IC.

Table 6 Java Card security objective statement

For the Java Card functionality, the Sensitive Result augmentation package from [PP-JC] is in the scope. Security Problem Definition of this optional package is taken from Appendix 2 of Java Card PP [PP-JC].

Additional Security Objectives have been defined for Secure Storage and Firmware Upgrade in section 4.2.

2.4.3 Security Functional Requirements Statement

2.4.3.1 Java Card

The Security Functional Requirements for the Java Card component are taken from the Java Card PP PP-JC without any modification. The Java Card OS also implements SFRs from the augmentation package Sensitive Results according to the Java Card PP Appendix 2 PP-JC.

Additional SFRs have been defined for Secure Storage and Firmware Upgrade in section 6.2.

3 Security Problem Definition

3.1 Java Card

The Security Problem Definition for the Java Card implementation is the same as the Security Problem Definition described in the Java Card PP PP-JC with the exceptions described in Section 2.4.1.

The TOE implements the following augmentation packages defined in Appendix 2 of the Java Card PP PP-JC: Sensitive Result. The Security Problem Definition for this augmentation package is taken from the Java Card PP PP-JC.

3.2 Secure Storage and Firmware Upgrade OS

3.2.1 Assets

Assets	Description
Data stored in memory	All data stored in the SFA secure memory and the Weaver secure memory, accessible to valid authenticated users through the SFA and Weaver applications interfaces.
Authentication keys	Keys used to authenticate the user for accessing the SFA and Weaver functionality, as well as for performing a software loading operation.
Software image	The software image running on the TOE, which can be updated by a valid authenticated user through the software loader functionality.

Table 7 Assets

3.2.2 Subjects

Subjects	Description
SFA user	User accessing the SFA secure memory through the SFA application.
Android user	User interacting with Android applications (e.g. Google Weaver and Google Keymint application).
Loader user	User accessing the Loader functionality to perform a software loading operation.

Table 8 Subjects

3.2.3 Threats

Threats	Description
T.DATA_DISCLOSE	An attacker performs unauthorised disclosure of data stored in the SFA secure memory and Weaver secure memory, or the disclosure of the authentication keys, by means of software/hardware attacks.
T.DATA_MODIFY	An attacker performs unauthorised modification of the data stored in the SFA secure memory and Weaver secure memory, or the disclosure of the authentication keys, by means of software/hardware attacks.
T.LOADER_MISUSE	An attacker performs unauthorised use of the software loader functionality to upload a modified or malicious software version.

Table 9 Threats

3.2.4 Organisational Security Policies

Policies	Description
P.KEY_PERSO	The default SFA binding keys and the Loader keys are updated by the car manufacturer during the manufacturing phase (CMS) of the TOE. After the final keys are set, the TOE state is changed to the final usage phase (URS).

Table 10 Organizational Security Policies

4 Security Objectives

4.1 Java Card

4.1.1 Security Objectives for the TOE

The Security Objectives for the TOE for the Java Card implementation are taken from the Security Objectives for the TOE described in the Java Card PP PP-JC with the exceptions described in Section 2.4.2.

The TOE implements the following augmentation packages defined in Appendix 2 of the Java Card PP PP-JC: Sensitive Result. The Security Objectives for the TOE added by the augmentation packages are taken from the Java Card PP PP-JC.

As discussed in Section 2.4.2, additional Security Objectives for the TOE have been added. Description is shown in the following table.

SO for the TOE	Description
O.CARD-MANAGEMENT	The card manager shall control the access to card management functions such as the installation, update or deletion of applets. It shall also implement the card issuer's policy on the card. The card manager is an application with specific rights, which is responsible for the administration of the smart card. This component will in practice be tightly connected with the TOE, which in turn shall very likely rely on the card manager for the effective enforcing of some of its security functions. Typically the card manager shall be in charge of the life cycle of the whole card, as well as that of the installed applications (applets). The card manager should prevent that card content management (loading, installation, deletion) is carried out, for instance, at invalid states of the card or by non-authorized actors. It shall also enforce security policies established by the card issuer.
O.SCP.IC	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 #.SCP: It is required that the IC is designed in accordance with a well-defined set of policies and Standards (likely specified in another protection profile), and will be tamper resistant to actually prevent an attacker from extracting or altering security data (like cryptographic keys) by using commonly employed techniques (physical probing and sophisticated analysis of the chip). This especially matters to the management (storage and operation) of cryptographic keys.
O.SCP.RECOVERY	If there is a loss of power, or if the smart card is withdrawn from the CAD while an operation is in progress, the SCP must allow the TOE to eventually complete the interrupted operation successfully, or recover to a consistent and secure state. This security objective for the environment refers to the security aspect #.SCP(1): The smart card platform must be secure with respect to the SFRs. Then after a power loss or sudden card removal prior to completion of some communication protocol, the SCP will allow the TOE on the next power up to either complete the interrupted operation or revert to a secure state.
O.SCP.SUPPORT	The SCP shall support the TSFs of the TOE. This security objective for the environment refers to the security aspects 2, 3, 4 and 5 of #.SCP: • (2) It does not allow the TSFs to be bypassed or altered and does not allow access to other low-level functions than those made available by the packages of the API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System. • (3) It provides secure low-level cryptographic processing to the Java Card System. • (4) It supports the needs for any update to a single persistent object or class field to be atomic, and possibly a low-level transaction mechanism. (5) It allows the Java Card System to store data in "persistent technology memory" or in volatile memory, depending on its needs (for instance, transient objects must not be stored in non-volatile memory). The memory model is structured and allows for low-level control accesses (segmentation fault detection).

Table 11 Additional Security Objectives for the TOE

4.1.2 Security Objectives for the Operational Environment

The Security Objectives for the Operational Environment for the Java Card implementation are taken from the Security Objectives for the Operational Environment described in the Java Card PP [PP-IC] with the exceptions discussed in Section 2.4.2.

4.1.3 Security Objectives Rationale

The Security Objectives Rationale for the Java Card implementation are taken from the Security Objectives Rationale section described in the Java Card PP PP-JC with the exceptions discussed in Sections 2.4.1 and 2.4.2.

The TOE implements the following augmentation packages defined in Appendix 2 of the Java Card PP PP-JC: Sensitive Result. The Security Objectives Rationale added by the augmentation packages are taken from the Java Card PP PP-JC.

4.2 **Secure Storage and Firmware Upgrade OS**

4.2.1 Security Objectives for the TOE

Objectives	**Description**
OT.DATA_PROTECTION	The TOE shall protect the integrity of the data stored in the secure memory from software/hardware attacks, to ensure that the stored data can only be modified by a valid authenticated user through the defined TOE interfaces.
OT.ACCESS_CONTROL	The TOE shall provide access control mechanisms to ensure only valid authenticated users can access the TOE functionality, i.e. SFA application, Weaver application and Loader functionality.

Table 12 Security Objectives for the TOE

4.2.2 Security Objectives for the Operational Environment

Objectives	Description
OE.KEY_PERSO	The operational environment shall ensure that when the TOE life cycle is in manufacturing state (CMS), and before it is set to release state (URS), all the default keys in the TOE are updated with final usage phase keys, including the SFA binding keys, FW authentication keys and the content loading keys.

Table 13 Security Objectives for the Operational Environment

4.2.3 Security Objectives rationale

The following table shows how the security objectives for the TOE cover the threats of the Secure Storage and Firmware Upgrade OS functionalities.

Threats / Security Objectives	**OT.DATA_PROTECTION**	**OT.ACCESS_CONTROL**
T.DATA_DISCLOSE		X
T.DATA_MODIFY	X	
T.LOADER_MISUSE		X

Table 14 Mapping of threats to TOE security objectives

The threats **T.DATA_DISCLOSE** and **T.LOADER_MISUSE** are covered by the security objective OT.ACCESS_CONTROL, which ensures that all the TOE functionality (SFA application, Weaver application and Loader functionality) can only be accessed by valid authenticated users.

The threat T.DATA_MODIFY is covered by the security objective OT.DATA_PROTECTION, which ensures the integrity of the data stored in secure memory so it can only be modified through the TOE interfaces and by valid authenticated users.

The following table shows how the security objectives for the operational environment cover the OSP.

OSPs / Security Objectives	**OE.KEY_PERSO**
P.KEY_PERSO	X

Table 15 Mapping of OSP to security objectives of the environment

The OSP P.KEY_PERSO is covered by the environment security objective OE.KEY_PERSO, which enforces that the default TOE keys are updated by the car manufacturer before the TOE is set to the final usage phase (URS).

5 Extended Component Definition

5.1 **Java Card**

Extended Component Definition from the Java Card PP-JC has been taken with no modification.

6 Security Functional Requirements

Reading notes:

- Selections having been made by the PP author are denoted as underlined text.
- Selections filled in by the ST author appear in square brackets with an indication that a selection is to be made [selection:] and are italicised.
- Assignments having been made by the PP author are denoted by showing as bold text.
- Assignments filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:] and are italicised.
- Refinements, if applicable, have been identified in bold and italicised text.
- Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier.

6.1 **Java Card**

6.1.1 COREG_LC SECURITY FUNCTIONAL REQUIREMENTS

The following table shows all the SFRs from Java Card PP PP-JC that do not require to perform any operation and therefore are an exact copy of the PP. SFRs containing operations that have not been performed by the Java Card PP PP-JC are addressed in the following sections.

