Security Target Lite of GEOP01 on GSEA01 Security Chip

V1.0

Shenzhen Goodix Technology Co., Ltd

Revision History

Date	Version	Comment
27 Apr 2023	1.0	Release version 1.0

Table of Content

Document information

Information	Content
Keywords	Goodix, Security OS, GSEA01, Secure Element, Crypto Library, Common Criteria, Security Target
Abstract	This document is the Security Target of the Goodix Security OS running on the Security Chip of the GSEA01 family with IC Dedicated Software, developed and provided by Goodix Ltd. The Security OS conforms to Evaluation Assurance Level 5 of the Common Criteria for Information Technology Security Evaluation Version 3.1 Revision 5 with augmentations ALC_DVS.2 and AVA_VAN.5.

Glossary

AES	Advanced Encryption Standard
API	Application Process Interface
APSD	Application Provider Security Domain
CASD	Controlling Authority Security Domain
CBC	Cipher Block Chaining Mode
CRC	Cyclic Redundancy Checks
CRT	Chinese Remainder Theorem
CTR	Counter Mode

DES/TDES	Data Encryption Standard/Triple Data Encryption Standard
DRNG	Deterministic Random Number Generation
ECB	Electronic Code Book Mode
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ES	Embedded Software
HAL	Hardware Abstraction Layer
HCI	Host Controller Interface
NIR	Near Infrared
OFB	Output Feedback Mode
OSCCA	China Office of State Commercial Cryptography Administration
RAM	Random Access Memory
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman Algorithm
TRNG	True Random Number Generator

1 ST Introduction

1.1 ST Reference

See title page.

1.2 TOE Reference

The TOE is named “GEOP01 on GSEA01 Security Chip”. It consists of

Group	Category	Component	Version
IC	IC Hardware	GSEA01 Security IC	A0
	IC Software	IC Dedicated Software	0101
COS	COS framework	Runtime Environment Virtual Machine Common API HCI API	1.0
	GlobalPlatform	GP API GP APDU	
	Proprietary software	Yula NFC Tag application	
		EDA framework	
		Kernel	
Root2	Proprietary Sub OS	Root2 (OS Update, OS Configuration)	1.0
Document	User Manual	GEOP User Manual[48]	1.6
		GEOP Root2 User Manual[49]	1.0
		GEOP01 Preparative Procedures[50]	1.7
		GEOP01 Operational User Guidance[51]	1.4
		GEOP01 Security Guidance[52]	1.4

Table 1 TOE Reference

In this document, the TOE name is abbreviated to “GEOP01”.

1.3 TOE Overview

1.3.1 TOE Introduction

The TOE is a composite TOE with the Security Card Operating System (COS) running on the Goodix GSEA01 Security Chip. 40nm technology with IC Dedicated Software. The GSEA01 Security Chip and associated IC Dedicated Software are Common Criteria certified to EAL5+ [CC3], comparable to a smart card controller.

The TOE Software, other than the IC Dedicated Software, is composed of the following components:

- Virtual Machine Software [28] and a Runtime Environment [26],
- Common Application Programming Interface Software [27],
- Application Programming Interface for HCI [38],
- GlobalPlatform (GP) Software[29],
- OS Update/Config Software (Root2). This component ensures that only Goodix Authorized updates may be applied,
- Proprietary Application Programming Interface Software (Extension API), including OSCCA algorithms (no security claimed) [53],
- Proprietary Native Application, Yula, as a NFC Tag application. (no security claimed)[48],
- EDA (Event Driven Architecture) for task management,
- Kernel, a basic native functional set that provide functions such as non-volatile memory management, key management, cryptographic API, etc..

Figure 1 provides an overview of the TOE and the communication Interfaces.

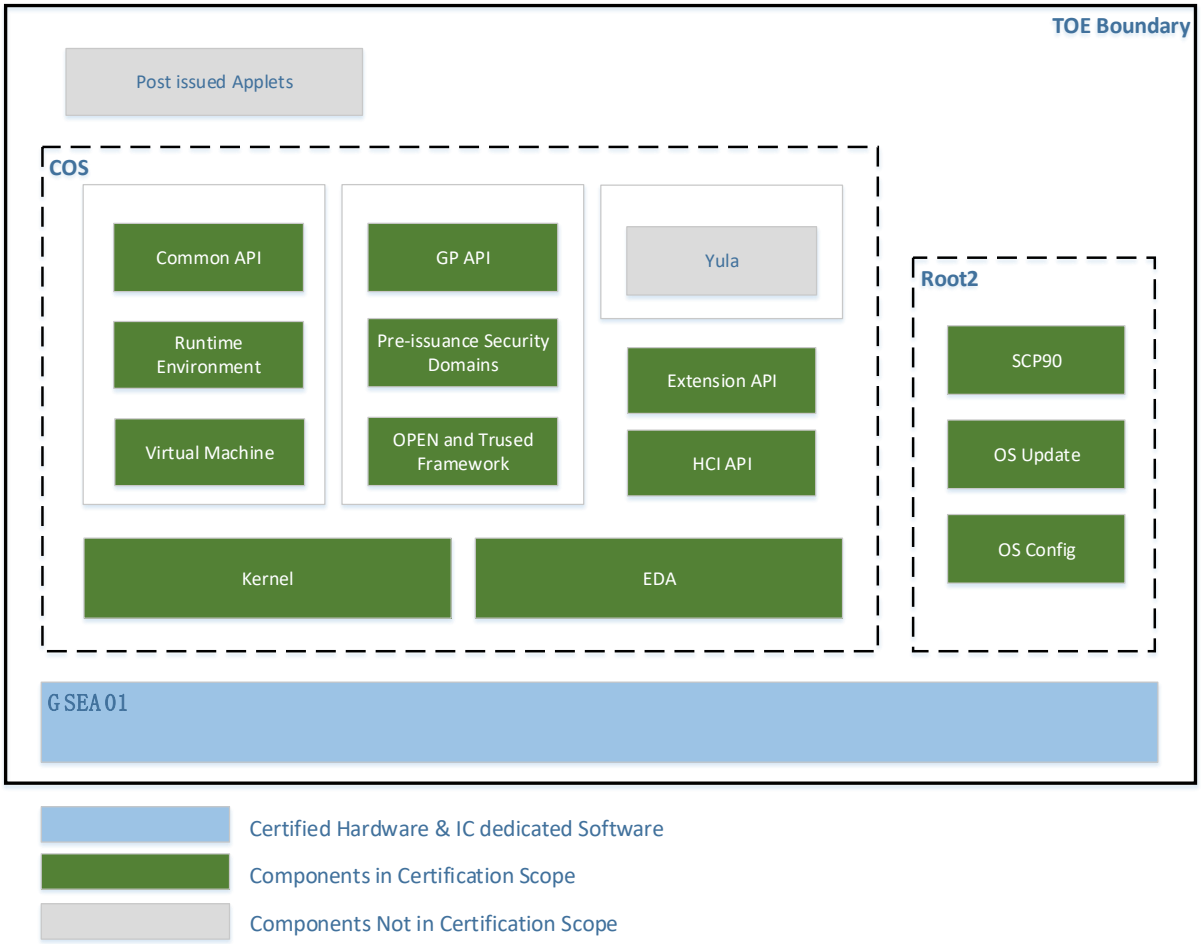

Figure 1 TOE Overview

1.3.2 TOE Type and Usage

The TOE is a Smart Card Platform (IC and OS) along with the native applications and the Java Card System.

The Security Card Operating System (COS) implements GlobalPlatform functionality allowing the installation of various applications, including but not limited to access control, mobile transaction, digital ID and digitial car key, etc. The TOE can load, install, instantiate and execute the off-card verified Javacard applets.

1.3.3 TOE Security Functionality

The TOE provides the following major security functionalities:

- GSEA01 security chip provides cryptographic functions and security features to protect the circuits and its IC Dedicated Software from physical attacks, side channel attacks and perturbation attacks.
- Cryptographic algorithms and functionality:

- DES/TDES for encryption/decryption (CBC and ECB) and MAC generation and verification (2-key/3-key 3DES, Retail-MAC, CMAC). (single DES security not claimed)
- AES (Advanced Encryption Standard) for encryption/decryption (GCM, CBC, ECB, OFB, CFB, CTR) and MAC generation and verification (CMAC)
- RSA and RSA CRT for encryption/decryption and signature generation and verification
- RSA and RSA CRT key generation
- ECC over GF(p) for signature generation and verification (ECDSA)
- ECC over GF(p) key generation for key agreement
- Random number generation conforming to class PTG.2 and DRG.3 of AIS 20/31 [16]
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512 hash algorithm. (security not claimed)
- HMAC (security not claimed)
- OSCCA Algorithms (SM2, SM3, SM4 and SM9) (security not claimed)

- Java Card 3.1.0 functionality:
  - Java Card Virtual Machine for bytecode execution
  - Transient and persistent memory management for applets
  - Applet firewall protection
  - Access control rules between applets and the JCRE
  - Javacard wrapper layer for native implementations
  - Garbage Collection
  - Support for Extended Length APDUs.
  - Sensitive result, Sensitive array, array view
  - Oneshot object
- GlobalPlatform 2.3.1 functionality:
  - Loading and installation of Java Card packages
  - Java CAP file deletion

- Java applet deletion
- Supplementary Security Domains (APSD and CASD) creation
- Applet and Security Domain association
- Key installation
- Applet signature verification
- CVM (PIN) Management
- SCP 02 and SCP 03 secure channels
- Delegated Management, DAP (RSA up to 4096 bits and ECC up to 512 bit)
- Compliance to Secure Element configuration. (security not claimed)

- HCI communication functionality
  - HCI APIs for HCI communication
- Goodix proprietary functionality
  - EDA framework for task management
  - Root2 OS for OS update and OS configuration over SCP 90 secure channel (only available for Goodix authorized entity)
  - Yula NFC Tag application (security not claimed)
  - API for proprietary stream cipher functionality (security not claimed)

1.3.4 TOE Life Cycle

The TOE development and production life cycle is scheduled in phases, which are defined in the Java Card Protection Profile [JCPP].

The Security OS is developed in Phase 1 “Security Embedded Software Development”. At the end of Phase 1, the TOE send the Security OS to Goodix hardware team, in a secure manner, to be programmed in Phase 4.

Phase 2 IC Development, Phase 3 IC Manufacturing as well as Phase 4 IC Packaging of this life cycle are evaluated during IC certification.

In Phase 2 IC Development of GSEA01, access to sensitive design data of GSEA01 is restricted to who are involved in the development of the product.

In Phase 3 IC Manufacturing, the wafer of GSEA01 is produced and tested on wafers. The confidentiality and integrity of any design and configuration data in this phase will

be ensured. This includes secure treatment and insertion of configuration data as well as manufacturing data, which are generated by Goodix.

In Phase 4 IC Packaging, the GSEA01 is embedded into packages. The IC Dedicated Software is programmed into the Flash and tested. At the end of the package testing, the Security OS is loaded to the user Flash area and the Flash Loader is disabled.

In Phase 5, the Composite Product Integrator, the Goodix Javacard Team, pre-personalize the Security OS and conduct tests after the Flash Loader is disabled in the same packaging and testing environment as Phase 4. Then the TOE is delivered to the client in a secure manner, which is evaluated during IC certification.

The TOE is personalized in Phase 6, if necessary. This is out of this certification scope.

In Phase 7, the TOE provides the full set of security functionalities to avoid abuse of the product by untrusted entities.

Note: User Applet development is outside the scope of this certification. Applets can be loaded into Flash memory. Applet loading into Flash memory can be done in phases 5, and 6. Applet loading in phase 7 is also allowed. This post-issuance loading of applets is allowed (except the native applets). The certification is only valid for platforms that return the Platform Identifier as stated in Table 1.

During Phases 1 to 3, the objectives for the environment 3 are covered by the developer's security measures. During phases 4 to 7, the TOE protects itself with its own Security functions in the environment. But additional requirements for the environment must be followed (OE.Resp-Appl, OE.USE_DIAG).

1.3.5 Required non-TOE Hardware/Software/Firmware

The end users of the TOE use the TOE with the loaded applets as a SE. These users communicate with the TOE with SPI interfaces. Therefore, the communication device supporting these interfaces is needed for using the TOE.

The administrators of SEs configure and update the TOE with root2 sub OS, install additional applets or delete applets with CCM functionality. These users require the same equipment as end-users.

The developers of Java Card applets load and execute the applets on the TOE also with the same equipment as end-users. They also need the development tools, byte code verifier for the development.

1.4 TOE Description

1.4.1 Physical scope of the TOE

The TOE consists of Security OS, IC hardware, IC Dedicated Software and guidance documentation.

Group	Category	Component	Version	Format of delivery
IC	IC Hardware	GSEA01 Security IC	A0	module
	IC Software	IC Dedicated Software	0101	binary in ROM or Flash
COS	COS framework	Runtime Environment Virtual Machine Common API HCI API	1.0	binary in Flash
	GlobalPlatform	GP API GP APDU		binary in Flash
	Proprietary software	Yula NFC Tag application		binary in Flash
		EDA framework		binary in Flash
		Kernel		binary in Flash
Root2	Proprietary Sub OS	Root2 (OS Update, OS Configuration)	1.0	binary in Flash
Document	User Manual	GEOP User Manual[48]	1.6	.pdf file
		GEOP Root2 User Manual[49]	1.0	.pdf file
		GEOP01 Preparative Procedures[50]	1.7	.pdf file
		GEOP01 Operational User Guidance[51]	1.4	.pdf file
		GEOP01 Security Guidance[52]	1.4	.pdf file

* See detail software components in Section 1.4.2

Table 2 TOE physical scope

The security IC is delivered to the client as module with IC software, COS and Root2 in the Flash using a secure delivery method with security seals. The user manuals are delivered to the client with emails using PGP signed and encrypted packages.

