TEQS V1.0 Platform

Common Criteria / ISO 15408
Security Target – Public version
EAL4+

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 Reference documents

1.1 EXTERNAL REFERENCES [ER]

[ISO]	ISO references
[ISO 7816]	Identification cards – Integrated circuit(s) cards with contacts - Books 1 to 9
[ISO/IEC 9797-1]	ISO/IEC 9797-1 Information Technology - Security techniques, Message Authentication Codes (MACs), Part 1: Mechanisms using a block cipher, 2011
[Javacard]	**Javacard references**
[JCRE310]	Java Card Platform - Runtime Environment Specification, Classic Edition Version 3.1, February 2021
[JCVM310]	Java Card Platform - Virtual Machine Specification, Classic Edition Version 3.1, February 2021
[JCAPI310]	Java Card Platform - Java Card API, Classic Edition Version 3.1, February 2021
[GP]	**Global Platform references**
[GPCS]	GlobalPlatform Technology - Card Specification v2.3.1, March 2018 Reference: GPC_SPE_034
[Amd D]	GlobalPlatform Card Technology - Secure Channel Protocol '03' Card Specification v2.3 – Amendment D v1.2 Reference: GPC_SPE_014
[Amd E]	GlobalPlatform Card Technology - Security Upgrade for Card Content Management Card Specification v2.3 – Amendment E v1.1 Reference: GPC_SPE_042
[Amd F]	GlobalPlatform Card - Secure Channel Protocol '11' Card Specification v2.3 – Amendment F v1.3 Reference: GPC_SPE_093
[Amd H]	GlobalPlatform Card - Executable Load File Upgrade Card Specification v2.3 – Amendment H v1.1 Reference: GPC_SPE_120
[CIC]	Common Implementation Configuration v2.1 Reference: GPC_GUI_080
[SE_CFG]	GlobalPlatform Secure Element Configuration v2.0 Reference: GPC_GUI_049
[CC]	**Common Criteria references**
[CC-1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.
[CC-2]	Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
[CC-3]	Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
[CCDB]	Common Criteria Supporting Document, Mandatory Technical Document – Composite product evaluation for Smart Cards and similar devices Version 1.5.1, May 2018.
[PP-GP]	GlobalPlatform Technology - Secure Element Protection Profile Ref: GPC_SPE_174, Version 1.0
[PP-JCS]	Java Card System – Open Configuration Protection Profile Ref: BSI-CC-PP-0099-V2-2020, Version 3.1, April 2020
[PP/0084]	Security IC Platform Protection Profile with augmentation Packages Ref: BSI-CC-PP-0084-2014

[ST_IC]	Common Criteria Information Technology Security Evaluation - S3SSE2A - ST(Security Target) Lite, Version 0.1, April 30th 2024, Samsung Electronics
[NIST]	**NIST references**
[FIPS PUB 180-4]	NIST, Secure Hash, Standard (SHS), 2012
[FIPS PUB 186-4]	NIST, Digital Signature Standard (DSS), 2013
[FIPS 197]	Federal Information Processing Standards Publication 197 (FIPS PUB 197), Advanced Encryption Standard (AES), 2001
[NIST-SP800-38A]	NIST, SP800-38A Recommendation for Block Cipher Modes of Operation: Methods and Techniques
[NIST-SP800-38B]	NIST, SP800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005
[NIST-SP800-38D]	NIST, SP800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007
[OTHERS]	**Other references**
[AIS 20/31]	A proposal for: Functionality classes for random number generators, version 2.0, 18.09.2011, Bundesamt für Sicherheit in der Informationstechnik
[ANSI-X9.63]	ANSI-X9.63, Key Agreement and Key Transport Using Elliptic Curve Cryptography, 2011
[ICAO Doc9303]	ICAO: Machine Readable Travel Documents, ICAO Doc9303, Part 11: Security Mechanisms for MRTDSs, seventh edition, 2015
[PKCS#1]	PKCS #1 v2.2: RSA Cryptographic Standard, https://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11300-wp-pkcs-1v2-2-rsa-cryptography-standard.pdf, 27.10.2012
[RFC2104]	RFC2104, HMAC: Keyed-Hashing for Message Authentication