Section	SFR
Firewall Policy	FDP_ACC.2/FIREWALL Complete access control
	FDP_ACF.1/FIREWALL Security attribute based access control
	FDP_IFC.1/JCVM Subset information flow control
	FDP_RIP.1/OBJECTS Subset residual information protection
	FMT_MSA.1/JCRE Management of security attributes
	FMT_MSA.1/JCVM Management of security attribute
	FMT_MSA.2/FIREWALL_JCVM Secure security attributes
	FMT_MSA.3/FIREWALL Static attribute initialisation
	FMT_MSA.3/JCVM Static attribute initialisation
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.1 Security roles
Application Programming Interface	FDP_RIP.1/ABORT Subset residual information protection
	FDP_RIP.1/APDU Subset residual information protection
	FDP_RIP.1/GlobalArray Subset residual information protection
	FDP_RIP.1/bArray Subset residual information protection
	FDP_RIP.1/KEYS Subset residual information protection
	FDP_RIP.1/TRANSIENT Subset residual information protection
	FDP_ROL.1/FIREWALL Basic rollback
Card Security Management	FPT_FLS.1 Failure with preservation of secure state
AID Management	FIA_UID.2/AID User identification before any action
	FMT_MTD.1/JCRE Management of TSF data
	FMT_MTD.3/JCRE Secure TSF data

Table 16 - SFRs from Java Card PP PP-JC

6.1.1.1 Firewall policy

6.1.1.1.1 FDP_IFF.1/JCVM Simple security attributes

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

Subjects	**Security attributes**
S.JCVM	Currently Active Context

Table 17 – JCVM information flow control Subjects and attributes

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:

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

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

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

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

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

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

6.1.1.2 Application Programming Interface

6.1.1.2.1 FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *see table below*] and specified cryptographic key sizes [assignment: *see table below*] that meet the following: [assignment: *see table below*].

Application Note:

- The keys can be generated and diversified in accordance with [JCAPI] specification in classes KeyBuilder and KeyPair (at least Session key generation).
- This component shall be instantiated according to the version of the Java Card API applying to the security target and the implemented algorithms ([JCAPI]).

Refer to Appendix 4 PP-JC to define the allowed/available key generation algorithms as per Java Card API specifications [JCAPI], in the table below the options supported by the this TOE are reported[1].

Iteration	**Cryptographic key generation algorithm**	**Cryptographic key size (in bits)**	**List of standards**
AES	AES key generation	128, 192, 256	FIPS PUB 197
TDES	TDES key generation	112, 168	FIPS PUB 46-3 (ANSI X3.92) FIPS PUB 81 GlobalPlatform v2.3
RSA key generation	RSA key pair (CRT and non-CRT)	2048 bits	ISO/IEC 9796-2 PKCS#1 v2.1
ECC	ECDH key generation ECDSA key generation	224, 256, 384, 512, 521 bits	[ANSI X9.62] [ISO 14888-3] [FIPS 186-4]

Table 18 - available key generation algorithms

6.1.1.2.2 FCS_CKM.4 Cryptographic key destruction

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

Application Note:

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

[1] Please note that, although the TOE supports other unlisted lengths (for legacy applications), some combinations algorithms/key sizes are considered not secure by SOG-IS (SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, Version 1.3, February 2023) and should not be used to handle sensitive data.

6.1.1.2.3 FCS_COP.1 Cryptographic operation

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

Application Note:

Refer to Appendix 4 to define the allowed/available algorithms as per Java Card API specifications [JCAPI] The ST Author should choose the algorithm implemented to perform crypto operations. For each algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

- The TOE shall provide a subset of cryptographic operations defined in [JCAPI] (see javacardx.crypto.Cipher and javacardx.security packages).
- This component shall be instantiated according to the version of the Java Card API applicable to the security target and the implemented algorithms [JCAPI].

Iteration	Cryptographic operation	Cryptographic algorithm	Supported key size	Standards
AES	Signature, signature's verification, encryption and decryption	AES with Modes CBC, GCM, CCM, and CMAC	128, 192 and 256 bits	FIPS PUB 197 SP800-38A (CBC) SP800-38B (CMAC) SP800-38C (CCM) SP800-38D (GCM)
DES	Signature, signature's verification, encryption and decryption	Single-key DES, 2-key and 3-key TDES in CBC mode	112 or 168 bits [2]	NIST SP 800-67 NIST SP 800-38A
RSA	RSA public key operation; RSA private key operation without CRT; RSA private key operation with CRT; EMSA PSS and PKCS1 signature scheme coding; RSA Key Encapsulation Method (KEM)	Rivest, Shamir & Adleman's	2048 bits	PKCS #1 v2.1
ECC	Diffie-Hellman (ECDH) key	Elliptic Curves Cryptography on	224, 256, 384, 512, 521 bits	FIPS 186-4

[2] Please note that, although the TOE supports other unlisted lengths (for legacy applications), some combinations algorithms/key sizes are considered not secure by SOG-IS (SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, Version 1.3, February 2023) and should not be used to handle sensitive data.

Iteration	Cryptographic operation	Cryptographic algorithm	Supported key size	Standards
	agreement computation Digital signature algorithm (ECDSA) generation and verification)	GF(p) on curves in Weierstrass form		ANSI X.9.62 section 7 NIST 800-56A
HASH	Hash	SHA-256 SHA-384 SHA-512 Protected SHA-1 Protected SHA-256 Protected SHA-384 Protected SHA-512	NA	FIPS 180-4
HMAC	Signature	SHA-1 SHA-256 SHA-384 SHA-512 Protected SHA-1 Protected SHA-256 Protected SHA-384 Protected SHA-512	NA	FIPS 198-1
DH	Key agreement	Diffie-Hellman	(2048, 224) (2048, 256)	ANSI X9.42
CTR-RBG	CTR-RBG	AES	128, 192 and 256 bits	NIST SP 800-90A FIPS 197

Table 19 – supported crypto algorithms

6.1.1.2.4 FCS_RNG.1 Random number generation

FCS_RNG.1.1 The TSF shall provide a [selection: *physical*] random number generator [selection: *PTG.2*] that implements: [assignment:

- *(PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.*
- *(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.*
- *(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.*

- *(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.*
- *(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered externally. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.*

].

FCS_RNG.1.2

The TSF shall provide random numbers that meet [assignment:

- *(PTG.2.6) Test procedure A does not distinguish the internal random numbers from output sequences of an ideal RNG.*
- *(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.*

].

Application Note:

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

6.1.1.3 Card Security Management

6.1.1.3.1 FAU_ARP.1 Security alarms

FAU_ARP.1.1

The TSF shall take **one of the following actions**:

- **throw an exception,**
- **lock the card session,**
- **reinitialize the Java Card System and its data,**
- [assignment: *none*]

upon detection of a potential security violation.

Refinement:

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

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

Application Note:

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

6.1.1.3.2 FDP_SDI.2/DATA Stored data integrity monitoring and action

FDP_SDI.2.1/DATA	The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *integrity protected data*].
FDP_SDI.2.2/DATA	Upon detection of a data integrity error, the TSF shall [assignment: *write a security error information persistently and mute the card*].

Application Note:

- *Although no such requirement is mandatory in the Java Card specification, at least an exception shall be raised upon integrity errors detection on cryptographic keys, PIN values and their associated security attributes. Even if all the objects cannot be monitored, cryptographic keys and PIN objects shall be considered with particular attention by ST authors as they play a key role in the overall security.*
- *It is also recommended to monitor integrity errors in the code of the native applications and Java Card applets.*
- *For integrity sensitive application, their data shall be monitored (D.APP_I_DATA): applications may need to protect information against unexpected modifications, and explicitly control whether a piece of information has been changed between two accesses. For example, maintaining the integrity of an electronic purse's balance is extremely important because this value represents real money. Its modification must*

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

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

6.1.1.3.3 FPR_UNO.1 Unobservability

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

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

6.1.1.3.4 FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret **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 • **the rules defined in [JCVM] specification,** • **the API tokens defined in the export files of reference implementation,** • [assignment: *none*] when interpreting the TSF data from another trusted IT product.

Application Note:

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

6.1.1.4 AID Management

6.1.1.4.1 FIA_ATD.1/AID User attribute definition

FIA_ATD.1.1/AID	The TSF shall maintain the following list of security attributes belonging to individual users: • CAP File AID, • Package AID, • Applet's version number, • Registered applet AID,

- Applet Selection Status,
- Associated Logical Secure Element

Refinement: "Individual users" stand for applets.

6.1.1.4.2 FIA_USB.1/AID User-subject binding

FIA_USB.1.1/AID	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **CAP file AID**.
FIA_USB.1.2/AID	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *for each loaded CAP file is associated an unique CAP file AID per Logical Secure Element*].
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 CAP file AID is unchangeable*].

Application Note:
The user is the applet and the subject is the S.CAP_FILE. The subject security attribute "Context" shall hold the user security attribute "CAP file AID".

6.1.2 InstG Security Functional Requirements

The following table shows all the SFRs from Java Card PP PP-JC that do not require to perform any operation and therefore are an exact copy of the PP. SFRs containing operations that have not been performed by the Java Card PP PP-JC are addressed in the following sections.