The TOE can be identified by the TOE ID (see Table 2). The TOE ID can be obtained by using two GET VERSION commands (see [48]). It has the following parts:

Data Element	Length	Value	Description
Part 1			
IC firmware version	2 bytes	0101	IC firmware version is v1.1
Cos version	2 bytes	0100	Cos version is v1.0
Patch version	2 bytes	0000	Patch is not supported, and its version is RFU as 0000
ROOT2 version	2 bytes	0100	Root2 version is v1.0
RFU	19 bytes	N.A.	Internal info
CID	16 bytes	XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX	Chip ID, Different value for each chip
RFU	26 bytes	N.A.	Internal info
Part 2			
ROOT2 patch version	4 bytes	0000 0000	ROOT2 patch version presented by its CRC. No patch is performed for the TOE, so it's set as default with 0000 0000

Table 3 TOE ID

1.4.2 Logical scope of the TOE

The certification of this TOE is a composite certification. The certificate of the underlying hardware platform GSEA01 (certificate ID: NSCIB-CC-21-0369941), which is part of this TOE, is re-used. In the following sections more detailed descriptions of the TOE components are provided. In the description it is also made clear whether a component is covered by a previous certification or whether it is covered in the certification of this TOE.

1.4.2.1 Security IC

The security IC, GSA01, is a hardware platform with ARM SC300 processor, AES, (T)DES cryptographic engines and a PKCC co-processor for RSA, ECC and OSCCA crypto operations. It contains RAM, ROM and Flash which protect the confidentiality and integrity of the stored data, MMU for memory protection, DRNG and TRNG for random number generation and other peripherals like DMA, I2C and SPI. It also has active shield and sensors for the detecting physical or environmental attacks. The security IC is certified according to Common Criteria EAL5+ in NSCIB (certificate ID: NSCIB-CC-21-0369941)

1.4.2.2 IC Dedicated Software

The IC Dedicated Software is certified in the certification of the security IC.

1.4.2.3 Security OS and GlobalPlatform Software

Security OS consists of Native OS, JCVM, JCRE, JCAPI and GP framework. JCVM, JCRE, JCAPI and GP Software are implemented according to the Java Card Specification and Global Platform Specification.

Global Platform Software consists of GP framework and Amendment A, C, D, E. The following GP components are excluded from the certification:

- Secure Element configuration.
- Common Implementation Configuration.

Security OS components version can be identified by using the VERSION command (see [48]). This command returns the platform identification data, which includes the Chip ID, ROM version, Security OS version, Root2 OS version, Java Card OS patch version, etc. GEOP01 version is a data string that allows to identify the Security OS component.

The specific versions of the components are described in [48].

1.4.2.4 Proprietary Software

The TOE implements the proprietary software HCI, EDA, Root2.

The following software is excluded from the certification: Yula NFC Tag Application and its Javacard API & Extension API.

The specific versions of the components are described in [48].

1.4.3 Interfaces of the TOE

1.4.3.1 Electrical and Physical interface

These interfaces are provided by the certified security IC.

1.4.3.2 Logical interface

The logical interface of the TOE is composed of the following:

- Javacard API interface [27]
- GP API interface [30]
- GP APDU command [29][31]
- HCI API interface
- Root2 APDU command [49]
- Yula NFC Tag API Interface [53] (security not claimed)

- Yula NFC command [48] (security not claimed)
- Extension API [53] (security not claimed)

1.4.4 Form of Delivery

The Security OS is delivered together with IC hardware package, including the IC dedicated software, to the applet developer. The delivery package will be sealed with secure tape. The delivery process will also be trackable with signature, which is certified during the security IC certification.

The user guidance and datasheet documents are delivered in electronic form to the user on request as encrypted and signed email attachment.

2 Conformance Claim

2.1 CC Conformance Claim

This Security Target (ST) and the TOE claim conformance to Common Criteria Version 3.1 Part 1[CC1], Part 2[CC2] and Part 3[CC3]:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017, CCMB-2017-04-001
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-002
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-003

Conformance of this ST is claimed for: Common Criteria Part 2 extended and Common Criteria Part 3 conformant.

2.2 PP Claim

The Security Target claims demonstrable conformance to the Java Card Protection Profile – Open Configuration [JCPP]. Only the “Sensitive Result”, “Sensitive Array”,” Monotonic Counters”, “Cryptographic Certificate Management” and “Key Derivation Functions“ packages defined in [JCPP] are claimed in this ST.

2.3 Package Claim

The assurance level for the TOE is EAL5 augmented with the components ALC_DVS.2 and AVA_VAN.5. The evaluation assurance level exceeds the requirement claimed by the [JCPP].

2.4 Conformance Claim Rationale

All sections of this Security Target, in which security problem definition, objectives and security requirements are defined, clearly state which of these items are taken from [JCPP] and which are added in this Security Target. Therefore, the rationales for the items from [JCPP] are not repeated here. Moreover, all additionally stated items in this Security Target do not contradict the items included from the [JCPP]. In case refinement or deletion of the items from the [JCPP], additional justification is provided in the corresponding section of the ST. The operations done for the SFRs taken from [JCPP] are also clearly indicated. The differences between this ST and the claimed Protection Profile are described in the following sections. These considerations show that the Security Target correctly claims demonstrable conformance to [JCPP].

2.4.1 TOE Type Rationale

The TOE type as stated in Section 1.3.2 of this ST corresponds to the TOE type of the PP as stated in Section 1.2 of [JCPP] namely a Java Card platform, implementing the Java Card Specification Version 3.1.0 [26][27][28].

This Security Target also claims conformance to the following packages of security requirements defined in the Smartcard IC Platform Protection profile [ICPP]:

- Package "TDES"
- Package "AES"

2.4.2 Security Problem Definition Rationale

All the items of the security problem definition defined in Section 5 of [JCPP] are taken into this Security Target except that T.INSTALL and T.DELETION in [JCPP] are refined by T.UNAUTHORIZED_CARD_MNGT which extends more threats related to card management. In addition, the following security problems are introduced in this Security Target. All the refined and introduced security problems are additions that [JCPP] allows.

The threat T.COMMUNICATION is included for the secure channel which is additional functionality to the threats in [JCPP].

The threat T.LIFE_CYCLE is included to cover content management attacks which is additional functionality to the threats in [JCPP].

The threat T.UNAUTHORIZED_OS_MNGT is introduced for OS update and config which is additional functionality [JCPP] allows.

The threat T.EXCEPTION-COUNTER is included for the Limited Mode which is additional functionality [JCPP] allows.

The OSP OSP.TOE_ID is included for the pre-personalization function of the TOE and it is an addition to the OSPs in [JCPP].

The assumption A.Process-Sec-IC and A.Resp-Appl are taken from the underlying certified secure IC [47], which are compliant to the Security IC PP [ICPP]. The assumptions A.Resp-Appl in this Security Target includes an application note to further clarify the application context which conforms to [ICPP]. These assumptions are allowed by [JCPP].

2.4.3 SO and SOE Rationale

All the security objectives defined in Section 6 of [JCPP] are taken into this Security Target, except O.LOAD, O.INSTALL and O.DELETION are refined by O.CARD-MANAGEMENT. All the following introduced security objectives are additions to [JCPP].

OE.CARDMANAGEMENT, OE.SCP.RECOVERY, OE.SCP.SUPPORT and OE.SCP.IC in [JCPP] Section 6.2 are replaced by O.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT and O.SCP.IC in this ST. O.SCP.RECOVERY, O.SCP.SUPPORT, and O.SCP.IC are objectives for the TOE as the Smart Card Platform belongs to the TOE for this evaluation. O.CARD-MANAGEMENT is an objective for the TOE as the Card Manager belongs to the TOE for this evaluation. In addition, O.LOAD, O.INSTALL and O.DELETION security objectives in Section 7.4 of [JCPP] are refined by O.CARD-MANAGEMENT. Moving objectives from the environment to the TOE, adds objectives to the TOE without changing the overall objectives. The statement of security objectives is therefore equivalent to the security objectives in the PP.

O.DOMAIN-RIGHTS, O.APPLI-AUTH, O.COMM_AUTH, O.COMM_INTEGRITY and O.COMM_CONFIDENTIALITY are security objectives for the GlobalPlatform API for Secure Channel and Security Domain. They are additional functionality the [JCPP] allows.

The following optional objectives are defined in the Protection Profile and are not included in the ST: O.REMOTE, O.BIO-MNGT and O.EXT-MEM.

The optional packages O.CRT-MNGT, O.MTC-CTR-MNGT, O.SENSITIVE_RESULTS_INTEG and O.SENSITIVE_ARRAYS_INTEG in [JCPP] Appendix 2 are included. Their rationales are defined in the PP.

Additionally, OE.SCP.RECOVERY, OE.SCP.SUPPORT and OE.SCP.IC Security Objectives for the Operational Environment from [JCPP] becomes O.SCP.RECOVERY, O.SCP.SUPPORT and O.SCP.IC Security Objectives for the Smart Card Platform of the TOE in this ST.

O.TOE-ID is included for the pre-personalisation feature of the TOE, which is allowed by [JCPP].

O.RND is part of the security objectives of the certified secure IC [47] that [JCPP] allows.

O.AUTH-OS-MNGT is included for the OS Image update and configuration that [JCPP] allows.

O.EXCEPTION-COUNTER and O.LIMITED-MODE are included for the Limited Mode functions that [JCPP] allows.

The ST introduces the following additional security objectives for the environment: OE.Process_Sec_IC, OE.Resp-Appl,

OE.Process_Sec_IC, OE.Resp-Appl are from the Security IC [ICPP] that is part of this composite product evaluation. Therefore the statement of security objectives for the environment is equivalent to the statement in the Security IC PP [ICPP]. These security objectives for the environment is allowed by [JCPP].

2.4.4 Security Functional Requirement Statement

The Security Functional Requirements for the Java Card component are taken from the Java Card PP [JCPP] except for the following exceptions:

FDP_IFC.2/OSM, FDP_IFF.1/OSM, FDP_UIT.1/OSM, FIA_UID.1/OSM, FMT_MSA.1/OSM, FMT_MSA.3/OSM, FMT_SMF.1/OSM, FMT_SMR.1/OSM, FTP_ITC.1/OSM and FPT_FLS.1/OSM are added in this ST to define the functionality related to OS management.

FPT_PHP.3, FCS_RNG.1/PTG.2, FCS_RNG.1/DRG.3, FAU_SAS.1 and FPT_EMSEC.1 are added from the Security IC [ICPP].O.RND is part of the security objectives of the certified secure IC [47].

3 Security Aspects

This chapter describes the main security issues of the Java Card System and its environment, security aspects, based on [JCPP]. All security aspects described in [JCPP] section 4 are applied. Additional security aspects are introduced in section 3.7 and 3.8.

3.1 Confidentiality

The security aspects #.CONFID-APPLI-DATA, #.CONFID-JCS-CODE and #.CONFID-JCS-DATA of stated in [JCPP] Section 4.1 are applied here as well.

3.2 Integrity

The security aspects #.INTEG-APPLI-CODE, #.INTEG-APPLI-DATA, #.INTEG-JCS-CODE, and #.INTEG-JCS-DATA in [JCPP] Section 4.2 are applied here as well. In addition, the following security aspect is introduced:

3.3 Unauthorized Executions

The security aspects #.EXE-APPLI-CODE, #.EXE-JCS-CODE, #.FIREWALL, and #.NATIVE stated in [JCPP] Section 4.3 are applied here as well.

3.4 Bytecode Verification

The security aspect #.VERIFICATION stated in [JCPP] Section 4.4 are applied here as well.

3.5 Card Management

The security aspect #.CARD-MANAGEMENT, #.INSTALL, #.SID, #.OBJ-DELETION and #.DELETION stated in [JCPP] Section 4.5 are applied here as well.

3.6 Services

The security aspects #.ALARM, #.OPERATE, #.RESOURCES, #.CIPHER, #.KEY-MNGT, #.PINOMNGT, #.SCP and #TRANSACTION stated in [JCPP] Section 4.6 are applied here as well.

3.7 Miscellaneous

The security aspect #.INTEG-APPLI-DATA-PHYS in [JCPP] Appendix 2 are applied here as well.

3.8 OS Management

#.OSM	The TOE allows only Goodix authorized entity to update or configure the Security OS. While performing OS update, the TOE ensures that only authenticated OS Image can be installed with an atomic operation.

3.9 Limited Mode

#.LM	If the Exception Counter reaches the limit, the TOE enters Limited Mode for performing a limited set of functions (e.g. reset the Exception Counter or read audit information.)

4 Security Problem Definition

This chapter describes the security problem definition of the TOE based on [JCPP]. All assets, threats, organizational security policy and assumptions defined in [JCPP] section 5 are applied. Additional assets are introduced in section 4.1.1 and 4.1.2.

4.1 Description of Assets

4.1.1 User Data

The user assets D.APP_CODE, D.APP_C_DATA, D.APP_I_DATA, D.APP_KEYs and D.PIN described in Section 5.1.1 of [JCPP] are the assets of the TOE.

Application Note: The D.APP_KEYs include the Application Provider Security Domains cryptographic keys, Issuer Security Domain cryptographic keys and Verification Authority Security Domain cryptographic keys.

D.CM_DATA	Card management data of the card management environment. To be protected from unauthorized modification.

4.1.2 TSF Data

The TSF assets D.API_DATA, D.CRYPTO, D.JCS_CODE, D.JCS_DATA and D.SEC_DATA described in Section 5.1.2 of [JCPP] are the assets of the TOE. The TOE also has the following assets.

D.TOE_ID	TOE Identification Data for identifying the TOE. To be protected from unauthorized modification.
D.OS_IMAGE	The update image of the Security OS. Only Goodix authorized entity can update the OS image with an atomic operation. To be protected from unauthorized disclosure and modification.
D.CONFIG_DATA	The OS configuration. Only Goodix authorized entity can change OS configuration data. To be protected from unauthorized disclosure and modification.
D.EXCEPTION_COUNTER	The exception counter used for attack detection. When a potential attack is detected the exception counter is updated up to a limit. Once its limit is reached, the TOE is put into the limited mode. To be protected from unauthorized modification.

4.2 Description of Threats

4.2.1 Confidentiality

Since this Security Target claims demonstrable conformance to the [JCPP], the threats T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE and T.CONFID-JCS-DATA described in Section 5.2.1 of [JCPP] are applied here as well.

4.2.2 Integrity

Since this Security Target claims demonstrable conformance to the [JCPP], the threats T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.INTEG-JCS-CODE and T.INTEG-JCS-DATA described in Section 5.2.2 of [JCPP] are applied here as well.