1.2 INTERNAL REFERENCES [IR]

[AGD]	**TOE guidance documentation**
[PRE]	Preparative guidance On CC platforms – TEQS V1.0 Ref: D1622628, Release 1.1, November 2024
[OPE]	Operational guidance on CC platforms – TEQS V1.0, With or Without Controlling Authority And Optional Verification Authority Ref: D1622715, Release 1.2, November 2024
[OPE_VA]	Operational guidance on CC platforms for Verification Authority – TEQS V1.0 Ref: D1622983, Release 1.0, July 2024
[TechNote]	Guidance for profile set up vs. JavaCard System Protection Profile Ref: D1578508, Release 1.6, July 4th 2024
[GUI_BasicApp]	GlobalPlatform Card - Composition Model - Security Guidelines for Basic Applications Ref: GPC_GUI_050, Version 2.0, November 2014, GlobalPlatform
[GUI_SecureApp]	Guidance for Secure application development on Thales Embedded Secure Elements (eSE) Ref: D1623097, Release 1.0, February 2025
[App_Mngt]	Application Verification for Certified Secure Elements – External Procedure Ref: D1258682, Release C03, February 2021
[PatchLoad_Mngt]	Patch Loading Management for Certified Secure Elements – External Procedure Ref: D1344508, Release A04, March 2022
[Ident_Conf]	TEQS V1.0 Platform - Identification & Configurability Ref: D1627926, Release 1.5, February 12th 2025
[APDU_Guide]	TEQS v1.0 APDU Guide Ref: D1621521, Release 1.1, July 5th 2024

2 Acronyms

AES	Advanced Encryption Standard
AID	Application Identifier
AM	Authorized Management
AP	Application Provider
APDU	Application Protocol Data Unit
API	Application Programming Interface
APSD	Application Provider Security Domain
CA	Controlling Authority
CAD	Card Acceptance Device
CASD	Controlling Authority Security Domain
CBC	Cipher Block Chaining
CC	Common Criteria
CPU	Central Processing Unit
CVM	Cardholder Verification Method
DAP	Data Authentication Pattern
DES	Data Encryption Standard
DM	Delegated Management
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
ELF	Executable Load File
GASD	GemActivate Security Domain
GP	GlobalPlatform
HMAC	Keyed-Hash Message Authentication Code
IC	Integrated Circuit
ISD	Issuer Security Domain
IT	Information Technology
JCAPI	JavaCard API
JCRE	JavaCard Runtime Environment
JCS	JavaCard System
JCVM	JavaCard Virtual Machine
KDF	Key Derivation Function
MAC	Message Authentication Code
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
OTA	Over-The-Air
PIN	Personal Identification Number
PP	Protection Profile
RAM	Random Access Memory
RMI	Remote Method Invocation
RNG	Random Number Generator
RSA	Rivest / Shamir / Adleman asymmetric algorithm
SAR	Security Assurance Requirement
SCP	Secure Channel Protocol; or (ETSI) Smart Card Platform
SD	Security Domain
SSD	Supplementary Security Domain
ST	Security Target
TDES	Triple Data Encryption Standard
TOE	Target Of Evaluation
TSF	TOE Security Functionality
VA	Verification Authority
VASD	Verification Authority Security Domain

3 Security Target introduction

3.1 SECURITY TARGET IDENTIFICATION

Title: TEQS V1.0 Platform – Security Target
Version: 1.0p
Author: Thales DIS
Reference: D1620313
Publication date: 12/02/2025

3.2 TOE IDENTIFICATION

Product name: TEQS V1.0
TOE name: Platform part of the TEQS V1.0 software
TOE revision: 1.0

- **TOE software identification** = See below *TOE identification data
- **TOE documentation:** Guidance [AGD]
- **TOE hardware part:** S3SSE2A security controller

Developer: Thales DIS

***TOE identification data**

The significant part is noted in bold.

Identification data Get Data command (tag FE)

Returned Value FE15060A2B060104012A026E01030607**D0027515E70120**

Field	Value
Javacard version	2B060104012A026E0103
OS information	**D0027515E7** (PDM counter) **0120** (OS release)

OS update Identification data Get Data command (tag FD)

Returned Value FD04**00000000**

OS Update Version = **00000000** (no patch present in the TOE)

3.3 TOE OVERVIEW

TOE type: eSE opened platform implementing the Javacard and GlobalPlatform standards.