Section	SFR
InstG SFRs	FDP_ITC.2/Installer Import of user data with security attributes
	FMT_SMR.1/Installer Security roles
	FPT_FLS.1/Installer Failure with preservation of secure state

Table 20 – InstG SFRs

6.1.2.1 FPT_RCV.3/Installer Automated recovery without undue loss

FPT_RCV.3.1/Installer	When automated recovery from [assignment: *none*] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.
FPT_RCV.3.2/Installer	For [assignment: *interrupted deletion, interrupted load or interrupted install (except if the register method has already been invoked)*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.
FPT_RCV.3.3/Installer	The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is

restored without exceeding [assignment: *0%*] for loss of TSF data or objects under the control of the TSF.

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

Application Note:

FPT_RCV.3.1/Installer:

- This element is not within the scope of the Java Card specification, which only mandates the behavior of the Java Card System in good working order. Further details on the "maintenance mode" shall be provided in specific implementations. The following is an excerpt from [CC2], p298: In this maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only 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 [JCRE], §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 ([JCRE], 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 [PP0035]) 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.

6.1.3 ADELG Security Functional Requirements

The following table shows all the SFRs from Java Card PP PP-JC that do not require to perform any operation and therefore are an exact copy of the PP. SFRs containing operations that have not been performed by the Java Card PP PP-JC are addressed in the following sections.

Section	SFR
ADELG SFRs	FDP_ACC.2/ADEL Complete access control
	FDP_ACF.1/ADEL Security attribute based access control
	FDP_RIP.1/ADEL Subset residual information protection
	FMT_MSA.1/ADEL Management of security attributes
	FMT_MSA.3/ADEL Static attribute initialisation
	FMT_SMF.1/ADEL Specification of Management Functions
	FMT_SMR.1/ADEL Security roles
	FPT_FLS.1/ADEL Failure with preservation of secure state

Table 21 - ADELG SFRs

6.1.4 ODELG Security Functional Requirements

The following table shows all the SFRs from Java Card PP PP-JC that do not require to perform any operation and therefore are an exact copy of the PP. This section does not contain any SFRs with operations still to be performed.

Section	SFR
ODELG SFRs	FDP_RIP.1/ODEL Subset residual information protection
	FPT_FLS.1/ODEL Failure with preservation of secure state

Table 22 - ODELG SFRs

6.1.5 CarG Security Functional Requirements

The following table shows all the SFRs from Java Card PP PP-JC that do not require to perform any operation and therefore are an exact copy of the PP. SFRs containing operations that have not been performed by the Java Card PP PP-JC are addressed in the following sections.

Section	SFR
CarG SFRs	FDP_IFC.2/CM Complete information flow control
	FTP_ITC.1/CM Inter-TSF trusted channel

Table 23 - CarG SFRs

6.1.5.1 FCO_NRO.2/CM Enforced proof of origin

FCO_NRO.2.1/CM	The TSF shall enforce the generation of evidence of origin for transmitted **application CAP files** at all times.
FCO_NRO.2.2/CM [Editorially Refined]	The TSF shall be able to relate the **identity** of the originator of the information, and the **application CAP file,** of the information to which the evidence applies.

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

Application Note:

FCO_NRO.2.1/CM:

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

FCO_NRO.2.3/CM:

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

6.1.5.2 FDP_IFF.1/CM Simple security attributes

FDP_IFF.1.1/CM	The TSF shall enforce the **CAP FILE LOADING information flow control SFP** based on the following types of subject and information security attributes: [assignment: *Load file, DAP authenticated, OTA authenticated*].
FDP_IFF.1.2/CM	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: *the rules describing the communication protocol used by the CAD and the card for transmitting a new package as detailed in [GP] Section 9.3.9*].
FDP_IFF.1.3/CM	The TSF shall enforce the [assignment: *none*].
FDP_IFF.1.4/CM	The TSF shall explicitly authorise an information flow based on the following rules: [assignment: *none*].
FDP_IFF.1.5/CM	The TSF shall explicitly deny an information flow based on the following rules: • **The TOE fails to verify the integrity and authenticity evidences of the application CAP file.** • [assignment: *the rules describing the communication protocol used by the CAD and the card for transmitting a new package as detailed in [GP] Section 9.3.9*].

Application Note:

FDP_IFF.1.1/CM:

- The security attributes used to enforce the CAP FILE LOADING SFP are implementation dependent. More precisely, they depend on the communication protocol enforced between the CAD and the card. For instance, some of the attributes

that can be used are: (1) the keys used by the subjects to encrypt/decrypt their messages; (2) the number of pieces the application package has been split into in order to be sent to the card; (3) the ordinal of each piece in the decomposition of the package, etc. See for example Appendix D of [GP].

FDP_IFF.1.2/CM:

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

FDP_IFF.1.5/CM:

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

6.1.5.3 FDP_UIT.1/CM Data exchange integrity

FDP_UIT.1.1/CM	The TSF shall enforce the **CAP FILE LOADING information flow control SFP** to [selection: *receive*] user data in a manner protected from [selection: *modification, deletion, insertion, replay*] errors.
FDP_UIT.1.2/CM [Editorially Refined]	The TSF shall be able to determine on receipt of user data, whether **modification, deletion, insertion, replay of some of the pieces of the application sent by the CAD** has occurred.

Application Note:

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

6.1.5.4 FIA_UID.1/CM Timing of identification

FIA_UID.1.1/CM	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/CM	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Application Note:

The list of TSF-mediated actions is implementation-dependent, but CAP file installation requires the user to be identified. Here by user is meant the one(s) that in the Security Target shall be associated to the role(s) defined in the component FMT_SMR.1/CM.

6.1.5.5 FMT_MSA.1/CM Management of security attributes

FMT_MSA.1.1/CM	The TSF shall enforce the **CAP FILE LOADING information flow control SFP** to restrict the ability to [selection: *modify* [assignment: *no other operations*]] the security attributes [assignment: *key data, card life cycle state, secure configuration, default SELECTED configuration*] to [assignment: *card manager*].

6.1.5.6 FMT_MSA.3/CM Static attribute initialisation

FMT_MSA.3.1/CM	The TSF shall enforce the **CAP FILE LOADING information flow control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/CM	The TSF shall allow the [assignment: *card manager*] to specify alternative initial values to override the default values when an object or information is created.

6.1.5.7 FMT_SMF.1/CM Specification of Management Functions

FMT_SMF.1.1/CM	The TSF shall be capable of performing the following management functions: [assignment: *key data, card life cycle state, secure configuration, default SELECTED configuration*].

6.1.5.8 FMT_SMR.1/CM Security roles

FMT_SMR.1.1/CM	The TSF shall maintain the roles [assignment: *card manager*].
FMT_SMR.1.2/CM	The TSF shall be able to associate users with roles.

6.1.6 Additional Security Functional Requirements

6.1.6.1 FPT_TST.1 TSF Testing

FPT_TST.1.1	The TSF shall run a suite of self tests **at the conditions: during start-up and periodically during normal operation** to demonstrate the correct operation of the **TSF**.
FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of **TSF Data**.

FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of **parts of TSF (TSF executable code)**.

6.1.7 Optional package: Sensitive Results

6.1.7.1 FDP_SDI.2/RESULT Integrity_Sensitive_Result

FDP_SDI.2.1/RESULT	The TSF shall monitor user data stored in containers controlled by the TSF for **[assignment: integrity errors]** on all objects, based on the following attributes: **[assignment: sensitive API result stored in the javacardx.security.SensitiveResult class]**.
FDP_SDI.2.2/RESULT	Upon detection of a data integrity error, the TSF shall **[assignment: throw an exception]**.

Application Note:
This requirement applies in particular to the results stored by the javacardx.security.SensitiveResult class (if supported).

6.2 **Secure Storage and Firmware Upgrade OS**

6.2.1 FTP_ITC.1/SFA-Weaver Inter-TSF trusted channel

FTP_ITC.1.1/SFA-Weaver The TSF shall provide a communication channel between itself and ***the SFA user or the Android user*** 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/SFA-Weaver The TSF shall permit [selection: *another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC.1.3/SFA-Weaver The TSF shall initiate communication via the trusted channel for [assignment: *performing Read, Write or Erase operations through the SFA application or Weaver application*].

6.2.2 FDP_ACC.1/SFA-Weaver Subset access control – SFA and Weaver

FDP_ACC.1.1/SFA-Weaver The TSF shall enforce the [assignment: *SFA-Weaver SFP*] on [assignment:

(1) the subjects: SFA user, Android user,
(2) the objects: data in SFA secure memory, data in Weaver secure memory
(3) the operations: Read, Write and Erase

]

6.2.3 FDP_ACF.1/SFA-Weaver Security attribute based access control – SFA and Weaver

FDP_ACF.1.1/SFA-Weaver The TSF shall enforce the [assignment: *SFA-Weaver SFP*] to objects based on the following [assignment:

(1) the subjects: SFA user, Android user with security attributes "Authenticated",
(2) *the objects: the data in SFA secure memory with security attributes none, and the data in Weaver memory with security attributes none.*

]

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

(1) the SFA user with security attribute "Authenticated" set to "yes" can read, write or erase the data in the SFA secure memory through the SFA application.
(2) the Android user with security attribute "Authenticated" set to "yes" can read, write or erase the data in the

Android secure memory through the Android application.