4.2.3 Identity Usurpation

Since this Security Target claims demonstrable conformance to the [JCPP], the threats T.SID.1 and T.SID.2 described in Section 5.2.3 of [JCPP] are applied here as well.

4.2.4 Unauthorized Execution

Since this Security Target claims demonstrable conformance to the [JCPP], the threats T.EXE-CODE.1, T.EXE-CODE.2 and T.NATIVE described in Section 5.2.4 of [JCPP] are applied here as well.

4.2.5 Denial of Service

Since this Security Target claims demonstrable conformance to the [JCPP], the threat T.RESOURCES described in Section 5.2.5 of [JCPP] is applied here as well.

4.2.6 Card Management

The TOE has the following threat refined from the threats T.INSTALL and T.DELETION defined in [JCPP] Section 5.2.6.

T.UNAUTHORIZED_CARD_MNGT Unauthorized Card Management

The attacker performs the following operations without authorization:

- CAP file loading
- CAP file installation (See #.INSTALL for details)
- CAP file or applet extradition
- CAP file or applet deletion (See #.DELETION for details)
- Applet or security domain personalization
- Applet or security domain privilege update

Directly threatened asset(s): D.APP_KEYS, D.APP_C_DATA, D.APP_I_DATA, D.APP_CODE, D.SEC_DATA and D.CM_DATA.

The TOE has the following additional threats other than those defined in [JCPP].

T.COMMUNICATION Communication Channel Exploitation

An attacker exploits the communication channel established between CAD and the TOE to modify or disclose confidential data. All assets are threatened.

T.LIFE_CYCLE Life Cycle

An attacker tries to access an application by reversing the life cycle status of the application. Directly threatened asset(s): D.APP_I_DATA, D.APP_C_DATA, and D.CM_DATA.

4.2.7 Service

Since this Security Target claims demonstrable conformance to the [JCPP], the threat T.OBJ-DELETION described in Section 5.2.7 of [JCPP] is applied here as well.

4.2.8 Miscellaneous

Since this Security Target claims demonstrable conformance to the [JCPP], the threat T.PHYSICAL described in Section 5.2.8 of [JCPP] is applied here as well.

4.2.9 OS Management

The TOE has the following OS management threats not defined in [JCPP].

T.UNAUTHORIZED_OS_MNGT Unauthorized OS Management

An attacker exploits the OS management secure channel established between OS Administrator and the TOE to

- modify/disclose OS Image or OS configuration commands,
- modify TOE ID
- interrupt OS Image update process

Directly threatened asset(s): D.OS_IMAGE, D.CONFIG_DATA, D.TOE_ID.

4.2.10 Limited Mode

The TOE has the following additional threats of the underlying hardware platform not defined in [JCPP].

T.EXCEPTION-COUNTER Exception Counter Manipulation

An attacker tries to manipulate the exception counter without authorization. Directly threatened assets: D.EXCEPTION_COUNTER.

4.3 Organizational Security Policies

Since this Security Target claims demonstrable conformance to the [JCPP], the organizational security policies OSP.VERIFICATION described in Section 5.3 of [JCPP] is applied here as well.

In addition, the following OSP is claimed.

OSP.TOE_ID Identification of the TOE

An accurate TOE identification must be established so that each instantiation of the TOE carries this identification.

4.4 Assumptions

Since this Security Target claims demonstrable conformance to the [JCPP], the organizational security policies A.CAP_FILE and A.VERIFICATION described in Section 5.4 of [JCPP] are applied here as well.

Note that the assumption A.DELETION from [JCPP] is excluded. The Card Manager of the TOE ensures the security of the applet deletion operation. Therefore the assumption is no longer relevant.

The following assumptions from the ST of security IC, GSEA01 [47], is refined in this ST.

A.Process-Sec-IC Protection during Packaging, Finishing and Personalization

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

This means that the Phases after TOE Delivery are assumed to be protected appropriately. For a preliminary list of assets to be protected are:

1. the Security IC Embedded Software and its specifications, implementation and related documentation,
2. Pre-personalisation Data and Personalisation Data including specifications of formats and memory areas, test related data,
3. the user data of the Composite TOE and related documentation, and
4. material for software development support

A.Resp-Appl	Treatment of user data of the Composite TOE All user data of the Composite TOE are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant user data of the Composite TOE (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.
Application Note:	The trusted Applet developers shall well protect their user data. During the TOE usage, the terminal or system in interaction with the TOE, shall ensure the protection (integrity and confidentiality) of their own keys and user data by operational means and/or procedures. Secure TOE communication protocols shall be supported and used by the environment. The Application Provider (AP) must well protect the security of the application together with its security domain keys (D.APP_KEYs). The AP must change its default security domain keys before performing any operation. The Verification Authority (VA) must well protect the security of the application verification key and securely verify the applications to be loaded on the card with the verification key.

5 Security Objectives

5.1 Security Objectives for the TOE

The security objectives O.SID, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_CONFID, O.ARRAY_VIEWS_INTEG, O.NATIVE, O.OPERATE, O.REALLOCATION, O.RESOURCES, O.ALARM, O.CIPHER, O.RND, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION and O.OBJ-DELETION defined in section 6.1 of [JCPP] are applied here.

The security objectives O.CRT-MNGT, O.MTC-CTR-MNGT, O.SENSITIVE_ARRAYS_INTEG, O.SENSITIVE_RESULTS_INTEG defined in [JCPP] Appendix 2 are applied here as well.

In addition, the Security Objectives described in the following sections are defined/refined for the TOE.

5.1.1 Card Management

The security objective for the environment OE.CARD-MANAGEMENT defined in [JCPP] section 6.2 is replaced by O.CARD-MANAGEMENT defined here.

O.CARD-MANAGEMENT Card Management

The card manager of the TOE shall control the access to card management functions such as the loading, installation, update, extradition or deletion of applets. The TOE shall use a mutual authenticated secure channel with integrity and confidentiality protection for the card management commands and messages.

The card manager is an application with specific rights, which is responsible for the administration of the smart card. The card manager is in charge of the life cycle of the whole card, as well as that of the installed applications (applets). The card manager prevents that card content management (loading, installation, deletion) is carried out at invalid states of the card or by non-authorized actors. It also enforces security policies established by the card issuer.

O.SECURITY-DOMAIN Application Security Domain

The Card Issuer shall not be able to access or change the personalized AP Security Domain keys belonged to the AP. Only the AP who owns the Security Domain can access or modify the security Domain key set.

Application Note: APs'Security Domain keyset is used to establish a secure channel between the APs and the platform. The key sets are unknown to the Card Issuer. They must be changed before any operation on the security domain (OE.Resp-Appl).

5.1.2 Security IC

The Security Objectives for the environment OE.SCP.IC, OE.SCP.RECOVERY and OE.SCP.SUPPORT defined in [JCPP] Section 6.2 are replaced by the Security Objectives O.SCP.IC, O.SCP.RECOVERY and O.SCP.SUPPORT of the TOE.

O.SCP.IC — IC Physical Protection

The SCP of the TOE provides security features against physical attacks which addresses the security aspect #.SCP (7).

O.SCP.RECOVERY — SCP Recovery

If there is a loss of power, or if the smart card is withdrawn from the CAD while an operation is in progress, the SCP of the must allow the TOE software 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).

O.SCP.SUPPORT — SCP Support

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

In addition, the following Security Objective is defined for the TOE.

O.TOE-ID — TOE identification

The TOE provides a secure storage in the Flash for the unique identification of the TOE together with its version which allows the user to distinguish the TOE before and after OS Image update.

5.1.3 Random Numbers

O.RNG — Random number quality

The TOE shall ensure the cryptographic quality of random number generation. For instance, random numbers shall not be predictable and shall have sufficient entropy.

The TOE shall ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

5.1.4 OS Management

O.AUTH-OS-MNGT Authorized OS Management

The TOE shall ensure that only Goodix authorized entity can

- update OS Image with atomic operations
- uniquely identify the OS images before and after OS update
- configure the OS

5.1.5 Limited Mode

O.EXCEPTION-COUNTER Exception Counter

The TOE shall ensure that only Card Issuer can reset the Exception Counter.

O.LIMITED-MODE Limited Mode

The TOE shall ensure that only limited set of commands are available when the TOE is put into Limited Mode. The rest operations only return error codes.

5.2 Security Objectives for the operational environment

The security objectives for the operation environment OE.CAP_FILE, OE.VERFICATION and OE.CODE-EVIDENCE defined in [JCPP] section 6.2 are applied here as well. OE.CARD-MANAGEMENT, OE.SCP.IC, OE.SCP.RECOVERY and OE.SCP.SUPPORT in [JCPP] section 6.2 are replaced by O.CARD-MANAGEMENT, O.SCP.IC, O.SCP.RECOVERY and O.SCP.SUPPORT in this ST. In addition, the ST introduced the following SOEs with application notes as required by the IC platform.

OE.Process_Sec_IC Protection during composite product manufacturing

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

OE.Resp-Appl Treatment of user data of the Composite TOE

All user data of the Composite TOE are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant user

data of the Composite TOE (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.

Application Note: The trusted Applet developers shall well protect their user data. During the TOE usage, the terminal or system in interaction with the TOE, shall ensure the protection (integrity and confidentiality) of their own keys and user data by operational means and/or procedures.

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

The Application Provider (AP) must well protect the security of the application together with its security domain keys (D.APP_KEYs).

The AP must change its default security domain keys before performing any operation.

The Verification Authority (VA) must well protect the security of the application verification key and securely verify the applications to be loaded on the card with the verification key.

5.3 Security Objectives Rationale

Section 6.3 in the [JCPP] provides a rationale how the assumptions, threats, and OSPs are addressed by the objectives that are specified in the [JCPP]. The rationales for OE.CARD-MANAGEMENT, OE.SCP.IC, OE.SCP.RECOVERY and OE.SCP.SUPPORT as defined in [JCPP] Section 6.3.4 remains valid for O.CARD-MANAGEMENT, O.SCP.IC, O.SCP.RECOVERY and O.SCP.SUPPORT in this ST.

The following table provide additional tracing from the assumptions, threats and OSPs to objectives introduced or modified by this ST.

Security Problem Definition	Security Objective Rationale
T.UNAUTHORIZED_CARD_MNGT	This threat is covered by the applet management commitments stated in O.CARD-MANAGEMENT. Only the Application Provider who owns the Security Domain can access or modify the security Domain key set as stated in O.SECURITY-DOMAIN.
T.COMMUNICATION	This threat is covered with the use of a mutual authenticated secure channel with integrity and confidentiality protection for the card management commands and messages as stated in O.CARD-MANAGEMENT. Only the Application Provider who owns the Security Domain can access or modify the security Domain key set as stated in O.SECURITY-DOMAIN.

T.LIFE_CYCLE	This threat is covered by the card manager that prevents that card content management (loading, installation, deletion) is carried out at invalid states of the card or by non-authorized actors, as detailed in O.CARD-MANAGEMENT. Only the Application Provider who owns the Security Domain can access or modify the security Domain key set as stated in O.SECURITY-DOMAIN.
T.UNAUTHORIZED_OS_MNGT	This threat is covered by enforcing authorized users to perform OS management operations as stated in O.AUTH-OS-MNGT.
T.EXCEPTION-COUNTER	This threat is covered by enforcing that only Card Issuer can reset the Exception Counter, as defined in O.EXCEPTION-COUNTER. Also, only limited set of commands are available in Limited Mode (O.LIMITED-MODE).
T.PHYSICAL	This threat is covered by only allowing limited set of commands are available, as described in O.LIMITED-MODE. In addition, this is also covered by the physical protections of the underlying platform as defined in O.SCP.IC.
OSP.TOE_ID	Since the TOE provides a secure storage in the Flash for the unique identification of the TOE together with its version which allows the user to distinguish the TOE before and after OS Image update, the OSP is covered by this objective.
A.Process-Sec-IC	Since OE.Process-Sec-IC requires the Composite Product Manufacturer to implement those measures assumed in A.Process-Sec-IC, the assumption is covered by this objective.
A.Resp-Appl	Since OE.Resp-Appl requires the Security IC Embedded Software to implement measures as assumed in A.Resp-Appl, the assumption is covered by the objective.

Table 4 Security Objective Rationale

6 Extended Components Definition

Two extended components defined and described in [ICPP] are applied here as well for the TOE:

6.1 Definition of FCS_RNG

The family FCS_RNG of the class FCS Cryptographic Support is defined and described in the [JCPP].

FCS_RNG Generation of random numbers

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

Component levelling:

FCS_RNG.1 Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic] random number generator [selection: DRG.2, DRG.3, DRG.4, PTG.2, PTG.3, NTG.1] [AIS20] [AIS31] that implements: [assignment: list of security capabilities].

FCS_RNG.1.2 The TSF shall provide random numbers that meet [assignment: a defined quality metric].

6.2 Definition of FAU_SAS

The family FAU_SAS of the class FAU Security Audit is defined and described in the [ICPP].

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

Component levelling:

FAU_SAS.1 Requires the TOE to provide the possibility to store audit data.

Management:	FAU_SAS.1 There are no management activities foreseen.
Audit:	FAU_SAS.1 There are no actions defined to be auditable.
FAU_SAS.1	**Audit storage**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FAU_SAS.1.1	The TSF shall provide [assignment: list of subjects] with the capability to store [assignment: list of audit information] in the [assignment: type of persistent memory].

6.3 Definition of FPT_EMSEC

The family FPT_EMSEC TOE Emanation of the class FPT Protection of the TSF is defined and described in the [JCPP].

Family behavior: This family defines requirements to mitigate intelligible emanations.

Component levelling:

FPT_EMSEC.1 TOE emanation has two constituents:

FPT_EMSEC.1.1	Limit of emissions requires to not emit intelligible emissions enabling access to TSF data or user data.
FPT_EMSEC.1.2	Interface emanation requires not emit interface emanation enabling access to TSF data or user data.
Management:	FPT_EMSEC.1 There are no management activities foreseen.
Audit:	FPT_EMSEC.1 There are no actions defined to be auditable.

FPT_EMSEC.1 TOE Emanation

Hierarchical to:	No other components.
Dependencies:	No other components.
FPT_EMSEC.1.1	The TOE shall not emit [assignment: types of emissions] in excess of [assignment: specified limits] enabling access to [assignment: list of types of TSF data] and [assignment: list of types of user data].
FPT_EMSEC.1.2	The TSF shall ensure [assignment: type of users] are unable to use the following interface [assignment: type of connection] to gain access to [assignment: list of types of TSF data] and [assignment: list of types of user data].