Product description:

TEQS V1.0 is an Embedded Secure Element (eSE) product intended to be embedded in an Android SE-ready mobile device. As such, it ensures that sensitive data is stored in a safe place and information is given to only authorized applications and people.

TEQS V1.0 is built upon an opened platform implementing the [Javacard] and GlobalPlatform [GP] standards, meaning that additional applications - which may not be known at the time of the present evaluation - can be remotely loaded and installed on the eSE “post-issuance”, i.e. after the mobile device has been delivered to the end-user. Applications can also be installed “pre-issuance” during the pre-personalization or personalization phases. Whatever the scenario (pre-issuance or post-issuance), applications’ loading and installation are secured by the GP security mechanisms and verification processes.

TEQS V1.0 is able to communicate with the host mobile device by means of [ISO 7816] (T=1) over SPI contact protocol.

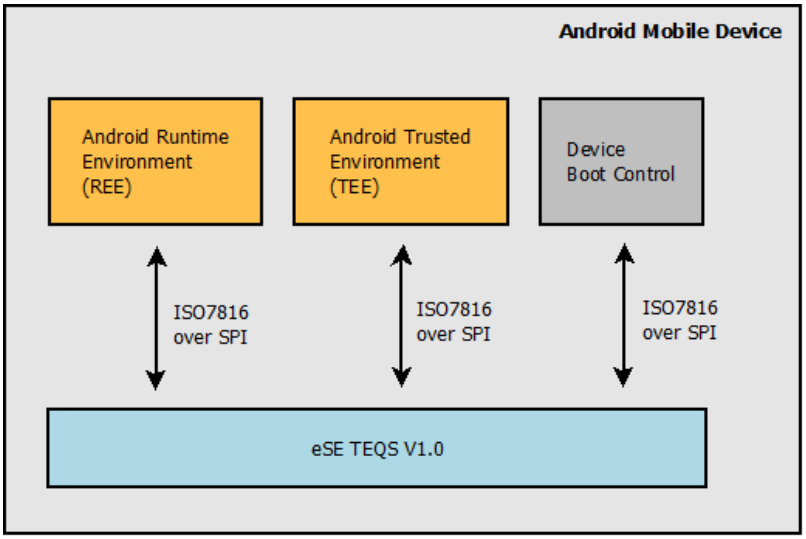

Figure 1: TOE product environment

For the present evaluation, the **Target of Evaluation (TOE)** is the platform part of the TEQS V1.0 software. The TOE boundaries encompass:

- **The Javacard System (JCS)** implemented according to the [Javacard] standard, which manages and executes applications called applets. It also provides Javacard APIs for applet development.
- **The GlobalPlatform (GP) functionalities** implemented according to the [GP] standard, which provide a common and widely used interface to communicate with a secure element and manage applications in a secure way.
- **The GemActivate application**, which is the Thales proprietary solution to activate services and/or load software patches post-issuance, under OEMs and Thales administration.
- **The S3SSE2A Integrated Circuit**.
- **The guidance documentation [AGD]**.

3.3.1 Available Non-TOE Hardware/Software/Firmware

This ST follows the Java Card PP approach, which consists of focusing on the definition of security problems, objectives and requirements that are specific to Java Card and GlobalPlatform features.

Therefore, formally, non-TOE components are the following:

- Bytecode Verifier (off-card component)
- In order to manage distant secure channel according to [GP], a remote system must be able to establish a connection with TOE and therefore must possess shared secret with TOE.
- Applets are supposed to be used with the platform to communicate to external world. Applet can create a dedicated secure channel using platform services. In such case, a remote system must be able to establish a connection with applet and therefore must possess shared secret with applet.

4 TOE Description

4.1 ARCHITECTURE OF TEQS V1.0

The high-level architecture of TEQS V1.0 on S3SSE2A can be represented by Figure 2. In this figure, the elements in **blue** are configurable.