]

Application note: the Read operation on the Weaver application requires the Android user to present the specific password of the memory slot that is to be read. The password for each slot is part of the data stored in the memory slot, which is written during a Write operation. Read is only allowed when the correct password is presented.

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

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

(1) *the SFA user with security attribute "Authenticated" set to "no" cannot access the data in secure memory*.
(2) *the Android user with security attribute "Authenticated" set to "no" cannot access the data in secure memory*.

]

6.2.4 FDP_ETC.1 Export of user data without security attributes

FDP_ETC.1.1 The TSF shall enforce the [assignment: *SFA-Weaver SFP*] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes.

6.2.5 FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1 The TSF shall enforce the [assignment: *SFA-Weaver SFP*] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

Application note: the password for each memory slot of the Weaver secure memory is stored as part of the slot data, so it is not considered a security attribute.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE [assignment: *none*].

6.2.6 FDP_SDI.2 Stored data integrity monitoring and action

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: *SFA secure memory integrity status, Weaver secure memory integrity status*].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [assignment:

(1) prohibit the use of the altered data
(2) inform the user about integrity error

]

6.2.7 FTP_ITC.1/Loader Inter-TSF trusted channel

FTP_ITC.1.1/Loader The TSF shall provide a communication channel between itself and ***the Loader user*** 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/Loader The TSF shall permit [selection: *another trusted IT product*] to initiate communication via the trusted channel.

FTP_ITC.1.3/Loader The TSF shall initiate communication via the trusted channel for [assignment: *performing a software loading operation*].

6.2.8 FDP_UCT.1 Basic data exchange confidentiality

FDP_UCT.1.1 The TSF shall enforce the [assignment: *Loader SFP*] to [selection: *receive*] user data in a manner protected from unauthorised disclosure.

6.2.9 FDP_UIT.1 Data exchange integrity

FDP_UIT.1.1 The TSF shall enforce the [assignment: *Loader SFP*] to [selection: *receive*] user data in a manner protected from [selection: *modification, deletion, insertion errors*].

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

6.2.10 FDP_ACC.1/Loader Subset access control – Loader

FDP_ACC.1.1/Loader The TSF shall enforce the [assignment: *Loader SFP*] on [assignment:

(1) the subjects: Loader user,
(2) the objects: software image data,
(3) the operation: performing a software loading

]

6.2.11 FDP_ACF.1/Loader Security attribute based access control – Loader

FDP_ACF.1.1/Loader The TSF shall enforce the [assignment: *Loader SFP*] to objects based on the following [assignment:

(1) the subjects: Loader user with security attributes "Authenticated",

(2) the objects: software image data in memory with security attributes none.

]

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

(1) the Loader user with security attribute "Authenticated" set to "yes" can perform a software loading operation.

]

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

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

(1) the Loader user with security attribute "Authenticated" set to "no" cannot perform a software loading operation.

]

7 Security Assurance Requirements

This Security Target claims conformance to EAL5 augmented with AVA_VAN.5 and ALC_DVS.2.

ADV_ARC is refined.

The requirements are summarised in the following table:

Assurance Class	Component	Component Title
ADV Development	ADV_ARC.1	Security architecture NOTE: This component has been refined as follows: *ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.* *Refinement:* *In particular, the TOE shall maintain the applet isolation without requiring more rules on applet verification than the* [GP-SGBA].
	ADV_FSP.5	Complete semi-formal functional specification with additional error information
	ADV_IMP.1	Implementation representation of the TSF
	ADV_INT.2	Well-structured internals
	ADV_TDS.4	Semiformal modular design
AGD: Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC_ Life-cycle support	ALC_CMC.4	Production support, acceptance procedures and automation
	ALC_CMS.5	Development tools CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.2	Sufficiency of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.2	Compliance with implementation standards
ASE: Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_TSS.1	TOE summary specification
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
ATE: Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.3	Testing: modular design
	ATE_FUN.1	Functional tests
	ATE_IND.2	Independent testing
AVA: Vulnerability assessment	AVA_VAN.5	Advanced methodical vulnerability analysis

Table 24 EAL5 requirements description extended with augmented with AVA_VAN.5 and ALC_DVS.2

8 TOE Summary Specification

8.1 Security Functionality

8.1.1 Java Card

SF.FIREWALL	The TOE implements an applet firewall according to [JCRE]. Each applet on the TOE must have been passed the Bytecode Verifier in order to ensure correct applet isolation. As an additional defensive security feature also a type check for API array parameters is performed. This TSF enforces the following SFRs: • FDP_ACC.2/FIREWALL Complete access control • FDP_ACF.1/FIREWALL Security attribute based access control • FDP_IFC.1/JCVM Subset information flow control • FDP_IFF.1/JCVM Simple security attributes • FMT_MSA.1/JCRE Management of security attributes • FMT_MSA.2/FIREWALL_JCVM Secure security attributes • FMT_MSA.3/FIREWALL Static attribute initialisation • FMT_MSA.3/JCVM Static attribute initialization • FMT_SMR.1 Security roles • FDP_ROL.1/FIREWALL Basic rollback • FMT_MSA.1/JCVM Management of security attributes • FMT_MTD.1/JCRE Management of TSF data • FMT_MTD.3/JCRE Secure TSF data • FMT_SMF.1 Specification of Management Functions
SF.RIP	The TOE ensures that sensitive information is made unavailable after deletion. This will be done by overwriting keys, APDU buffer and transient objects with zeros or random values. Applications and persistent objects will be marked as deleted. If the deleted resource is reused by a new object creation, the previous content will be set to a random value. This TSF enforces the following SFRs: • FDP_RIP.1/bArray Subset residual information protection • FDP_RIP.1/APDU Subset residual information protection • FDP_RIP.1/KEYS Subset residual information protection • FDP_RIP.1/TRANSIENT Subset residual information protection • FDP_RIP.1/ADEL Subset residual information protection • FDP_RIP.1/ODEL Subset residual information protection • FDP_RIP.1/ABORT Subset residual information protection • FDP_RIP.1/OBJECTS Subset residual information protection • FDP_RIP.1/GlobalArray Subset residual information protection
SF.Rollback	The TOE implements atomicity and rollback mechanism for Java Card runtime environment [JCRE] and GlobalPlatform management functions (see [GP]). The TOE also ensures that objects created during an aborted transaction are made unavailable. This TSF enforces the following SFRs: • FPT_RCV.3/Installer Automated recovery without undue loss • FDP_ROL.1/FIREWALL Basic rollback • FDP_RIP.1/ABORT Subset residual information protection
SF.SCP	The TOE implements secure channel protocols according to [GP], chapter 10. The following protocols are supported: • SCP02 according to [GP-E]. • SCP03 according to . • SCP03t according to • SCP11 according to [GP-F]; • Contactless Services [GP-C]; • Executable Load File Upgrade [GP-H].

	The SCP uses as the basic cryptographic primitives the security hardened symmetric cryptographic library which is CC certified together with the underlying platform. This TSF enforces the following SFRs: • FDP_UIT.1/CM Data exchange integrity • FTP_ITC.1/CM Inter-TSF trusted channel • FCO_NRO.2/CM Enforced proof of origin • FDP_IFC.2/CM Complete information flow control • FDP_IFF.1/CM Simple security attributes • FMT_MSA.1/CM Management of security attributes • FMT_MSA.3/CM Static attribute initialisation • FMT_SMF.1/CM Specification of Management Functions • FIA_UID.1/CM Timing of identification • FMT_SMR.1/CM Security roles • FCS_COP.1 Cryptographic operation
SF.CM	The TOE implements an access control policy for GlobalPlatform card management functions according to [GP] , C [GP-C], D and E [GP-E]. In addition to the GP specification, the Java Card Runtime Environment specification [JCRE] is followed to support for application loading, installation, and deletion. AID management is provided by SF.CM according to the GlobalPlatform Specification [GP], the Java Card Runtime Environment Specification [JCRE], and the Java Card API Specification [JCAPI]. This TSF enforces the following SFRs: • FMT_MSA.1/CM Management of security attributes • FMT_MSA.3/CM Static attribute initialisation • FMT_SMF.1/CM Specification of Management Functions • FMT_SMR.1/CM Security roles • FPT_TDC.1 Inter-TSF basic TSF data consistency • FIA_ATD.1/AID User attribute definition • FIA_UID.2/AID User identification before any action • FIA_USB.1/AID User-subject binding • FDP_ITC.2/Installer Import of user data with security attributes • FMT_SMR.1/Installer Security roles • FPT_RCV.3/Installer Automated recovery without undue loss • FPT_FLS.1/Installer Failure with preservation of secure state • FDP_ACC.2/ADEL Complete access control • FDP_ACF.1/ADEL Security attribute based access control • FDP_RIP.1/ADEL Subset residual information protection • FMT_MSA.1/ADEL Management of security attributes • FMT_MSA.3/ADEL Static attribute initialisation • FMT_SMR.1/ADEL Security roles • FPT_FLS.1/ADEL Failure with preservation of secure state • FMT_SMF.1/ADEL Specification of Management Functions • FPT_FLS.1/ODEL Failure with preservation of secure state
SF.Physical	The TOE provides means to protect SFRs against physical tampering and leakage. The TOE uses mainly the physical security measures of the underlying hardware platform. Security mechanisms involved in this protection are: • Memories scrambiling and encryption • Protection of NVM sectors • Memory Protection Unit (MPU) • Library Protection Unit (LPU) This TSF enforces the following SFRs: • FAU_ARP.1 Security alarms • FDP_SDI.2/DATA Stored data integrity monitoring and action • FPT_TST.1 TSF testing • FPT_FLS.1 Failure with preservation of secure state
SF.CRYPTO	The TOE provides key creation, key management, key deletion and cryptographic functionality. It provides the API in accordance to the Java Card API Specification [JCAPI].