7 Security Requirements

This part of the Security Target defines the detailed security requirements that shall be satisfied by the TOE. The statement of TOE security requirements shall define the functional and assurance security requirements that the TOE needs to satisfy in order to meet the security objectives for the TOE. This chapter consists of the sections "Security Functional Requirements", "Security Assurance Requirements" and "Security Requirements Rationale".

The CC allows several operations to be performed on security requirements (on the component level); refinement, selection, assignment, and iteration are defined in paragraph 8.1 of the CC Part1 [CC1]. These operations are used in [JCPP] and in this Security Target, respectively.

The **refinement** operation is used to add details to requirements, and, thus, further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in **bold text** and removed/changed words are crossed out as ~~crossed out text~~.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made are denoted by showing as *italic text.*

The **selection** operation is used to select one or more options provided by [JCPP] or CC in stating a requirement. Selections having been made are denoted as *underlined italic*.

The **iteration** operation is used when a component is repeated with varying operations. It is denoted by showing brackets "/iteration indicator" and the iteration indicator after the slash.

Security functional requirements from the Protection Profile are applied to this Security Target. In compliance with Application Note 12 in the Protection Profile

7.1 Security Functional Requirements

This section states the security functional requirements for the TOE. For readability and for compatibility with previous versions, requirements are arranged into groups. The following groups defined [JCPP] Section 7.2 are applied here: Core with Logical Channels (CoreG_LC), Installation (InstG), Applet deletion (ADELG), Object deletion (ODELG) and Secure carrier (CarG).

All subjects (prefixed with an "S") defined in [JCPP] Section 7.2 are applied here: S.ADEL, S.APPLET, S.BCV, S.CAD, S.INSTALLER, S.JCRE, S.JCVM, S.LOCAL, S.MEMBER and S.CAP_FILE, except that the S.BCV defined in [JCPP] is refined as S.SD described in the following table together with the new subjects introduced in this ST.

Subject	Description
S.SD	A GlobalPlatform Security Domain represents an off-card entity (e.g. Card Issuer, Application Provider).
S.Root2	Root2 supports secure OS update TOE created by a trusted off-card entity and OS configuration functions.

Table 5 Subjects introduced in this ST

Objects (prefixed with an "O") defined in [JCPP] Section 7.2 are applied here: O.APPLET, O.CODE_CAP_FILE and O.JAVAOBJECT

Information (prefixed with an "I") defined in [JCPP] Section 7.2 are applied here: I.APDU and I.DATA

Security attributes linked to these subjects, objects and information defined in [JCPP] Section 7.2 are applied here: Active Applets, Applet Selection Status, Applet's version number, CAP File AID, Context, Currently Active Context, Dependent package AID, LC Selection Status, LifeTime, Owner, Package ID, Registered Applets, Resident CAP files, Resident Packages, Selected Applet Context, Sharing and Static References.

In addition, the following security attributes are used in this ST.

Security Attributes	Description
Key Set	Key Set of the Secure Channel.
Image Sequence Number	Image Sequence number of the uploaded D.OS_IMAGE.
Security Level	Security Level of the Secure Communication
Secure Channel Protocol	Secure Channel Protocol Version
Session Key	Session key of the Secure Channel
Sequence Counter	Sequence Counter of the Secure Channel Session
ICV	ICV of the Secure Channel Session.
Exception Counter	A counter for the number of exceptions triggered by attacks

Table 6 Security attributes introduced in this ST

Operations (prefixed with "OP") defined in [JCPP] Section 7.2 are applied here: OP.ARRAY_ACCESS, OP.ARRAY_LENGTH, OP.ARRAY_T_ALOAD, OP.ARRAY_T_ASTORE, OP.ARRAY_AASTORE, OP.CREATE, OP.DELETE_APPLET, OP.DELETE_CAP_FILE, OP.DELETE_CAP_FILE_APPLET, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.JAVA, OP.PUT, OP.THROW and OP.TYPE_ACCESS.

In addition, the following Operations are used in this ST.

Operation	Description

OP.READ_CONFIG_ITEM	Reading a configuration field from the configuration area.
OP.MODIFY_CONFIG_ITEM	Writing of a configuration field.
OP.USE_CONFIG_ITEM	Operational usage of configuration field by subjects inside the TOE.
OP.TRIGGER_UPDATE	APDU Command for initialing OS Update procedure.

Table 7 Operations introduced in this ST

Table 8 lists all the security functional requirements of the TOE from section 7.2 and two additional security functional requirements, FDP_SDI.2/ARRAY and FDP_SDI.2/RESULT, in Appendix 2 of [JCPP] are included here. They apply to this Security Target with some modification (i.e. operation performed or replaced) as identified in the following table. In addition, the SFRs added by this ST are also presented in this table.

SFR	Description	Modified/Added in ST
CoreG_LC Management Security Functional Requirements		
FDP_ACC.2/FIREWALL	Complete access control	No
FDP_ACF.1/FIREWALL	Security attribute based access control	No
FDP_IFC.1/JCVM	Subset information flow control	No
FDP_IFF.1/JCVM	Simple security attributes	Modified, see section 7.1.1.1
FDP_RIP.1/OBJECTS	Subset residual information protection	No
FMT_MSA.1/JCRE	Management of security attributes	No
FMT_MSA.1/JCVM	Management of security attributes	No
FMT_MSA.2/FIREWALL_JCVM	Secure security attributes	No
FMT_MSA.3/FIREWALL	Static attribute initialization	No
FMT_MSA.3/JCVM	Static attribute initialization	No
FMT_SMF.1	Specification of Management Functions	No
FMT_SMR.1	Security roles	No
FCS_CKM.1	Cryptographic key generation	Modified, see section 7.1.1.1
FCS_CKM.4	Cryptographic key destruction	Modified, see section 7.1.1.1
FCS_COP.1	Cryptographic operation	Modified, see section 7.1.1.1
FCS_RNG.1	Random number generation	Modified, see section 7.1.1.1
FDP_RIP.1/ABORT	Subset residual information protection	No

FDP_RIP.1/APDU	Subset residual information protection	No
FDP_RIP.1/GlobalArray	Subset residual information protection	No
FDP_RIP.1/bArray	Subset residual information protection	No
FDP_RIP.1/KEYS	Subset residual information protection	No
FDP_RIP.1/TRANSIENT	Subset residual information protection	No
FDP_ROL.1/FIREWALL	Basic rollback	No
FAU_ARP.1	Security alarms	Modified, see section 7.1.1.1
FDP_SDI.2/DATA	Stored data integrity monitoring and action	Modified, see section 7.1.1.1
FDP_SDI.2/ARRAY	Stored data integrity monitoring and action	No
FDP_SDI.2/RESULT	Stored data integrity monitoring and action	No
FDP_SDI.2/MONOTONIC_COUNTER	Stored data integrity monitoring and action	No
FDP_SDI.2/CRT_MNGT	Stored data integrity monitoring and action	No
FCS_COP.1.1/CRT_MNGT	Cryptographic operation	Modified, see section 7.1.1.1
FPR_UNO.1	Unobservability	Modified, see section 7.1.1.1
FPT_FLS.1	Failure with preservation of secure state	No
FPT_TDC.1	Inter-TSF basic TSF data consistency	Modified, see section 7.1.1.1
FIA_ATD.1/AID	User attribute definition	No
FIA_UID.2/AID	User identification before any action	No
FIA_USB.1/AID	User-subject binding	Modified, see section 7.1.1.1
FMT_MTD.1/JCRE	Management of TSF data	No
FMT_MTD.3/JCRE	Secure TSF data	No
InstG Security Functional Requirements		
FDP_ITC.2/Installer	Import of user data with security attributes	No
FMT_SMR.1/Installer	Security roles	No
FPT_FLS.1/Installer	Failure with preservation of secure state	No
FPT_RCV.3/Installer	Automated recovery without undue loss	Modified, see section 7.1.1.2
AdelG Security Functional Requirements		
FDP_ACC.2/ADEL	Complete access control	No

FDP_ACF.1/ADEL	Security attribute based access control	No
FDP_RIP.1/ADEL	Subset residual information protection	No
FMT_MSA.1/ADEL	Management of security attributes	No
FMT_MSA.3/ADEL	Static attribute initialization	No
FMT_SMF.1/ADEL	Specification of Management Functions	No
FMT_SMR.1/ADEL	Security roles	No
FPT_FLS.1/ADEL	Failure with preservation of secure state	No
OdelG Security Functional Requirements		
FDP_RIP.1/ODEL	Subset residual information protection	No
FPT_FLS.1/ODEL	Failure with preservation of secure state	No
CarG Security Functional Requirements		
FCO_NRO.2/CM	Enforced proof of origin	Modified, see section 7.1.1.3
FDP_IFC.2/CM	Complete information flow control	Modified, see section 7.1.1.3
FDP_IFF.1/CM	Simple security attributes	Modified, see section 7.1.1.3
FDP_UIT.1/CM	Data exchange integrity	Modified, see section 7.1.1.3
FIA_UID.1/CM	Timing of identification	Modified, see section 7.1.1.3
FIA_UAU.1/CM	Timing of authentication	Modified, see section 7.1.1.3
FIA_UAU.4/CM	Single-use authentication mechanisms	Modified, see section 7.1.1.3
FMT_MSA.1/CM	Management of security attributes	Modified, see section 7.1.1.3
FMT_MSA.3/CM	Static attribute initialization	Modified, see section 7.1.1.3
FMT_SMF.1/CM	Specification of Management Functions	Modified, see section 7.1.1.3
FMT_SMR.1/CM	Security roles	Modified, see section 7.1.1.3
FTP_ITC.1/CM	Inter-TSF trusted channel	Modified, see section 7.1.1.3
OS Management Security Functional Requirements		
FDP_IFC.2/OSM	Complete information flow control	Added, see section 7.1.2.1
FDP_IFF.1/OSM	Simple security attributes	Added, see section 7.1.2.1
FDP_UIT.1/OSM	Data exchange integrity	Added, see section 7.1.2.1

FIA_UID.1/OSM	Timing of identification	Added, see section 7.1.2.1
FMT_MSA.1/OSM	Management of security attributes	Added, see section 7.1.2.1
FMT_MSA.3/OSM	Static attribute initialization	Added, see section 7.1.2.1
FMT_SMF.1/OSM	Specification of Management Functions	Added, see section 7.1.2.1
FMT_SMR.1/OSM	Security roles	Added, see section 7.1.2.1
FTP_ITC.1/OSM	Inter-TSF trusted channel	Added, see section 7.1.2.1
FPT_FLS.1/OSM	Failure with preservation of secure state	Added, see section 7.1.2.1
Smart Card Platform Security Functional Requirements		
FAU_SAS.1	Audit Data Storage	Added, see section 7.1.3
FCS_RNG.1/PTG.2	Random Number Generation (PTG.2)	Added, see section 7.1.3
FCS_RNG.1/DRG.3	Random Number Generation (class DRG.3)	Added, see section 7.1.3
FPT_EMSEC.1	TOE Emanation	Added, see section 7.1.3
FPT_PHP.3	Resistance to physical attack	Added, see section 7.1.3
Limited Mode Group		
FDP_ACF.1/LM	Security attribute based access control	Added see section 7.1.1.3
FDP_ACC.2/LM	Complete access control	Added see section 7.1.1.3
FMT_MSA.1/LM	Management of security attribute	Added see section 7.1.1.3
FMT_MSA.3/LM	Static attribute initialisation	Added see section 7.1.1.3
FMT_SMF.1/LM	Specification of Management Functions	Added see section 7.1.1.3
FIA_UID.1/LM	Timing of Identification	Added see section 7.1.1.3
FIA_UAU.1/LM	Timing of authentication	Added see section 7.1.1.3

Table 8 Security Functional Requirements from [JCPP]

7.1.1 Security Functional Requirements refined or modified in this Security Target

7.1.1.1 CoreG_LC Group

The Core with Logical Channels SFRs from the [JCPP] are refined by the following SFRs.

FDP_IFF.1/JCVM Simple security attributes

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialization
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: • *subject: S.JCVM* • *security attribute: Currently Active Context*
FDP_IFF.1.2/JCVM	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: • *An operation OP.PUT(S1, S.MEMBER, I.DATA) is allowed if and only if the Currently Active Context is "Java Card Runtime Environment".* • *Any other OP.PUT operations are allowed regardless of the Currently Active Context.*
FDP_IFF.1.3/JCVM	The TSF shall enforce *no additional information flow control SFP rules.*
FDP_IFF.1.4/JCVM	The TSF shall explicitly authorise an information flow based on the following rules: *none*
FDP_IFF.1.5/JCVM	The TSF shall explicitly deny an information flow based on the following rules: *none*
Application note:	The storage of temporary Java Card Runtime Environment's objects references is runtime-enforced ([26], §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.

FCS_CKM.1	**Cryptographic key generation**
Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1/DES	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *TRNG* and specified cryptographic key sizes*: 128, 192 bits* that meet the following: *BSI-TR02102 [25].*
FCS_CKM.1.1/AES	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *TRNG* and specified cryptographic key sizes*: 128, 192, 256 bits* that meet the following: *BSI-TR02102 [25].*

FCS_CKM.1.1/RSA	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *TRNG* and specified cryptographic key sizes *RSA-ND and RSA-CRT: any length that is multiple of 64 from 512 to 4096 bits* that meet the following: *FIPS PUB 186-4[9].*
Application Note:	The keys can be generated and diversified in accordance with [27] specification in classes KeyPair.
FCS_CKM.1.1/ECC	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *TRNG* and specified cryptographic key sizes *ECC: 128, 160, 192, 224, 256, 384, 512 bits* that meet the following: *FIPS PUB 186-4[9].*
Application Note:	The keys can be generated and diversified in accordance with [27] specification in classes KeyPair.
FCS_CKM.1.1/KDF	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ALG_KDF_COUNTER_MODE* and specified cryptographic key sizes *any bits within the specification limit* that meet the following: *NIST SP 800-108 (Recommendation for Key Derivation Using Pseudorandom Functions)[5].*
FCS_CKM.1.1/TLS	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ALG_PRF_TLS12* and specified cryptographic key sizes *of any length within the specification limit* that meet the following: *IETF RFC 5246[12].*

FCS_CKM.4	**Cryptographic key destruction**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *overwriting the keys with random numbers* that meets the following: *none*.
Application Note	• The keys are reset as specified in [27] 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 ([27]).

FCS_COP.1	**Cryptographic Operation**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/TDES The TSF shall perform *decryption and encryption* in accordance with a specified cryptographic algorithm:
- *ALG_DES_CBC_ISO9797_M1*
- *ALG_DES_CBC_ISO9797_M2*
- *ALG_DES_CBC_NOPAD*
- *ALG_DES_ECB_ISO9797_M1*
- *ALG_DES_ECB_ISO9797_M2*
- *ALG_DES_ECB_NOPAD*
- *ALG_DES_CBC_PKCS5*
- *ALG_DES_ECB_PKCS5*

and cryptographic key sizes *112 and 168 bits* that meet the following: *Java Card API specification* [27]

FCS_COP.1.1/DESMAC The TSF shall perform *MAC generation and verification* in accordance with a specified cryptographic algorithm *TDES in outer CBC for Mode:*
- *ALG_DES_MAC4_ISO9797_1_M1_ALG3*
- *ALG_DES_MAC4_ISO9797_1_M2_ALG3*
- *ALG_DES_MAC4_ISO9797_M1*
- *ALG_DES_MAC4_ISO9797_M2*
- *ALG_DES_MAC4_NOPAD*
- *ALG_DES_MAC8_ISO9797_1_M1_ALG3*
- *ALG_DES_MAC8_ISO9797_1_M2_ALG3*
- *ALG_DES_MAC8_ISO9797_M1*
- *ALG_DES_MAC8_ISO9797_M2*
- *ALG_DES_MAC8_NOPAD*
- *ALG_DES_CMAC*

and cryptographic key sizes *112 and 168 bits* that meet the following: *For ALG_DES_CMAC see API specified in Goodix API signatureX class spec [53], for the rest see Java Card API specification[27].*

FCS_COP.1.1/AES The TSF shall perform *decryption and encryption* in accordance with a specified cryptographic algorithm:
- *ALG_AES_BLOCK_128_CBC_NOPAD*
- *ALG_AES_BLOCK_128_ECB_NOPAD*
- *ALG_AES_CBC_ISO9797_M1*
- *ALG_AES_CBC_ISO9797_M2*
- *ALG_AES_CBC_PKCS5*
- *ALG_AES_ECB_ISO9797_M1*
- *ALG_AES_ECB_ISO9797_M2*
- *ALG_AES_ECB_PKCS5*
- *ALG_AES_CFB*
- *ALG_AES_CTR*
- *ALG_AES_OFB*
- *ALG_AES_GCM*

and cryptographic key sizes *128, 192 and 256 bits* that meet the following: *for ALG_AES_OFB see API specified in Goodix API cipherX class spec [53], for ALC_AES_GCM see FIPS 197[10], NIST Special Publication 800-38D Recommendation for BlockCipher[3], for the rest see Java Card API specification[27].*

FCS_COP.1.1/AES_MAC The TSF shall perform *CMAC generation and verification* in accordance with a specified cryptographic algorithm:
- *ALG_AES_CMAC_128*
- *ALG_AES_MAC_128_NOPAD*

and cryptographic key sizes *128, 192 and 256 bits* that meet the following: *see Java Card API specification [27].*

FCS_COP.1.1/RSA The TSF shall perform *decryption and encryption* in accordance with a specified cryptographic algorithm:
- *ALG_RSA_NOPAD*
- *ALG_RSA_PKCS1*
- *ALG_RSA_PKCS1_OAEP*

and cryptographic key sizes *any key length that is a multiple of 64 between 512 and 4096 bits* that meet the following: *Java Card API specification* [27].

FCS_COP.1.1/RSASignature The TSF shall perform *digital signature generation and verification* in accordance with a specified cryptographic algorithm:
- *ALG_RSA_SHA_PKCS1*
- *ALG_RSA_SHA_PKCS1_PSS*
- *ALG_RSA_SHA_224_PKCS1*
- *ALG_RSA_SHA_224_PKCS1_PSS*
- *ALG_RSA_SHA_256_PKCS1*
- *ALG_RSA_SHA_256_PKCS1_PSS*
- *ALG_RSA_SHA_384_PKCS1*
- *ALG_RSA_SHA_384_PKCS1_PSS*
- *ALG_RSA_SHA_512_PKCS1*
- *ALG_RSA_SHA_512_PKCS1_PSS*
- *ALG_RSA_SHA_ISO9796*

and cryptographic key sizes *any key length that is a multiple of 64 between 512 and 4096 bits* that meet the following: *Java Card API specification [27].*

FCS_COP.1.1/RSASignatureMessageRecovery The TSF shall perform *digital signature generation and verification* in accordance with a specified cryptographic algorithm
- *ALG_RSA_SHA_ISO9796_MR*

and cryptographic key sizes *any key length that is a multiple of 64 between 512 and 4096 bits* that meet the following: *Java Card specification [27] and ISO/IEC 9796-2[21]*

FCS_COP.1.1/ECDSA The TSF shall perform *digital signature generation and verification* in accordance with a specified cryptographic algorithm:
- *ALG_ECDSA_SHA*
- *ALG_ECDSA_SHA_224*
- *ALG_ECDSA_SHA_256*
- *ALG_ECDSA_SHA_384*
- *ALG_ECDSA_SHA_512*

and cryptographic key sizes *128, 160, 192, 224, 256, 384 and 512 bits* that meet the following: *Java Card API specification[27].*

FCS_COP.1.1/ECDH	The TSF shall perform *Diffie-Hellman Key Agreement* in accordance with a specified cryptographic algorithm: • *ALG_EC_SVDP_DH* • *ALG_EC_SVDP_DH_KDF* • *ALG_EC_SVDP_DHC* • *ALG_EC_SVDP_DHC_KDF* • *ALG_EC_SVDP_DH_PLAIN* • *ALG_EC_SVDP_DHC_PLAIN* • *ALG_EC_SVDP_DH_PLAIN_XY* and cryptographic key sizes *128, 160, 192, 224, 256, 384* and cryptographic key sizes *128, 160, 192, 224, 256, 384 and 512 bits* that meet the following: *Java Card API specification [27].*
FCS_COP.1.1/DAP	The TSF shall perform *verification of the DAP signature attached to Executable Load Applications* in accordance with a specified cryptographic algorithm • *ALG_RSA_SHA_PKCS1* • *ALG_ECDSA_SHA_256* and cryptographic key sizes *4096(RSA) and 512(EC_FP)* that meet the following: *GP Spec [35].*
FCS_COP.1.1/CRT_MNGT	The TSF shall perform *verification of X.509 Certificate* in accordance with a specified cryptographic algorithm: • *Signature:* – *ALG_RSA_SHA_PKCS1_PSS,* –*ALG_RSA_SHA_224_PKCS1_PSS,* – *ALG_RSA_SHA_256_PKCS1_PSS,* – *ALG_RSA_SHA_384_PKCS1_PSS,* – *ALG_RSA_SHA_512_PKCS1_PSS,* • *Cipher with Hash cannulated against SHA/ SHA224/ SHA256/ SHA384/ SHA512:* – *ALG_RSA_PKCS1* and cryptographic key sizes: *any key length that is a multiple of 64 between 512 and 4096 bits* that meet the following: *Java Card API specification [27]*

FAU_ARP.1	**Security alarms**
Hierarchical to:	No other components.
Dependencies:	FAU_SAA.1 Potential violation analysis
FAU_ARP.1.1	The TSF shall take *one of the following actions:* • *throw an exception,* • *lock the card session (after a predefined number of resetted sessions the card shall switch to Limited Mode),* • *reinitialize the Java Card System and its data,* • *response with error code to S.CAD* • *reset session* 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 and power failure,
- abort of a transaction in an unexpected context [JCAPI3] and ([JCRE3], §7.6.2)
- violation of the Firewall or JCVM SFPs,
- unavailability of resources,
- array overflow,
- checksum mismatch of sensitive arrays
- functionality of a not present Module is invoked
- verification fails of Sensitive Result
- Abnormal environmental condition
- Card Manager Life Cycle inconsistency
- General Fault Injection Detection
- FLASH defects
- Integrity protected persistent data inconsistency
- Integrity protected transient data inconsistency
- Logical Memory Access Violation
- MMU window access violation
- Times of try for PIN verification or SCP authentication reach the limit

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.

FDP_SDI.2/DATA	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1/DATA	The TSF shall monitor user data stored in containers controlled by the TSF for *integrity errors* on all objects, based on the following attributes: *the following integrity protected data:* • *D.APP_KEYs* • *D.PIN* • *D.TOE_ID.*
FDP_SDI.2.2/DATA	Upon detection of a data integrity error, the TSF shall *reset the card session and do the attack velocity check*.
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.

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

FPT_TDC.1	**Inter-TSF basic TSF data consistency**
Hierarchical to:	No other components.
Dependencies:	No dependencies
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret *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 [28] specification,* • *the API tokens defined in the export files of reference implementation* when interpreting the TSF data from another trusted IT product.

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

FIA_USB.1/AID **User-subject binding**
Hierarchical to: No other components.
Dependencies: FIA_ATD.1 User attribute definition.
FIA_USB.1.1/AID The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: *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: *Each uploaded CAP file is associated with a unique CAP file AID.*
FIA_USB.1.3/AID The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: *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 "package AID".

7.1.1.2 InstG Group

The installation SFR from the [JCPP] is refined by the following SFRs.

FPT_RCV.3/INSTALLER **Automated recovery without undue loss**
Hierarchical to: FPT_RCV.2 Automated recovery.
Dependencies: AGD_OPE.1 Operational user guidance.
FPT_RCV.3.1/Installer When automated recovery from *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 *a failure during load/installation of a CAP file/applet and deletion of a CAP file/applet/object*, the TSF shall ensure the return of the TOE to a secure state using automated procedures.
FPT_RCV.3.3/Installer The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding *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], p296: 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 [26], §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 ([26], 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 [ICPP]) 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 (Flash). Data stored on the RAM and subject to such failure is intended to have a limited lifetime anyway (runtime data on the stack, transient objects' contents). According to this, the loss of data within the TSF scope should be limited to the same restrictions of the transaction mechanism.

7.1.1.3 Limited Mode Group

The SFRs for Limited Mode are provided here.

FDP_ACC.2/LM	**Complete access control**
Hierarchical to:	FDP_ACC.1 Subset access control
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.2.1/LM	The TSF shall enforce the *Limited Mode access control SFP* on: • *subject: S.SD, OS Administrator* • *object: O.JAVAOBJECT, O.APPLET and O.CODE_CAP_FILE* and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/LM	The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.
FDP_ACF.1/LM	**Security attribute based access control**

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/LM	The TSF shall enforce the *Limited Mode access control SFP* to objects based on the following:

Subject/Object	*Attributes*
S.SD	*D.EXCEPTION_COUNTER*

FDP_ACF.1.2/LM	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *The D.EXCPTION_COUNTER can be reset by ISD or by the OS Administrator.*
FDP_ACF.1.3/LM	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*
FDP_ACF.1.4/LM	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *Deny all operations other than specific limited operations on all objects if the D.EXCEPTION_COUNTER has reached the limit.*

FMT_MSA.1/LM Management of security attribute

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/LM	The TSF shall enforce the *Limited Mode access control policy* to restrict the ability to *modify* the security attributes: • *D.EXCEPTION_COUNTER,* to • *The Card Issuer for ISD,* • *OS Administrator.*

FMT_MSA.3/LM Static attribute initialisation

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

FMT_SMF.1/LM Specification of Management Functions

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1/LM	The TSF shall be capable of performing the following management functions: - *reset D.EXCEPTION_COUNTER,* - *select ISD* - *get TOE version* - *select Root2*

FIA_UID.1/LM	**Timing of Identification**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1/LM	The TSF shall allow following operations on behalf of the user to be performed before the user is identified. • *select ISD* • *get TOE version* • *select Root2*
FIA_UID.1.2/LM	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.1/LM	**Timing of authentication**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FIA_UAU.1.1/LM	The TSF shall allow *select ISD* on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2/LM	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

7.1.1.4 CarG Group

The card management SFRs from the [JCPP] are refined by the following SFRs.

FCO_NRO.2/CM	**Enforced proof of origin**
Hierarchical to:	FCO_NRO.1 Selective proof of origin.
Dependencies:	FIA_UID.1 Timing of identification.
FCO_NRO.2.1/CM	The TSF shall enforce the generation of evidence of origin for transmitted *application CAP file* at all times.
FCO_NRO.2.2/CM	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 *a new application not-yet-verified package is received*.
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.SD. 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.

FDP_IFC.2/CM	**Complete information flow control**
Hierarchical to:	FDP_IFC.1 Subset information flow control.

Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.2.1/CM	The TSF shall enforce the *CAP FILE LOADING and SCP information flow control SFP* on *S.INSTALLER, S.SD, S.CAD, S.SD and I.APDU* and all operations that cause that information to flow to and from subjects covered by the SFP.
FDP_IFC.2.2/CM	The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

FDP_IFF.1/CM	**Simple security attributes**
Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control FMT_MSA.3 Static attribute initialisation
FDP_IFF.1.1/CM	The TSF shall enforce the *SCP information flow control SFP* based on the following types of subject and information security attributes: *Subjects:* *- S.CAD* *- S.SD of ISD or APSD (card commands),* *- S.INSTALLER (applet installation),* *Information: D.CM_DATA (Installation Application, Card Management Commands)* *Security Attributes: MAC, Keys, etc.*
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: *1. rules defined in GlobalPlatform:* *- loading ([24] Section 9.3.5);* *- installation ([24] Section 9.3.6);* *- extradition ([24] Section 9.4.1);* *- registry update ([24] Section 9.4.2);* *- content removal ([24] Section 9.5]).* *2. For Card Management commands, the external entity must be authenticated with SCP02 or SCP03 protocol.*
FDP_IFF.1.3/CM	The TSF shall enforce the *none*.
FDP_IFF.1.4/CM	The TSF shall explicitly authorise an information flow based on the following rules: *none*.
FDP_IFF.1.5/CM	The TSF shall explicitly deny an information flow based on the following rules: *- The TOE fails to authenticate the user or verify the integrity and authenticity evidences of the application package*

FDP_UIT.1/CM	**Data exchange integrity**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path].
FDP_UIT.1.1/CM	The TSF shall enforce the *CAP FILE LOADING and SCP information flow control SFP* to *receive* user data in a manner protected from *modification, deletion, insertion and replay* errors.