	The cryptographic API uses as the basic cryptographic implementation the security hardened cryptographic library which is CC certified together with the underlying platform. The integrity of the cryptographic assets is monitored. In addition, key destructions and residual information purging is implemented. SF.CRYPTO provides secure random number generation and makes this functionality available through an API according to the Java Card API Specification [JCAPI]. This TSF enforces the following SFRs: • FCS_CKM.1 Cryptographic key generation • FCS_CKM.4 Cryptographic key destruction • FCS_COP.1 Cryptographic operation • FPR_UNO.1 Unobservability • FDP_SDI.2/DATA Stored data integrity monitoring and action • FCS_RNG.1 Random number generation • FCS_COP.1/DRBG Cryptographic operation
SF.PIN	The TOE implements secure PIN compare functions and integrity protection of the PIN. This TSF enforces the following SFRs: • FPR_UNO.1 Unobservability • FDP_SDI.2/DATA Stored data integrity monitoring and action

Table 25 – JC Security Functionalities

8.1.2 Secure Storage and Firmware Upgrade OS

SF.AUTH-WEAVER	For Android User, SFA User and Loader User, authentication is achieved through the establishment of the secure channel, which is a proprietary implementation based on and very similar to the SCP03 protocol. The secure channel can be established using 3 different key sets, each corresponding to the 3 different TOE users: SFA user, Android user and Loader user. The TOE enforces access control by verifying that specific actions are authorized by the proper credential. Establishment of the secure channel with either of the 3 key sets provides access to the SFA application, Android applications and the Loader functionality for each corresponding user. Operational OS security operations are authorized to SFA and Android Users, while Firmware Upgrade OS security operations are authorized to Loader User. This functionality meets the SFR related to user authentication and access control: Operational OS functionalities authentication and access control: • FTP_ITC.1/SFA-Weaver • FDP_ACC.1/SFA-Weaver • FDP_ACF.1/SFA-Weaver • FDP_ETC.1 • FDP_ITC.1 • FDP_UCT.1 • FDP_UIT.1 Firmware Upgrade OS functionalities authentication and access control: • FTP_ITC.1/Loader • FDP_UCT.1 • FDP_ACC.1/Loader • FDP_ACF.1/Loader • FDP_UIT.1
SF.STORE-DATA-PROTECTION	The TOE provides secure storage based on flash memory being managed by tearing safe transfer and wear-levelling mechanisms. This functionality meets the SFR related to integrity protection of stored data, keys and firmware images: • FDP_SDI.2

Table 26 - Secure Storage and Firmware Upgrade OS Security Functionalities

9 Rationales

9.1 Conformance Claim Rationale

The statement of security functional requirements copies most SFRs as defined in the PP [PP-JC], apart from FIA_ATD.1/AID.
To the SFR FIA_ATD.1/AID from the PP an extra security attribute belonging to individual users is added, the Associated Logical Secure Element, thus making more restrictive than the SFR from PP [PP-JC].

9.2 Security Requirements Rationale

9.2.1 Java Card

Objective	Rationale
O.SID	Subjects' identity is AID-based (applets, packages and CAP files), and is met by the following SFRs: FDP_ITC.2/Installer, FIA_ATD.1/AID, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_MSA.1/ ADEL, FMT_MSA.1/CM, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.3/CM, FMT_SMF.1/CM, FMT_SMF.1/ADEL, FMT_SMF.1/ADEL, FMT_MTD.1/JCRE and FMT_MTD.3/JCRE. Installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities (FIA_UID.2/AID, FIA_USB.1/AID).
O.FIREWALL	This objective is met by the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL, the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) and the functional requirement FDP_ITC.2/Installer. The functional requirements of the class FMT (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1/Installer, FMT_SMR.1, FMT_SMF.1, FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_SMF.1/CM, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMR.1/CM, FMT_MSA.2/FIREWALL_JCVM, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM) also indirectly contribute to meet this objective.
O.GLOBAL_ARRAYS_CONFID	Only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer, the global byte array input parameter (bArray) to an applet's install method and the global arrays created by the JCSystem.makeGlobalArray(…) method. The clearing requirement of these arrays is met by (FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray and FDP_RIP.1/bArray respectively). The JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.
O.GLOBAL_ARRAYS_INTEG	This objective is met by the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM), which prevents an application from keeping a pointer to the APDU buffer of the card, to the global byte array of the applet's install method or to the global arrays created by the JCSystem.makeGlobalArray(…) method. Such a pointer could be used to access and modify it when the buffer is being used by another application.
O.NATIVE	This security objective is covered by FDP_ACF.1/FIREWALL: the only means to execute native code is the invocation of a Java Card API method. This objective mainly relies on the environmental objective OE.CAP_FILE, which uphold the assumption A.CAP_FILE.
O.OPERATE	The TOE is protected in various ways against applets' actions (FPT_TDC.1), the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL, and is able to detect and block various failures or security violations during usual working (FPT_FLS.1/ADEL, FPT_FLS.1, FPT_FLS.1/ODEL, FPT_FLS.1/Installer, FAU_ARP.1). Its security-critical parts and procedures are also protected: safe recovery from failure is ensured (FPT_RCV.3/Installer), applets' installation may be cleanly aborted (FDP_ROL.1/FIREWALL), communication with external users and their internal subjects is well-controlled (FDP_ITC.2/Installer, FIA_ATD.1/AID, FIA_USB.1/AID) to prevent alteration of TSF data (also protected by components of the FPT class).

Objective	Rationale
	Almost every objective and/or functional requirement indirectly contributes to this one too. Application note: Startup of the TOE (TSF-testing) can be covered by FPT_TST.1. This SFR component is not mandatory in [JCRE], but appears in most of security requirements documents for masked applications. Testing could also occur randomly. Self-tests may become mandatory in order to comply with FIPS certification [FIPS 140-2].
O.REALLOCATION	This security objective is satisfied by the following SFRs: FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ADEL, which imposes that the contents of the re-allocated block shall always be cleared before delivering the block.
O.RESOURCES	The TSFs detects stack/memory overflows during execution of applications (FAU_ARP.1, FPT_FLS.1/ADEL, FPT_FLS.1, FPT_FLS.1/ODEL, FPT_FLS.1/Installer). Failed installations are not to create memory leaks (FDP_ROL.1/FIREWALL, FPT_RCV.3/Installer) as well. Memory management is controlled by the TSF (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1/Installer, FMT_SMR.1, FMT_SMF.1 FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_SMF.1/CM and FMT_SMR.1/CM). Additionally, if the TOE provides JCRMI functionality, memory management is controlled by the TSF FMT_SMR.1/JCRMI, and FMT_SMF.1/JCRMI.
O.ALARM	This security objective is met by FPT_FLS.1/Installer, FPT_FLS.1, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL which guarantee that a secure state is preserved by the TSF when failures occur, and FAU_ARP.1 which defines TSF reaction upon detection of a potential security violation.
O.CIPHER	This security objective is directly covered by FCS_CKM.1, FCS_CKM.4 and FCS_COP.1. The SFR FPR_UNO.1 contributes in covering this security objective and controls the observation of the cryptographic operations which may be used to disclose the keys.
O.RNG	This security objective is directly covered by FCS_RNG.1 and FCS_COP.1/DRBG which ensure the cryptographic quality of random number generation.
O.KEY-MNGT	This relies on the same security functional requirements as O.CIPHER, plus FDP_RIP.1 and FDP_SDI.2/DATA as well. Precisely it is met by the following components: FCS_CKM.1, FCS_CKM.4, FCS_COP.1, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL and FDP_RIP.1/TRANSIENT.
O.PIN-MNGT	This security objective is ensured by FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FPR_UNO.1, FDP_ROL.1/FIREWALL and FDP_SDI.2/DATA security functional requirements. The TSFs behind these are implemented by API classes. The firewall security functions FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL shall protect the access to private and internal data of the objects.
O.TRANSACTION	Directly met by FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT, FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT and FDP_RIP.1/OBJECTS (more precisely, by the element FDP_RIP.1.1/ABORT).
O.OBJ-DELETION	This security objective specifies that deletion of objects is secure. The security objective is met by the security functional requirements FDP_RIP.1/ODEL and FPT_FLS.1/ODEL.
O.DELETION	This security objective specifies that applet and CAP file deletion must be secure. The non-introduction of security holes is ensured by the ADEL access control policy (FDP_ACC.2/ADEL, FDP_ACF.1/ADEL). The integrity and confidentiality of data that does not belong to the deleted applet or CAP file is a by-product of this policy as well. Non-accessibility of deleted data is met by FDP_RIP.1/ADEL and the TSFs are protected against possible failures of the deletion procedures (FPT_FLS.1/ADEL, FPT_RCV.3/Installer). The security functional requirements of the class FMT (FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL) included in the group ADELG also contribute to meet this objective.