FDP_UIT.1.2/CM	The TSF shall be able to determine on receipt of user data, whether *modification, deletion, insertion, replay* **of the application or card management commands** has occurred.
FIA_UID.1/CM	**Timing of Identification**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1/CM	The TSF shall allow • *application selection* • *secure channel initialization* • *requesting TOE identification data* 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:	User are the roles defined in the component FMT_SMR.1/CM.
FIA_UAU.1/CM	**Timing of authentication**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FIA_UAU.1.1/CM	The TSF shall allow *the TSF mediated actions listed in FIA_UID.1/CM* on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2/CM	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
FIA_UAU.4/CM	**Single-use authentication mechanisms**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.4.1/CM	The TSF shall prevent reuse of authentication data related to *the authentication mechanism used to create a secure communication channel*.
FMT_MSA.1/CM	**Management of security attributes**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/CM	The TSF shall enforce the *CAP FILE LOADING and SCP information flow control SFP* to restrict the ability to *modify* the security attributes • *Key Set,* • *Security Level,* • *Secure Channel Protocol,* • *Session Keys,* • *Sequence Counter,* • *ICV* to • *The Card Issuer for ISD,* • *The Application Provider for APSD.*

FMT_MSA.3/CM	**Static attribute initialisation**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/CM	The TSF shall enforce the *CAP FILE LOADING and SCP 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 *Card Issuer and Application Provider* to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/CM	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1/CM	The TSF shall be capable of performing the following management functions: *The management functions identified in [29]* - *loading ([29] Section 9.3.5);* - *installation ([29] Section 9.3.6);* - *extradition ([29] Section 9.4.1);* - *registry update ([29] Section 9.4.2);* - *content removal ([29]Section 9.5).*

FMT_SMR.1/CM	**Security roles**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1/CM	The TSF shall maintain the roles *Card Issuer (ISD), Supplementary Security Domain (SSD).*
FMT_SMR.1.2/CM	The TSF shall be able to associate users with roles.

FTP_ITC.1/CM	**Inter-TSF trusted channel**
Hierarchical to:	No other components.
Dependencies:	No dependencies
FTP_ITC.1.1/CM	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2/CM	The TSF shall permit **the CAD placed in the card issuer secured environment** to initiate communication via the trusted channel.
FTP_ITC.1.3/CM	The TSF shall initiate communication via the trusted channel for - *loading* - *installation* - *extradition* - *registry update* - *content removal* - *changing the life cycle of the Application or SD*
Application Note:	There is no dynamic package loading on the Java Card platform. New packages can be installed on the card only by the card issuer.

7.1.1.5 Package Sensitive Results

The TOE implements the optional package "Sensitive Results" from [JCPP] Appendix 2.

FDP_SDI.2/RESULT	**Integrity_Sensitive_Result**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1/RESULT	The TSF shall monitor user data stored in containers controlled by the TSF for [*integrity errors* on all objects, based on the following attributes: *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 *throw an exception*.

7.1.2 Security Functional Requirements introduced in this ST

7.1.2.1 OS Management Group

The TOE implements the following OS management SFRs.

FDP_IFC.2/OSM	**Complete information flow control**
Hierarchical to:	FDP_IFC.1 Subset information flow control.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.2.1/OSM	The TSF shall enforce the *OS Management information flow control SFP* on *S.Root2 and I.APDU* and all operations that cause that information to flow to and from subjects covered by the SFP.
FDP_IFC.2.2/OSM	The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

FDP_IFF.1/OSM	**Simple security attributes**
Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control, FMT_MSA.3 Static attribute initialisation
FDP_IFF.1.1/OSM	The TSF shall enforce the *OS Management information flow control SFP* based on the following types of subject and information security attributes: *Subject: S.Root2 (OS management)* *Information: D.OS_IMAGE, D.CONFIG_DATA, I.APDU* *Security Attributes: MAC, Keys, Image Sequence Number, etc.*
FDP_IFF.1.2/OSM	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

	- *the external entity is authenticated with SCP90 protocol.* - *The Image Sequence Number of the new image is larger than the current one*
FDP_IFF.1.3/OSM	The TSF shall enforce the *none*.
FDP_IFF.1.4/OSM	The TSF shall explicitly authorise an information flow based on the following rules: *none*.
FDP_IFF.1.5/OSM	The TSF shall explicitly deny an information flow based on the following rules: - *The TOE fails to authenticate the user* - *The configuration option is to update the SCP keys*

FDP_UIT.1/OSM	**Data exchange integrity**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path].
FDP_UIT.1.1/OSM	The TSF shall enforce the *OS Management information flow control SFP* to *receive* user data in a manner protected from *modification, deletion, insertion and replay* errors.
FDP_UIT.1.2/OSM	The TSF shall be able to determine on receipt of user data, whether *modification, deletion, insertion, replay* **of the OS image or configuration commands** has occurred.

FIA_UID.1/OSM	**Timing of Identification**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1/OSM	The TSF shall allow • *secure channel initialization* • *requesting TOE identification data* on behalf of the user to be performed before the user is identified.
FIA_UID.1.2/OSM	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
Application Note:	User are the roles defined in the component FMT_SMR.1/OSM.

FMT_MSA.1/OSM	**Management of security attributes**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control], FMT_SMR.1 Security roles, FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/OSM	The TSF shall enforce the *OS Management information flow control SFP* to restrict the ability to *modify* the security attributes *all security attributes* to *OS Administrator*.

FMT_MSA.3/OSM	**Static attribute initialisation**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/OSM	The TSF shall enforce the *OS Management information flow control SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/OSM	The TSF shall allow the *OS Administrator* to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/OSM	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1/OSM	The TSF shall be capable of performing the following management functions: - *update OS image* - *config OS functionality and commands*

FMT_SMR.1/OSM	**Security roles**
Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification
FMT_SMR.1.1/OSM	The TSF shall maintain the roles *OS Administrator*.
FMT_SMR.1.2/OSM	The TSF shall be able to associate users with roles.

FTP_ITC.1/OSM	**Inter-TSF trusted channel**
Hierarchical to:	No other components.
Dependencies:	No dependencies
FTP_ITC.1.1/OSM	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2/OSM	The TSF shall permit **the CAD placed in the OS administrator secured environment** to initiate communication via the trusted channel.
FTP_ITC.1.3/OSM	The TSF shall initiate communication via the trusted channel for - *update OS image* - *issue card configuration commands*

FPT_FLS.1/OSM	**Failure with preservation of secure state**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1/OSM	The TSF shall preserve a secure state when the following types of failures occur: • *Corrupted D.OS_IMAGE is received.* • *Unauthorized D.OS_IMAGE is received.* • *The OS Update Process is interrupted.*

.

7.1.3 Security Functional Requirements from the Smart Card Platform

The TOE has the following functionality provided by the underlying hardware platform [47].

FPT_PHP.3	**Resistance to physical attack**
Hierarchical to:	No other components.

Dependencies:	No dependencies.
FPT_PHP.3.1	The TSF shall resist *physical manipulation and physical probing* to the *TSF* by responding automatically such that the SFRs are always enforced.
Refinement:	**The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, "automatic response" means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.**
Application note:	If a physical attack is detected, an alarm is triggered and the chip will reset or generate an interrupt. The alarm is handled by

FCS_RNG.1/PTG.2	**Random Number Generation (PTG.2)**
Hierarchical to:	No other components
Dependencies:	No dependencies
FCS_RNG.1.1/PTG.2	The TSF shall provide a physical random number generator that implements: *(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/PTG.2	The TSF shall provide *numbers of 32 bits* that meet: *(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.*

FCS_RNG.1/DRG.3	**Random Number Generation (Class DRG.3)**
Hierarchical to:	No other components
Dependencies:	No dependencies
FCS_RNG.1.1/DRG.3	The TSF shall provide a deterministic random number generator that implements:

	(DRG.3.1) If initialized with a random seed using a PTRNG of class PTG.2 as random source, the internal state of the RNG shall have at least 112 bits entropy. *Note: The seed is provided by a certified PTG.2 physical TRNG with guaranteed 7.976 bit of entropy per byte.* *(DRG.3.2) The RNG provides forward secrecy.* *(DRG.3.3) The RNG provides backward secrecy even if the current internal state is known.*
FCS_RNG.1.2/DRG.3	The TSF shall provide random numbers that meet: *(DRG.3.4) The RNG, initialized with a random seed from a PTRNG of class PTG.2, generates output for which 2^{48} strings of bit length 128 are mutually different with probability at least $1-2^{-24}$.* *(DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A.*

FAU_SAS.1	**Audit Data Storage**
Hierarchical to:	No other components.
Dependencies:	No other components.
FAU_SAS.1.1	The TSF shall provide *OS Administrator or Tester before TOE Delivery* with the capability to store the *TOE identification information* in the *audit records*.

FPT_EMSEC.1	**TOE Emanation**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_EMSEC.1.1	The TOE shall not emit *variations in power consumption or timing during TOE execution* in excess of *non-meaningful information* enabling access to *TSF data: D.CRYPTO* and *User data: D.PIN, D.APP_KEYs*.
FPT_EMSEC.1.2	The TOE shall ensure *the unauthorized users* are unable to use the following interface *contact PINs or chip surfaces* to gain access to *TSF data D.CRYPTO* and *User data D.PIN, D.APP_KEYS*.

7.2 Security Assurance Requirements

The evaluation assurance level is EAL5 augmented with ALC_DVS.2 and AVA_VAN.5. In the following Table 9, the security assurance requirements are given.

Aspect	Acronym	Description
Development	ADV_ARC.1	Security Architecture design

	ADV_FSP.5	Functional specification
	ADV_IMP.1	Implementation representation
	ADV_INT.2	TSF internals
	ADV_TDS.4	TOE design
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-Cycle Support	ALC_CMC.4	CM capabilities
	ALC_CMS.5	CM scope
	ALC_DEL.1	Delivery procedures
	ALC_DVS.2	Development security
	ALC_LCD.1	Life-cycle definition
	ALC_TAT.2	Tools and techniques
Security Target Evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.3	Depth
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability Assessment	AVA_VAN.5	Advanced methodical vulnerability testing

Table 9: Assurance components

7.3 Security Requirements Rationale

7.3.1 Rationale for Security Functional Requirements

The SFR rationales for the SOs and SFRs provided in [JCPP] Section 7.4.1 and 7.4.2 are applicable for this ST as well.

The rationales for the SOs and SFRs not mentioned in [JCPP] are provided below which shows how the security functional requirements are combined to meet the security objectives.

Objective	TOE Security Functional Requirements
O.CARD-MANAGEMENT	FCO_NRO.2/CM, FDP_IFC.2/CM, FDP_IFF.1/CM, FDP_UIT.1/CM, FIA_UID.1/CM, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMF.1/CM, FMT_SMR.1/CM, FTP_ITC.1/CM Contributes to meet this security objective by enforcing PACKAGE LOADING and Secure Channel Protocol information flow control policy that ensures the integrity and the authenticity of card management operations.
O.SECURITY-DOMAIN	FCO_NRO.2/CM, FDP_IFC.2/CM, FDP_IFF.1/CM, FDP_UIT.1/CM, FIA_UID.1/CM, FIA_UAU.1/CM, FIA_UAU.4/CM, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMF.1/CM, FMT_SMR.1/CM, FTP_ITC.1/CM, Contributes to cover this security objective by enforcing a Security Domain access control policy (rules and restrictions) that ensures a secure card content management.
O.SCP.IC	FAU_ARP.1 contributes to the coverage of the objective by resetting the card session or terminating the card in case of physical tampering. FPR_UNO.1, FPT_EMSEC.1 contributes to the coverage of the objective by ensuring leakage resistant implementations of the unobservable operations. FPT_PHP.3 contributes to the coverage of the objective by preventing bypassing, deactivation or changing of other security features.
O.SCP.RECOVERY	FPT_FLS.1 contributes to the coverage of the objective by preserving a secure state after failure.
O.SCP.SUPPORT	FCS_RNG.1/PTG.2, FCS_RNG.1/DRG.3, FCS_COP.1, FCS_CKM.1, FCS_CKM.4 contribute to meet the objective.
O.TOE-ID	FAU_SAS.1 contribute to the objective by providing secure audit storage for TOE identification.

O.RNG	FCS_RNG.1/PTG.2 and FCS_RNG.1/DRG.3 contribute to the objective by providing true and pseudo random number generators.
O.AUTH-OS-MNGT	FDP_IFC.2/OSM, FDP_IFF.1/OSM, FDP_UIT.1/OSM, FIA_UID.1/OSM, FMT_MSA.1/OSM, FMT_MSA.3/OSM, FMT_SMF.1/OSM, FMT_SMR.1/OSM, FTP_ITC.1/OSM, FPT_FLS.1/OSM Contributes to meet this security objective by enforcing OS Management information flow control policy that ensures the integrity and the authenticity of OS management operations.
O.EXCEPTION-COUNTER	FMT_SMR.1/CM Contributes to cover the objective by defining the security role ISD. FMT_MSA.3/LM Contributes to cover the objective by restricting the initial value of the Exception Counter and allowing nobody to change the initial value. FMT_MSA.1/LM Contributes to cover the objective by only allowing the ISD to modify the Exception Counter. FIA_UAU.1/LM Contributes to cover the objective by requiring authentication before resetting the Exception Counter. FIA_UID.1/LM Contributes to cover the objective by requiring identification before resetting the Exception Counter.
O.LIMITED-MODE	FMT_SMR.1/CM Contributes to cover the objective by defining the security role ISD. FDP_ACC.2/LM Contributes to the coverage of the objective by defining the subject of the Restricted Mode access control SFP. FDP_ACF.1/LM Contributes to cover the objective by controlling access to objects for all operations. FMT_SMF.1/LM Contributes to cover the objective by defining the management functions of the restricted mode. FIA_UAU.1/LM Contributes to cover the objective by requiring authentication before resetting the Exception Counter. FIA_UID.1/LM Contributes to cover the objective by requiring identification before resetting the Exception Counter.