Objective	Rationale
O.LOAD	This security objective specifies that the loading of a CAP file into the card must be secure. Evidence of the origin of the CAP file is enforced (FCO_NRO.2/CM) and the integrity of the corresponding data is under the control of the CAP FILE LOADING information flow policy (FDP_IFC.2/CM, FDP_IFF.1/CM) and FDP_UIT.1/CM. Appropriate identification (FIA_UID.1/CM) and transmission mechanisms are also enforced (FTP_ITC.1/CM).
O.INSTALL	This security objective specifies that installation of applets must be secure. Security attributes of installed data are under the control of the FIREWALL access control policy (FDP_ITC.2/Installer), and the TSFs are protected against possible failures of the installer (FPT_FLS.1/Installer, FPT_RCV.3/Installer).
O.CARD-MANAGEMENT	This objective is fulfilled by the following set of SFR: FDP_ACC.2/ADEL and FDP_ACF.1/ADEL contribute to meet the objective by the ADEL access control policy which ensures the non-introduction of security holes. FDP_RIP.1/ADEL ensures the non-accessibility of deleted data. FMT_MSA.1/ADEL and FMT_MSA.3/ADEL enforce the ADEL access control SFP. FMT_SMR.1/ADEL maintains the role applet deletion manager. FPT_RCV.3/Installer protects the TSFs against possible failures of the deletion procedures. FPT_FLS.1/Installer protects the TSFs against possible failures of the installer. FPT_FLS.1/ADEL protects the TSFs against possible failures of the deletion procedures. FDP_UIT.1/CM enforces the Secure Channel Protocol information flow control policy and the Security Domain access control policy which controls the integrity of the corresponding data. FDP_IFF.1/CM ensures the access control policy for the loaded data (as packages). The FCO_NRO.2/CM ensures the origin of the load file. It verifies the identity of the origin of the load file before start the loading FDP_IFC.2/CM ensures that loading commands are issued in the Secure Channel session. FDP_ROL.1/Firewall ensures that the card management operations are cleaned aborted FDP_ITC.2/Installer enforces the Firewall access control policy and flow control policy when importing card management data. FPT_FLS.1/ODEL ensures the preservation of secure state when failures occur. FMT_MSA.1/CM ensures the management of the security attributes to the card manager, for the modification of the defined security attributes. FMT_MSA.3/CM ensures that the security attributes can only be changed by the card manager. FMT_SMF.1/CM allows only the card manger to modify the security attributes of the management functions. The security role is specified in the FMT_SMR.1/CM. FTP_ITC.1/CM ensures the trusted Channel Communications. FPR_UNO.1 ensures the un-observability of the CM key when imported.. FPT_TST.1 ensures the correct operation of the card management functions as it tests the integrity of the TSF functions during initial start-up.
O.SCP.RECOVERY	FPT_RCV.3/Installer is used to assist the TOE to recover in the event of a power failure. The component FAU_ARP.1 is used to ensure the reinitialization of the Java Card System and its data after card tearing and power failure. The component FPT_FLS.1 is used to preserve a secure state after failure. O.SCP.SUPPORT
O.SCP.SUPPORT	All crypto SFRs supports this objective as they provide the functionality to the Java Card and Global Platform (FCS_CKM.1, FCS_CKM.4, FCS_COP.1) All the FSRs related to the Firewall contribute to the realization of the objective (FDP_ROL.1/FIREWALL).
O.SCP.IC	This objective is met by providing physical protection (FPR_UNO.1, FPT_EMS.1 and FPT_PHP.3) and taking action upon security violation (FAU_ARP.1).

Table 27 – Rationale Security Objectives/SFRs

9.2.2 Secure Storage and Firmware Upgrade OS

SFR / TOE objectives	OT.DATA_PROTECTION	OT.ACCESS_CONTROL
FTP_ITC.1/SFA-Weaver		X
FDP_ACC.1/SFA-Weaver		X
FDP_ACF.1/SFA-Weaver		X
FDP_ETC.1		X
FDP_ITC.1		X
FDP_SDI.2	X	
FTP_ITC.1/Loader		X
FDP_UCT.1	X	X
FDP_UIT.1	X	X
FDP_ACC.1/Loader		X
FDP_ACF.1/Loader		X

Table 28 - Rationale SFRs/Security Objectives

9.3 Dependency Rationale

9.3.1 Java Card

Requirement	Dependency	Satisfied by
FDP_ACC.2/FIREWALL	(FDP_ACF.1)	FDP_ACF.1/FIREWALL
FDP_ACF.1/FIREWALL	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2/FIREWALL, FMT_MSA.3/FIREWALL
FDP_IFC.1/JCVM	(FDP_IFF.1)	FDP_IFF.1/JCVM
FDP_IFF.1/JCVM	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1/JCVM, FMT_MSA.3/JCVM
FDP_RIP.1/OBJECTS	No Dependencies	
FMT_MSA.1/JCRE	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL, FMT_SMR.1
FMT_MSA.1/JCVM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL, FDP_IFC.1/JCVM, FMT_SMF.1, FMT_SMR.1
FMT_MSA.2/FIREWALL_JCVM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_MSA.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL, FDP_IFC.1/JCVM, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_SMR.1
FMT_MSA.3/FIREWALL	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_SMR.1
FMT_MSA.3/JCVM	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCVM, FMT_SMR.1
FMT_SMF.1	No Dependencies	
FMT_SMR.1	(FIA_UID.1)	FIA_UID.2/AID
FCS_CKM.1	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1, FCS_CKM.4
FCS_CKM.4	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2)	FCS_CKM.1
FCS_COP.1	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1, FCS_CKM.4
FCS_RNG.1	No Dependencies	
FDP_RIP.1/ABORT	No Dependencies	
FDP_RIP.1/APDU	No Dependencies	

FDP_RIP.1/bArray	No Dependencies	
FDP_RIP.1/GlobalArray	No Dependencies	
FDP_RIP.1/KEYS	No Dependencies	
FDP_RIP.1/TRANSIENT	No Dependencies	
FDP_ROL.1/FIREWALL	(FDP_ACC.1 or FDP_IFC.1)	FDP_ACC.2/FIREWALL, FDP_IFC.1/JCVM
FAU_ARP.1	(FAU_SAA.1)	
FDP_SDI.2/DATA	No Dependencies	
FPR_UNO.1	No Dependencies	
FPT_FLS.1	No Dependencies	
FPT_TDC.1	No Dependencies	
FIA_ATD.1/AID	No Dependencies	
FIA_UID.2/AID	No Dependencies	
FIA_USB.1/AID	(FIA_ATD.1)	FIA_ATD.1/AID
FMT_MTD.1/JCRE	(FMT_SMF.1) and (FMT_SMR.1)	FMT_SMF.1, FMT_SMR.1
FMT_MTD.3/JCRE	(FMT_MTD.1)	FMT_MTD.1/JCRE
FDP_ITC.2/Installer	(FDP_ACC.1 or FDP_IFC.1) and (FPT_TDC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.2/CM, FTP_ITC.1/CM, FPT_TDC.1
FMT_SMR.1/Installer	(FIA_UID.1)	
FPT_FLS.1/Installer	No Dependencies	
FPT_RCV.3/Installer	(AGD_OPE.1)	AGD_OPE.1
FDP_ACC.2/ADEL	(FDP_ACF.1)	FDP_ACF.1/ADEL
FDP_ACF.1/ADEL	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2/ADEL, FMT_MSA.3/ADEL
FDP_RIP.1/ADEL	No Dependencies	
FMT_MSA.1/ADEL	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/ADEL, FMT_SMF.1/ADEL, FMT_SMR.1/ADEL
FMT_MSA.3/ADEL	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/ADEL, FMT_SMR.1/ADEL
FMT_SMF.1/ADEL	No Dependencies	
FMT_SMR.1/ADEL	(FIA_UID.1)	
FPT_FLS.1/ADEL	No Dependencies	
FDP_RIP.1/ODEL	No Dependencies	
FPT_FLS.1/ODEL	No Dependencies	
FCO_NRO.2/CM	(FIA_UID.1)	FIA_UID.1/CM
FDP_IFC.2/CM	(FDP_IFF.1)	FDP_IFF.1/CM
FDP_IFF.1/CM	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.2/CM, FMT_MSA.3/CM
FDP_UIT.1/CM	(FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.2/CM, FTP_ITC.1/CM
FIA_UID.1/CM	No Dependencies	
FMT_MSA.1/CM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_IFC.2/CM, FMT_SMF.1/CM, FMT_SMR.1/CM
FMT_MSA.3/CM	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/CM, FMT_SMR.1/CM
FMT_SMF.1/CM	No Dependencies	
FMT_SMR.1/CM	(FIA_UID.1)	FIA_UID.1/CM
FTP_ITC.1/CM	No Dependencies	

Table 29 - JC Dependency Rationale

Rationale for the exclusion of dependencies:

- The dependency FIA_UID.1 of FMT_SMR.1/Installer is discarded. The Java Card PP [PP-JC] does not require the identification of the "installer" since it can be considered as part of the TSF.
- The dependency FIA_UID.1 of FMT_SMR.1/ADEL is discarded. The Java Card PP [PP-JC] does not require the identification of the "deletion manager" since it can be considered as part of the TSF.
- The dependency FMT_SMF.1 of FMT_MSA.1/JCRE is discarded. The dependency between FMT_MSA.1/JCRE and FMT_SMF.1 is not satisfied because no management functions are required for the Java Card RE.
- The dependency FAU_SAA.1 of FAU_ARP.1 is discarded. The dependency of FAU_ARP.1 on FAU_SAA.1 assumes that a "potential security violation" generates an audit event. On the contrary, the events listed in FAU_ARP.1 are self-contained (arithmetic exception, ill-formed bytecodes, access failure) and ask for a straightforward reaction of the TSFs on their occurrence at runtime. The JCVM or other components of the TOE detect these events during their usual working order. Thus, there is no mandatory audit recording in the Java Card PP [PP-JC].