Table 10: Rational for Additional Security Functional Requirements in the ST

7.3.2 Dependencies of Security Functional Requirements

The analysis of the dependency of the SFRs, including the refined SFRs identified in Section 7.1.1 of this ST, in [JCPP] Section 7.4.3.1 is valid for this ST as well.

The dependencies of the SFRs introduced in Section 7.1.2 and 7.1.3, not analyzed in [JCPP] Section 7.4.3.1, are further analyzed below.

Security Functional Requirement	Dependencies	Fulfilled by security requirements
CarG Security Functional Requirements		
FIA_UAU.1/CM·	FIA_UID.1 Timing of identification	FIA_UID.1/CM
FIA_UAU.4/CM	No dependencies	N/A
Limited Mode Security Functional Requirements		
FDP_ACF.1/LM	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation	FDP_ACC.2/LM FMT_MSA.3/LM
FDP_ACC.2/LM	FDP_ACF.1 Security attribute based access control	FDP_ACF.1/LM
FMT_MSA.1/LM	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.2/LM FMT_SMR.1 dependency not met since no associated roles are required FMT_SMF.1/LM
FMT_MSA.3/LM	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1/LM FMT_SMR.1 dependency not met since no associated roles are required
FMT_SMF.1/LM	No dependencies	N/A
FIA_UID.1/LM	No dependencies	N/A
FIA_UAU.1/LM	FIA_UID.1 Timing of identification	FIA_UID.1/LM
OS Management Security Functional Requirements		
FDP_IFC.2/OSM	FDP_IFF.1 Simple security attributes	FDP_IFF.1/OSM
FDP_IFF.1/OSM	FDP_IFC.1 Subset information flow control	FDP_IFC.2/OSM
	FMT_MSA.3 Static attribute initialisation	FMT_MSA.3/OSM
FIA_UID.1/OSM	No dependencies	N/A
FMT_MSA.1/OSM	FDP_ACC.1 1 Subset access control or FDP_IFC.1 Subset information flow control	FDP_IFC.2/OSM
	FMT_SMR.1 Security roles	FMT_SMR.1/OSM

	FMT_SMF.1 Specification of Management Functions	FMT_SMF.1/OSM
FMT_MSA.3/OSM	FMT_MSA.1 Management of security attributes	FMT_MSA.1/OSM
	FMT_SMR.1 Security roles	FMT_SMR.1/OSM
FMT_SMF.1/OSM	No dependencies	N/A
FMT_SMR.1/OSM	FIA_UID.1 Timing of identification	FIA_UID.1/OSM
FDP_UIT.1/OSM	FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control	FDP_IFC.2/OSM
	FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path	FTP_ITC.1/OSM
FTP_ITC.1/OSM	No dependencies	N/A
FPT_FLS.1/OSM	No dependencies	N/A
Security Functional Requirements from the Smart Card Platform		
FPT_PHP.3	No dependencies	N/A
FCS_RNG.1/PTG.2	No dependencies	N/A
FCS_RNG.1/DRG.3	No dependencies	N/A
FAU_SAS.1	No dependencies	N/A
FPT_EMSEC.1	No dependencies	N/A

Table 11: Dependency for SFRs introduced in this ST

7.3.3 Rationale for Security Assurance Requirements

The chosen assurance level EAL5 and the augmentation with the requirements ALC_DVS.2 and AVA_VAN.5 were chosen in order to meet the assurance expectations explained in the following paragraphs. In Table 8, the different assurance levels are shown as well as the augmentations. The augmentations are in compliance with the Protection Profile.

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.

In the particular case of a Security IC the TOE is developed and produced within a complex and distributed industrial process which must especially be protected. Details about the implementation, (e.g. from design, test and development tools as well as Initialization Data) may make such attacks easier. Therefore, in the case of a Security IC, maintaining the confidentiality of the design is very important.

This assurance component is a higher hierarchical component to EAL5 (which only requires ALC_DVS.1). ALC_DVS.2 has no dependencies.

AVA_VAN.5 Advanced methodical vulnerability analysis

Due to the intended use of the TOE, it must be shown to be highly resistant to penetration attacks. This assurance requirement is achieved by the AVA_VAN.5 component.

Independent vulnerability analysis is based on highly detailed technical information. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing high attack potential.

AVA_VAN.5 has dependencies to ADV_ARC.1 “Security architecture description”, ADV_FSP.5 “Security enforcing functional specification”, ADV_TDS.4 “Basic modular design”, ADV_IMP.1 “Implementation representation of the TSF”, AGD_OPE.1 “Operational user guidance”, and AGD_PRE.1 “Preparative procedures”.

All these dependencies are satisfied by EAL5.

It has to be assumed that attackers with high attack potential try to attack Security ICs like smart cards used for digital signature applications or payment systems. Therefore, specifically AVA_VAN.5 was chosen in order to assure that even these attackers cannot successfully attack the TOE.

8 IC Composition rationale

8.1 Common Criteria rationale

Assurance level of the Platform-TOE is EAL5 augmented by ALC_DVS.2 and AVA_VAN.5

Assurance level of the composite-TOE is EAL5 augmented with ALC_DVS.2 and AVA_VAN.5.

Assurance level claimed in the composite-ST is consistent with the assurance level claimed in the Platform-ST.

8.2 Compatibility between threats (TOE and IC)

IC Threats	Rationale	Link to the composite-TOE
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.	T.PHYSICAL
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.	T.PHYSICAL
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.	T.PHYSICAL
T.Phys-Manipulation	This threat is related to physical manipulation of the Security IC. This is covered by the IC evaluation.	T.PHYSICAL
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 threat has been considered in the current evaluation.	T.PHYSICAL
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.	T.LIFE_CYCLE
T.RND	This threat is related to the deficiency of random numbers. This threat has been considered in the current evaluation.	T.INTEGAPPLI-DATA T.CONFID-APPLI-DATA

T.Unauthorized-Access	The TOE implements memory access violation mechanisms based on the IC security policy. Therefore, this threat also covered by the TOE evaluation.	T.PHYSICAL

8.3 Compatibility between assumptions (TOE and IC)

IC Assumptions	Rationale	Link to the composite-TOE
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.	A.Process-Sec-IC
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.	A.Resp-Appl

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

IC OEs	Rationale	Link to the composite-TOE
OE.Resp-Appl	This objective for the environment ensures that the TOE will not disclose security relevant user data of the Composite TOE to unauthorised users or processes when communicating with a terminal. It is covered by the current evaluation.	OE.Resp-Appl
OE.Process-Sec-IC	This objective for the environment ensures that the TOE should be maintained confidentiality and integrity of the TOE and of its manufacturing and test data using security procedures during delivery. It is covered by the current evaluation.	OE.Process_Sec_IC

8.5 Compatibility between Security Objectives (TOE and IC)

IC Objectives	Rationale	Link to the composite-TOE
O.Leak-Inherent	Covered by both IC and current evaluation.	O.SCP.IC O.SCP-SUPPORT
O.Phys-Probing	Covered by both IC and current evaluation.	O.SCP.IC O.SCP-SUPPORT
O.Malfunction	Covered by both IC and current evaluation.	O.OPERATE
O.Phys-Manipulation	Covered by both IC and current evaluation.	O.SCP.IC O.SCP-SUPPORT
O.Leak-Forced	Covered by both IC and current evaluation.	O.SCP.IC O.SCP-SUPPORT
O.Abuse-Func	Covered by both IC and current evaluation.	O.SCP-SUPPORT
O.Identification	Covered by both IC and current evaluation.	O.TOE-ID
O.RND	Covered by both IC and current evaluation.	O.RNG
O.TDES	Covered by both IC and current evaluation.	O.CIPHER
O.AES	Covered by both IC and current evaluation.	O.CIPHER
O.Mem-Access	Covered by both IC and current evaluation.	O.SCP-SUPPORT
O.SFR-Access	Covered by the IC evaluation.	-
O.RSA	Covered by both IC and current evaluation.	O.CIPHER
O.ECC	Covered by both IC and current evaluation.	O.CIPHER

8.6 Compatibility between Organisational Security Policies (TOE and IC)

IC Policies	Rationale	Link to the composite-TOE
P.Process-TOE	This policy is related to the accurate unique identification during IC Development and Production. It was covered by the IC evaluation.	OSP.TOE_ID
P.Crypto-Service	The TOE provides secure hardware based cryptographic services for the IC Embedded Software. It was covered by the IC evaluation.	No such policy is defined in the Composite TOE, for which the corresponding security TSFs are defined.

8.7 Compatibility between SFRs (TOE and IC)

IC SFRs are separated in the following groups as defined in [23]:

- 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	Link to composite-TOE
FRU_FLT.2	RP_SFR-SERV	FAU_ARP.1
FPT_FLS.1	RP_SFR-SERV	FPT_FLS.1/Installer FPT_FLS.1/ADEL FPT_FLS.1/ODEL FPT_FLS.1/OSM
FMT_LIM.1	IP_SFR	The Composite TOE does not use Platform's test features after Platform delivered.
FMT_LIM.2	IP_SFR	The Composite TOE does not use Platform's test features after Platform delivered.
FAU_SAS.1	RP_SFR-SERV	FIA_UID.1/CM FIA_UID.1/OSM

		FIA_UID.1/LM
FDP_SDC.1	RP_SFR-MECH	SM.KEYBASE SM.SECU_ARR_OP SM.CL_INVK
FDP_SDI.2	RP_SFR-SERV	FDP_SDI.2/DATA FDP_SDI.2/ARRAY FDP_SDI.2/RESULT FDP_SDI.2/MONOTONIC_COUNTER FDP_SDI.2/CRT_MNGT
FPT_PHP.3	RP_SFR-SERV	FPT_PHP.3
FDP_ITT.1	RP_SFR-MECH	SM.CL_INVK. TOE invokes the CL APIs to perform cryptographic calculations. These APIs prevent the disclosure of sensitive data of the Composite TOE when it is transmitted between memory, CPU and cryptographic co-processor.
FPT_ITT.1	RP_SFR-MECH	SM.CL_INVK. TOE invokes the CL APIs to perform cryptographic calculations. These APIs prevent the disclosure of sensitive data of the Composite TOE when it is transmitted between memory, CPU and cryptographic co-processor.
FDP_IFC.1	RP_SFR-SERV	FPT_PHP.3. The Platform provides the physical and logical countermeasures to prevent disclosure of sensitive data of the Composite TOE from side channel attacks and other physical attacks while processed or transferred.
FCS_RNG.1/PTG.2	RP_SFR-SERV	FCS_RNG.1/PTG.2 FCS_RNG.1/DRG.3
FCS_COP.1/TDES	RP_SFR-SERV	FCS_COP.1.1/TDES FCS_COP.1.1/DESMAC
FCS_COP.1/AES	RP_SFR-SERV	FCS_COP.1.1/AES FCS_COP.1.1/AES_MAC
FCS_CKM.4/TDES	RP_SFR-SERV	FCS_CKM.4.1
FCS_CKM.4/AES	RP_SFR-SERV	FCS_CKM.4.1
FCS_RNG.1/DRG.3	RP_SFR-SERV	FCS_RNG.1
FCS_COP.1/RSA	RP_SFR-SERV	FCS_COP.1.1/RSA FCS_COP.1.1/RSASignature

		FCS_COP.1.1/RSASignatureMessageRecovery FCS_COP.1.1/DAP FCS_COP.1.1/CRT_MNGT
FCS_COP.1/ECDSA	RP_SFR-SERV	FCS_COP.1.1/ECDSA FCS_COP.1.1/DAP
FCS_COP.1/ECDH	RP_SFR-SERV	FCS_COP.1.1/ECDH
FCS_CKM.1/RSA	RP_SFR-SERV	FCS_CKM.1.1/RSA
FCS_CKM.1/ECC	RP_SFR-SERV	FCS_CKM.1.1/ECC
FCS_CKM.4/CL	RP_SFR-MECH	SM.CL_INVK. The CL of the Platform provides APIs with functions to destroy keys in working-buffer. When the Composite TOE finishes invoking these APIs, the involved keys are also destroyed.
FMT_SMF.1	RP_SFR-MECH	SM.MMU The Composite TOE is running in in CPU privileged level. To set the MMU, The Composite TOE invokes the configuration function in the HAL of the Platform.
FDP_ACC.1	RP_SFR-MECH	SM.MMU SM.KEYBASE The register access for MMU setting, the memory access to the dedicated key storage space both conform to the Memory and Register Access Control Policy of the Platform.
FDP_ACF.1	RP_SFR-MECH	SM.MMU SM.KEYBASE The register access for MMU setting, the memory access to the dedicated key storage space both conform to the Memory and Register Access Control Policy of the Platform.
FMT_MSA.3	IP_SFR	The Composite TOE does not set the default value to the Platform's security attributes for its Memory Access Control Policy
FMT_MSA.1	RP_SFR-MECH	SM.MMU The Composite TOE is running in in CPU privileged level. The register access for MMU setting conforms to the Memory and Register Access Control Policy of the Platform.

9 TOE Summary Specification

9.1 Security Functionality of the TOE

The TOE Security Functionality (TSF) is composed of Security Features (SF) and Security Mechanisms (SM). They together fulfill the security functional requirements (SFR) for the TOE.

The Security Functions and Security Mechanisms related to SFRs of the TOE are summarized in Table 12 and described in section 9.2.

Security Function / Security Mechanism	Name	Name Fulfilled SFR
SF.JCVM	Java Card Virtual Machine	FDP_IFC.1/JCVM FDP_IFF.1/JCVM FMT_MSA.1/JCVM FMT_MSA.1/JCRE FMT_MSA.3/JCVM FMT_SMR.1 FMT_SMF.1 FTP_ITC.1/CM FDP_ROL.1/FIREWALL FDP_ACF.1/FIREWALL FDP_ACC.2/FIREWALL FMT_MSA.2/FIREWALL-JCVM FMT_MSA.3/FIREWALL FIA_UID.2/AID FAU_ARP.1 FPT_FLS.1 FDP_RIP.1/ABORT
SF.GP_CCM	GlobalPlatform Management	FCO_NRO.2/CM FDP_IFF.1/CM FDP_IFC.2/CM FDP_UIT.1/CM FIA_UID.1/CM FMT_MSA.1/CM FMT_MSA.3/CM FMT_SMR.1/CM FMT_SMF.1/CM FTP_ITC.1/CM FIA_ATD.1/AID FIA_UID.2/AID FIA_USB.1/AID FDP_ACC.2/ADEL

		FDP_ACF.1/ADEL FDP_RIP.1/ADEL FDP_RIP.1/bArray FMT_SMF.1/ADEL FMT_MSA.1/ADEL FMT_MSA.3/ADEL FMT_SMF.1/ADEL FMT_SMR.1/ADEL FPT_FLS.1/ADEL FMT_MTD.1/JCRE FMT_MTD.3/JCRE FMT_SMR.1/INSTALLER FDP_ITC.2/INSTALLER FPT_FLS.1/INSTALLER FPT_RCV.3/INSTALLER FCS_COP.1 FAU_ARP.1 FPT_FLS.1
SF.CRYPTO	Cryptographic Functionality	FCS_CKM.1 FCS_CKM.4 FCS_COP.1
SF.RNG	Random Number Generator	FCS_RNG.1/PTG.2 FCS_RNG.1/DRG.3
SF.KEY_STORAGE	Secure Key Storage	FCS_CKM.1 FCS_CKM.4 FDP_SDI.2/DATA FAU_ARP.1 FPT_FLS.1 FPR_UNO.1
SF.LIMITED_MODE	Limited Mode	FDP_ACC.2/LM FDP_ACF.1/LM FMT_MSA.1/LM FMT_MSA.3/LM FMT_SMF.1/LM FDP_UIT.1/OSM FTP_ITC.1/OSM FPT_FLS.1/OSM
SF.OS_UPDATE and SF.CONFIG	Operating System Management	FDP_IFC.2/OSM FDP_IFF.1/OSM FIA_UID.1/OSM FMT_MSA.1/OSM FMT_MSA.3/OSM FMT_SMF.1/OSM

		FMT_SMR.1/OSM FDP_UIT.1/OSM FTP_ITC.1/OSM FPT_FLS.1/OSM
SF.OBJ_MNG	Java Object Management	FDP_RIP.1/OBJECTS FDP_RIP.1/ODEL FPT_FLS.1/ODEL FAU_ARP.1 FPT_FLS.1
SF.TRANSIENT_MEM	Memory Management	FDP_RIP.1/TRANSIENT FIA_ATD.1/AID FDP_RIP.1/APDU FDP_RIP.1/bArray
SF.PERS_MEM	Persistent Memory Management	FAU_ARP.1 FPT_FLS.1 FDP_ROL.1/FIREWALL FDP_RIP.1/ABORT
SF.SENS_ARRAY	Data Error Detection	FAU_ARP.1 FPT_FLS.1 FDP_SDI.2/ARRAY
SF.EXCP_HANDLE	Hardware Protection and Error Handling	FAU_ARP.1 FPT_FLS.1 FPT_PHP.3
SF.TOEID	TOE Identification	FAU_SAS.1
SF.PIN	PIN Management	FDP_SDI.2/DATA FPR_UNO.1
SF.SCA	Side-Channel Protection	FPR_UNO.1 FPT_EMSEC.1
SF.SENS_RES	Sensitive Result	FAU_ARP.1 FPT_FLS.1 FDP_SDI.2/RESULT

Table 12 Security Functions/Mechanisms of the TOE

9.2 Security Functions

9.2.1 SF.JCVM

SF.JCVM provides the bytecode interpreter and the firewall to execute the bytecodes correctly to access the java objects under the proper access control according to the specifications [26], [27] and [28].

9.2.2 SF.GP_CCM

SF.GP provides the card content management functionality and prevent users who are not authorized or have no respective rights to do it. It also provides a secure communication channel for sensitive data exchange to prevent from tampering and disclosure according the GlobalPlatform Specification [29] and GlobalPlatform Amendments A[31], D[34] and E[35].

9.2.3 SF.CRYPTO

SF.CRYPTO provides key creation, key management, key deletion and cryptographic functionality against state-of-the-art attacks, including side-channel analysis. It provides the API in accordance to the Java Card API Specification [27].

9.2.4 SF.RNG

SF.RNG provides random number generation functions TRNG and DRNG, which conform to class PTG.2 and DRG.3 classes in AIS 20/31[16].

9.2.5 SF.KEY_STORAGE

SF.KEY_STORAGE provides a secure data storage for keys. Cryptographic keys are stored with integrity protection.

9.2.6 SF.LIMITED_MODE

SF.LIMITED_MODE prevents the TOE from further attack by providing a Limited Mode which TOE will enter in case that a maximum times of attacks are detected. In this mode, only limited functionality is available.

9.2.7 SF.OS_UPDATE

SF.OS_UPDATE provides a method to update TOE securely. It prevents the updating from unauthorized users or unexpected update packages.

9.2.8 SF.OS_CONFIG

SF.OS_CONFIG provides a method to setup the initial states, pre-personalization data, features configurations, etc. of the TOE securely. It realizes an authentication mechanism to prevent the TOE from unauthorized accessing.

9.2.9 SF.OBJ_MNG

SF.OBJ provides the creation and deletion of java objects under the proper memory resource management and access right control according to the Java Card Runtime Environment Specification [26]. SF.OBJ throws Java Exceptions in case object creation error.

9.2.10 SF.TRANSIENT_MEM

SF.TRANSIENT_MEM provides memory deletion for transient arrays, global arrays, and logical channels according to the Java Card Runtime Environment Specification [26].

9.2.11 SF.PERS_MEM

SF.PERS_MEM provides atomic write operations and transaction management according to the Java Card Runtime Environment Specification[26].

9.2.12 SF.SENS_ARRAY

SF.SENS_ARRAY defines a type of array with a checksum of its content. Applications can use it to check its integrity before access it for Java arrays [27]. The API throws Java exceptions in case the checksum in invalid.

9.2.13 SF.EXCP_HANDLE

SF.EXCP_HANDLE stops the current execution of TOE instructions immediately since any security exception is detected. That is to prevent TOE from working incorrectly risking disclosure of sensitive data or manipulation of TOE behaviors. It also prevents unlimited brute trying on TOE from attackers.

9.2.14 SF.TOEID

SF.TOEID provides the TOE identification stored in a secure audit storage.

9.2.15 SF.PIN

SF.PIN provides an authentication method based on PIN to applets to identify and verify the users securely, which prevent TOE from the disclosure of PIN value and malicious trying brutally.

9.2.16 SF.SCA

SF.SCA provides side-channel protection function for timing attack, SPA, DPA, DFA, EMA and DEMA to prevent keys and PINs leakage while processing them.

9.2.17 SF.SENS_RES

SF.SENS_RES provides applications to check whether a method executes correctly so as to prevent some critical operations or variables are manipulated or bypassed.

10 Bibliography

10.1 Standards

[CC1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017, CCMB-2017-04-001
[CC2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-002
[CC3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-003
[ICPP]	Security IC Platform Protection Profile, Version 1.0, 13th Jan. 2014, registered and certified by Bundesamtfür Sicherheitinder Informationstechnik (BSI) under the reference BSI-PP-0084
[JCPP]	Java card protection profile – open configuration, version 3.1.0 (Apr. 2020), published by oracle, Inc. (bsi-cc-pp-0099-2020)
[1]	NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation, National Institute of Standards and Technology, Edition 2001
[2]	Addendum to NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode, National Institute of Standards and Technology, October 2010
[3]	NIST SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007, Morris Dworkin, National Institute of Standards and Technology
[4]	NIST SP 800-90A, Revision 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015, Elaine Barker and John Kelsey, National Institute of Standards and Technology

[5]	NIST SP 800-108: Recommendation for Key Derivation Using Pseudorandom Functions, revised, October 2009
[6]	ISO/IEC 9797-1: 2011 Information technology -- Security techniques - Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher
[7]	FIPS PUB 81-1980: DES modes of operation, Federal Information Processing Standards Publication, December 2nd, 1980, US Department of Commerce/National Institute of Standards and Technology
[8]	FIPS PUB 180-4-2011: Secure Hash Standard, Federal Information Processing Standards Publication, February 2011, US Department of Commerce/National Institute of Standards and Technology
[9]	FIPS PUB 186-4-2013: Digital Signature Standard, Federal Information Processing Standards Publication, 2013, July, National Institute of Standards and Technology
[10]	FIPS PUB 197-2001: ADVANCED ENCRYPTION STANDARD (AES), Federal Information Processing Standards Publication 197, November 26, 2001, U.S. Department of Commerce/National Institute of Standards and Technology
[11]	ANSI X9.62-1998: Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), September 1998, American National Standards Institute
[12]	IETF RFC 5246: The Transport Layer Security (TLS) Protocol, version 1.2, August 2008
[13]	PKCS#1 v2.2: RSA Cryptography Standard, October 2012, RSA Laboratories
[14]	PKCS#1 v1.5: RSA Encryption, March 1998, RSA Laboratories
[15]	NIST SP 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, National Institute of Standards and Technology, Revised January 2012
[16]	Anwendungshinweise und Interpretationen zum Schema (AIS), Funktionalitätsklassen und Evaluationsmethodologie für deterministische

	Zufallszahlengeneratoren, Version 3, 15.05.2013, Bundesamt für Sicherheit in der Informationstechnik (BSI)
[17]	ISO/IEC 14888-3-2015: Information technology - Security techniques - Digital signatures with appendix -- Part 3: Discrete logarithm based mechanisms, 2016
[18]	ISO/IEC 11770-3-2015: Information technology - Security techniques - Key management -- Part 3: Mechanisms using asymmetric techniques, 2015
[19]	ANSI X9.62: Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), September 1998, American National Standards Institute
[20]	ISO/IEC 15946-1-2008: Information technology - Security techniques Cryptographic techniques based on elliptic curves - Part 1: General, 2008
[21]	ISO/IEC 9796-2 Information Technology - Security Techniques - Digital Signature Scheme Giving Message Recovery, Third edition, 2010
[22]	JIL-ATT-SC: Attack Methods for Smartcards and Similar Devices, Joint Interpretation Library, Version 1.5, February 2009
[23]	JIL-Composite product evaluation for Smart Cards and similar devices, Version 1.5.1, May 2018
[24]	ANSI X9.63: Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography, December 2011, American National Standards Institute
[25]	Cryptographic Mechanisms: Recommendations and Key Lengths, Version 2021-01,
[26]	Runtime Environment Specification, Java Card™ Platform, Version 3.1, Classic Edition, 2019-11
[27]	Application Programming Interface, Java Card™ Platform, v3.1 Classic Edition, 2019-11
[28]	Virtual Machine Specification, Java Card™ Platform, v3.1 Classic Edition, 2019-11

[29]	GlobalPlatform Card Specification v2.3.1, 2018-03
[30]	Java Card API and Export File for Card Specification v2.2.1 (org.globalplatform) v1.6, 2014-03
[31]	GlobalPlatform Card Specification 2.3 Amendment A v1.2 – Confidential Card Content Management, 2019-07
[32]	GlobalPlatform Card Specification 2.2 Amendment B v1.1.3 –Remote Application Management over HTTP, 2015-05
[33]	GlobalPlatform Card Specification 2.3 Amendment C v1.3 – Contactless Services, 2019-07
[34]	GlobalPlatform Card Specification 2.3 Amendment D v1.2 – Secure Channel Protocol '03', 2020-04
[35]	GlobalPlatform Card Specification 2.3 Amendment E v1.1 – Security Upgrade for Card Content Management, 2016-10
[36]	GlobalPlatform Technology Secure Element Configuration v2.0, 2018-08
[37]	GlobalPlatform Card Common Implementation Configuration v2.1, 2018-07
[38]	ETSI TS 102 705 UICC Application Programming Interface for Java Card™ for Contactless Applications V13.0.0 (2019-05)
[39]	GM_T SM2-2012 Elliptic Curve Public Key Cryptography
[40]	GM_T SM3-2012 Cryptographic Hash Algorithm
[41]	GM_T SM4-2012 Block Cipher Algorithm
[42]	GM_T SM9-2016 Identification Cipher Algorithm

10.2 Developer Documents

[43]	GSEA0 Datasheet, Version 1.7, 30 Dec 2021, Shenzhen Goodix Technology Co., Ltd.
[44]	GSEA01 Preparative Procedures, Version 1.6, 18 Jan 2022, Shenzhen Goodix Technology Co., Ltd.
[45]	GSEA01 User Manual, Version 1.0, 11 Jan 2022, Shenzhen Goodix Technology Co., Ltd.
[46]	GSEA01 Security User Guidance Manual, Version 1.11, 18 Jan 2022, Shenzhen Goodix Technology Co., Ltd.
[47]	Security Target of Security Chip GSEA01.x.D00 with IC Dedicated Software, version 1.31, 2021
[48]	GEOP User Manual v1.6, 2023
[49]	GEOP Root2 User Manual v1.0, 2021
[50]	GEOP01 Preparative Procedures v1.7, 2023
[51]	GEOP01 Operational User Guidance v1,4, 2023
[52]	GEOP01 Security Guidance v1.4, 2023
[53]	Goodix API Specification v1.0, 2021

11 Legal and Contact Information

Trademarks and Permissions

GOODIX and other Goodix trademarks are trademarks of Shenzhen Goodix Technology Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Disclaimer

Information contained in this document is intended for your convenience only and is subject to change without prior notice. It is your responsibility to ensure its application complies with technical specifications.

Shenzhen Goodix Technology Co., Ltd. (hereafter referred to as "Goodix") makes no representation or guarantee for this information, express or implied, oral or written, statutory or otherwise, including but not limited to representation or guarantee for its application, quality, performance, merchantability or fitness for a particular purpose. Goodix shall assume no responsibility for this information and relevant consequences arising out of the use of such information.

Without written consent of Goodix, it is prohibited to use Goodix products as critical components in any life support system. Under the protection of Goodix intellectual property rights, no license may be transferred implicitly or by any other means.

Shenzhen Goodix Technology Co., Ltd.

Headquarters: 2F. & 13F., Tower B, Tengfei Industrial Building, Futian Free Trade Zone, Shenzhen, China

TEL: +86-755-33338828 FAX: +86-755-33338099

Website: http://www.goodix.com