9.3.2 Secure Storage and Firmware Upgrade OS

SFR	Dependency	Satisfied by
FTP_ITC.1/SFA-Weaver	None.	n/a
FDP_ACC.1/SFA-Weaver	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/SFA-Weaver
FDP_ACF.1/SFA-Weaver	FMT_MSA.3 Static attribute initialization FDP_ACC.1 Subset access control	FMT_MSA.3 is not required because the security attributes used to enforce the SFA-Weaver SFP are fixed during manufacturing phase and no new objects under control of the SFA-Weaver SFP are created. FDP_ACC.1/SFA-Weaver
FDP_ETC.1	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	FDP_ACC.1/SFA-Weaver
FDP_ITC.1	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialization	FDP_ACC.1/SFA-Weaver FMT_MSA.3 is not required because the security attributes used to enforce the SFA-Weaver SFP are fixed during manufacturing phase and no new objects under control of the SFA-Weaver SFP are created.
FDP_SDI.2	None.	n/a
FTP_ITC.1/Loader	None.	n/a
FDP_UCT.1	FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	FTP_ITC.1/Loader FDP_ACC.1/Loader
FDP_UIT.1	FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path	FTP_ITC.1/Loader

	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	FDP_ACC.1/Loader
FDP_ACC.1/Loader	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/Loader
FDP_ACF.1/Loader	FMT_MSA.3 Static attribute initialization FDP_ACC.1 Subset access control	FMT_MSA.3 is not required because the security attributes used to enforce the Loader SFP are fixed during manufacturing phase and no new objects under control of the Loader SFP are created. FDP_ACC.1/Loader

Table 30 - Secure Storage and Firmware Upgrade OS Dependency Rationale

9.4 **Rationale for the Security Assurance Requirements**

EAL5 is required for this type of TOE and product since it is intended to defend against sophisticated attacks. This evaluation assurance level allows a developer to gain maximum assurance from positive security engineering based on good practices. EAL5 represents the highest practical level of assurance expected for a commercial grade product. In order to provide a meaningful level of assurance that the TOE and its embedding product provide an adequate level of defense against such attacks: the evaluators should have access to the low level design and source code. The lowest for which such access is required is EAL5.

9.4.1 ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE and the embedding product. The standard ALC_DVS.1 requirement mandated by EAL5 is not enough. Due to the nature of the TOE and embedding product, it is necessary to justify the sufficiency of these procedures to protect their confidentiality and integrity. ALC_DVS.2 has no dependencies.

9.4.2 AVA_VAN.5 Advanced methodical vulnerability analysis

The TOE is intended to operate in hostile environments. AVA_VAN.5 "Advanced methodical vulnerability analysis" is considered as the expected level for Java Card technology-based products hosting sensitive applications. AVA_VAN.5 has dependencies on ADV_ARC.1, ADV_FSP.1, ADV_TDS.3, ADV_IMP.1, AGD_PRE.1 and AGD_OPE.1. All of them are satisfied by EAL5.

9.5 **IC Composition rationale**

9.5.1 Common Criteria rationale

Assurance level of the IC evaluation is EAL6 augmented by ALC_FLR.1
Assurance level of the TOE is EAL5 augmented with ALC_DVS.2 and AVA_VAN.5.

Assurance level of the current evaluation is consistent with the assurance level in IC evaluation.

9.5.2 Compatibility between threats (TOE and IC)

IC Threats	Rationale
BSI.T.Leak-Inherent	This threat is related to the information which is leaked from the TOE during usage of the Security IC in order to disclose sensitive data of the TOE. This threat has been considered in the current evaluation.
BSI.T.Phys-Probing	This threat is related to physical probing of the TOE to disclose relevant information. This threat has been considered in the current evaluation.
BSI.T.Malfunction	This threat is related to force malfunctions of the TSF due to environmental stress that could lower or bypass the implemented security mechanisms. This threat has been considered in the current evaluation.
BSI.T.PhysManipulation	This threat is related to physical manipulation of the Security IC. This is covered by the IC evaluation.
BSI.T.Leak-Forced	This threat is related to information which is leaked from the TOE during usage of the Security IC in order to disclose confidential user data of the composite TOE. This is covered by the IC evaluation.
BSI.T.Abuse-Func	This threat is related to the usage of functions of the TOE that are not allowed once the TOE Delivery and can impact the security of the TOE. This threat has been considered in the current evaluation.
BSI.T.RND	This threat is related to the deficiency of random numbers. This is covered by the IC evaluation.
BSI.T.Masquerade-TOE	This threat relates to the capacity of an attacker to produce a chip which is not a genuine TOE but wrongly identifying itself as genuine TOE sample. Mitigation of masquerade requires tightening up the identification by authentication and is covered by IC evaluation.
AUG4.T.Mem-Access	The TOE implements memory access violation mechanisms based on the IC security policy. Therefore, this threat also covered by the TOE evaluation.
JIL.T.Open-SamplesDiffusion	This threat refers to the diffusion of open samples: an attacker may get access to open samples of the TOE and use them to gain information about the TSF (loader, memory management unit, ROM code, …). This threat is also covered by the TOE evaluation.
T.Confid-Applic-Code	Application code of the TOE is protected against unauthorized disclosure. Therefore, this threat also covered by the TOE evaluation.
T.Confid-Applic-Data	Application data of the TOE is protected against unauthorized disclosure. Therefore, this threat also covered by the TOE evaluation.
T.Integ-Applic-Code	Application code of the TOE is protected against unauthorized modification. Therefore, this threat also covered by the TOE evaluation.
T.Integ-Applic-Data	Application data of the TOE is protected against unauthorized modification. Therefore, this threat also covered by the TOE evaluation.

Table 31 - Compatibility between threats

9.5.3 Compatibility between assumptions (TOE and IC)

IC Assumptions	Rationale

BSI.A.Process-Sec-IC	This assumption ensures the security of the delivery and storage of the IC. It is covered by the ALC_DVS.2 activity of the current TOE evaluation.
BSI.A.Resp-Appl	This assumption ensures that security relevant data of the current TOE are properly treated according to the IC security needs. It is covered by the ADV_IMP.1 activity of the TOE evaluation.

Table 32 - Compatibility between assumptions

9.5.4 Compatibility between security objectives for the environment (TOE and IC)

IC OEs	**Rationale**
BSI.OE.Resp-Appl	This objective deals with the treatment of TOE user data by the TOE itself. It is covered by the ADV_IMP.1 activity of the TOE evaluation.
BSI.OE.Process-Sec-IC	This objective is covered by the IC evaluation.
BSI.OE.Lim-Block-Loader	This objective is covered by the IC evaluation.
BSI.OE.Loader-Usage	This objective is covered by the IC evaluation.
BSI.OE.TOE-Auth	This objective is covered by the IC evaluation.
OE.Composite-TOE-Id	Also covered by the current evaluation.
OE.TOE-Id	This objective is covered by the IC evaluation.
OE.Enable-Disable-Secure-Diag	This objective is covered by the IC evaluation.
OE.Secure-Diag-Usage	This objective is covered by the IC evaluation.

Table 33 - Compatibility between security objectives for the environment

9.5.5 Compatibility between Security Objectives (TOE and IC)

BSI.O.Leak-Inherent	Also covered by the current evaluation.
BSI.O.Phys-Probing	Also covered by the current evaluation.
BSI.O.Malfunction	Also covered by the current evaluation.
BSI.O.Phys-Manipulation	Covered by the IC evaluation.
BSI.O.Leak-Forced	Covered by the IC evaluation.
BSI.O.Abuse-Func	Also covered by the current evaluation.
BSI.O.Identification	Covered by the IC evaluation.
BSI.O.RND	Covered by the IC evaluation.
BSI.O.Cap-Avail-Loader	Also covered by the ALC_DVS.2 activity of the current evaluation.
BSI.O.Ctrl-Auth-Loader	Also covered by the current evaluation.
BSI.O.Authentication	Also covered by the current evaluation.
JIL.O.Prot-TSF-Confidentiality	Covered by the IC evaluation.
JIL.O.Secure-Load-ACode	Covered by the IC evaluation.
JIL.O.Secure-AC-Activation	Covered by the IC evaluation.
JIL.O.TOE-Identification	Covered by the IC evaluation.
O.Secure-Load-AMemImage	Covered by the IC evaluation.
O.MemImage-Identification	Covered by the IC evaluation.
AUG1.O.Add-Functions	Covered by the IC evaluation.

AUG4.O.Mem-Access	Also covered by the current evaluation.
O.Firewall	Also covered by the current evaluation.

Table 34 - Compatibility between Security Objectives

9.5.6 Compatibility between Organisational Security Policies (TOE and IC)

IC Policies	Rationale
BSI.P.Process-TOE	This policy is related to the accurate unique identification during IC Development and Production. It was covered by the IC evaluation.
BSI.P.Lim-Block-Loader	Limiting and blocking the loader functionality for loading of Security IC Embedded Software. It was covered by the ALC_DVS.2 activity of the current TOE evaluation.
BSI.P.Ctrl-Loader	Authorized user controls the usage of the Loader functionality in order to protect stored and loaded user data from disclosure and manipulation. It was covered by the IC evaluation.
AUG1.P.Add-Functions	Additional Specific Security Functionality is provided by the IC, including NesLib. It was covered by the IC evaluation.

Table 35 - Compatibility between Organizational Security Policies

9.5.7 Compatibility between SFRs (TOE and IC)

IC SFRs are separated in the following groups as defined in [SOGIS-COMP]:

- IP_SFR: irrelevant IC SFR not being used by the current TOE.
- RP_SFR-SERV: relevant IC SFR being used by the current TOE to implement a security service with associated TSFI.
- RP_SFR-MECH: relevant IC SFR being used by the current evaluation because its security properties providing protection attacks to the TOE.

IC SFR	Rationale
FRU_FLT.2	RP_SFR-MECH
FPT_FLS.1	RP_SFR-MECH
FMT_LIM.1/Test	RP_SFR-MECH
FMT_LIM.2/Test	RP_SFR-MECH
FMT_LIM.1/Loader	RP_SFR-MECH
FMT_LIM.2/Loader	RP_SFR-MECH
FMT_LIM.1/Sdiag	RP_SFR-MECH
FMT_LIM.2/Sdiag	RP_SFR-MECH
FAU_SAS.1	RP_SFR-MECH
FDP_SDC.1	RP_SFR-MECH
FDP_SDI.2	RP_SFR-MECH
FPT_PHP.3	RP_SFR-MECH
FDP_ITT.1	RP_SFR-MECH
FPT_ITT.1	RP_SFR-MECH
FDP_IFC.1	RP_SFR-MECH

FCS_RNG.1	RP_SFR_SERV
FCS_COP.1/TDES	RP_SFR_SERV
FCS_COP.1/AES	RP_SFR_SERV
FDP_ACC.2/Memories	RP_SFR-MECH
FDP_ACF.1/Memories	RP_SFR-MECH
FMT_MSA.3/Memories	RP_SFR-MECH
FMT_MSA.1/Memories	RP_SFR-MECH
FMT_SMF.1/Memories	RP_SFR-MECH
FIA_API.1	RP_SFR-MECH
FDP_ITC.1/Loader	RP_SFR-MECH
FDP_UCT.1/Loader	RP_SFR-MECH
FDP_UIT.1/Loader	RP_SFR-MECH
FDP_ACC.1/Loader	RP_SFR-MECH
FDP_ACF.1/Loader	RP_SFR-MECH
FMT_MSA.3/Loader	RP_SFR-MECH
FMT_MSA.1/Loader	RP_SFR-MECH
FMT_SMR.1/Loader	RP_SFR-MECH
FIA_UID.1/Loader	RP_SFR-MECH
FIA_UAU.1/Loader	RP_SFR-MECH
FMT_SMF.1/Loader	RP_SFR-MECH
FPT_FLS.1/Loader	RP_SFR-MECH
FAU_SAS.1/Loader	RP_SFR-MECH
FAU_SAR.1/Loader	RP_SFR-MECH
FTP_ITC.1/Sdiag	RP_SFR-MECH
FAU_SAR.1/Sdiag	RP_SFR-MECH

Table 36 - Compatibility between SFRs

10 Abbreviations and glossary

[CC]	Common Criteria
[EAL]	Evaluation Assurance Level
[LPU]	Library Protection Unit
[LSE]	Logical Secure Element
[LSI]	Logical Secure element Interface
[MPU]	Memory Protection Unit
[NVM]	Non-Volatile Memory
[ST]	Security Target
[TOE]	Target of Evaluation
[TSF]	TOE Security Functionality
[PP]	Protection Profile

Logical Secure Element	Secure element functionalities, applications and files grouped together to act like a secure element when multiple logical secure element interfaces are supported
Logical Secure element Interface	Logical connection between an endpoint in the terminal and one logical secure element

11 References

[ANSI X9.31]	Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry (rDSA), ANSI X9.31-1998, American Bankers Association
[AGD_OPE]	STSAFEVJ200 Operational Guidance (AGD_OPE), rev E
[AGD_PRE]	STSAFEVJ200 Preparative Procedure (AGD_PRE.1), rev. C
[GP_I2CSPI]	GlobalPlatform I2C/SPI GlobalPlatform Technology APDU Transport over SPI / I2C Version 1.0
[AIS20/31]	Bundesamt fuer Sicherheit in der Informationstechnik. AIS20/31: A proposal for: Functionality classes for random number generators, Bundesamt für Sicherheit in der Informationstechnik (BSI), Version 2.0, September 18th, 2011.
[CERT-IC]	ST33K1M5A and ST33K1M5M B02, NSCIB-CC-2300112-01
[CC31R5P1]	Common Criteria for Information Technology Security Evaluation. Part 1: Introduction to General Model, Version 3.1, Revision 5, April 2016.
[CC31R5P2]	Common Criteria for Information Technology Security Evaluation. Part 2: Security functional components, Version 3.1, Revision 5, April 2016.
[CC31R5P3]	Common Criteria for Information Technology Security Evaluation. Part 3: Security assurance components, Version 3.1, Revision 5, April 2016.
[FIPS 46-3]	FIPS PUB 46-3, Data Encryption Standard, October 25, 1999 (ANSI X3.92), National Institute of Standards and Technology
[FIPS 81]	FIPS PUB 81, DES Modes of Operation, April 17, 1995, National Institute of Standards and Technology
[FIPS 140-2]	FIPS PUB 140-2, Security requirements for cryptographic modules, March 2002 , National Institute of Standards and Technology
[FIPS 180-2]	FIPS PUB 180-2 Secure Hash Standard with Change Notice 1 dated February 25,2004, National Institute of Standards and Technology, U.S.A., 2004
[FIPS 197]	FIPS PUB 197, The Advanced Encryption Standard (AES) U.S. DoC/NIST, November 26, 2001
[GP]	Global Platform Inc., Global Platform Card Specification 2.3, October 2015. Document Reference: GPC_SPE_034
[GP-SGBA]	GlobalPlatform Card - Composition Model - Security Guidelines for Basic Applications - Version 1.0 - June 2012 – ref. GPC_GUI_050
[GP-C]	GlobalPlatform Inc., GlobalPlatform Technology, Contactless Services, Card Specification 2.3 – Amendment C, version 1.2.1, July 2018
[GP-D]	GlobalPlatform Inc., GlobalPlatform Card Technology, Secure Channel Protocol ‘03’, Card Specification v2.2 – Amendment D, Version 1.1.1, July 2014
[GP-E]	GlobalPlatform Inc., GlobalPlatform Card Technology, Security Upgrade for Card Content Management Card Specification v2.3 – Amendment E, version 1.1, October 2016
[GP-H]	GlobalPlatform Inc., GlobalPlatform Technology. Executable Load File Upgrade Card Specification v2.3 - Amendment H, Version 1.1, March 2018
[GP-SGBA]	GlobalPlatform Card - Composition Model - Security Guidelines for Basic Applications - Version 1.0 - June 2012 – ref. GPC_GUI_050
[IEEE 1363a]	IEEE Std 1363a-2004 Standard Specification of Public-Key Cryptography
[JCVM]	Java Card Virtual Machine Java Card Platform, Version 3.0.5, 2015, Oracle Technology Network
[JCAPI]	Java Card Application Programming Interfaces, Version 3.0.5, 2015, Oracle Technology Network
[JCRE]	Java Card Runtime Environment Specification, Classic Edition Version 3.0.5, 2015, Oracle Technology Network
[KS2011]	W. Killmann, W. Schindler, „A proposal for: Functionality classes for random number generators“, Version 2.0, September 18, 2011

[PP-IC]	Security IC Platform Protection Profile with Augmentation Packages Version 1.0 - BSI-CC-PP-0084-2014
[PP-JC]	Java Card System - Open Configuration Protection Profile, April 2020, Version 3.1
[SOGIS-COMP]	Composite product evaluation for Smart Cards and similar devices, version 1.5.1, May 2018
[NISTSP800-90]	NIST SP 800-90 NIST Special Publication 800-90, Recommendation for random number generation using deterministic random bit generators (Revised), National Institute of Standards and Technology (NIST), March 2007
[TS102221]	ETSI TS 102 221 v17.2.0 - Smart Cards; UICC-Terminal interface; Physical and logical characteristics (Release 17)