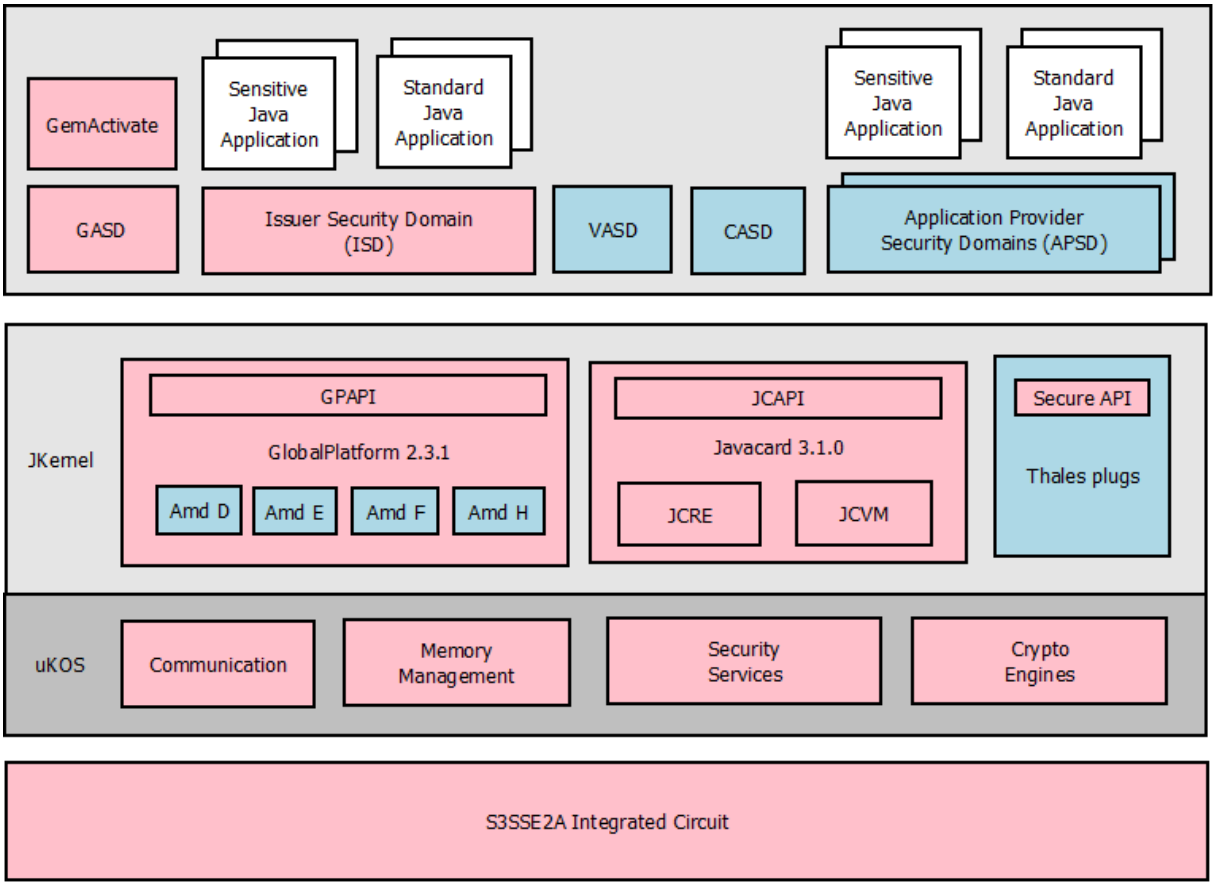

Figure 2: TEQS V1.0 architecture

The architecture can be decomposed in three layers:

- The hardware layer composed of the S3SSE2A integrated circuit.
- The TEQS V1.0 platform, which is the operating system of the product.
- The application layer, encompassing standard and sensitive applications, as well as the security domains (ISD, GASD, VASD, CASD and APSDs).

4.2 TOE BOUNDARIES

4.2.1 TOE physical boundaries

The S3SSE2A IC is a tamper-proof chip in Wafer Level Chip Scale Package (WLCSP) format, which can be soldered in any device PCB.

For the present evaluation, the TOE physical boundaries encompass the S3SSE2A IC with the Thales TEQS V1.0 embedded software. Any other item is outside the scope of the evaluation.

4.2.2 TOE logical boundaries

The present Security Target claims conformance to the [PP-GP] protection profile; the TOE logical boundaries are delimited (dash line in **red**) in Figure 3.

In this figure, the TSF components have been put in **yellow** color. The other components (in white color) do not participate to the TOE security.

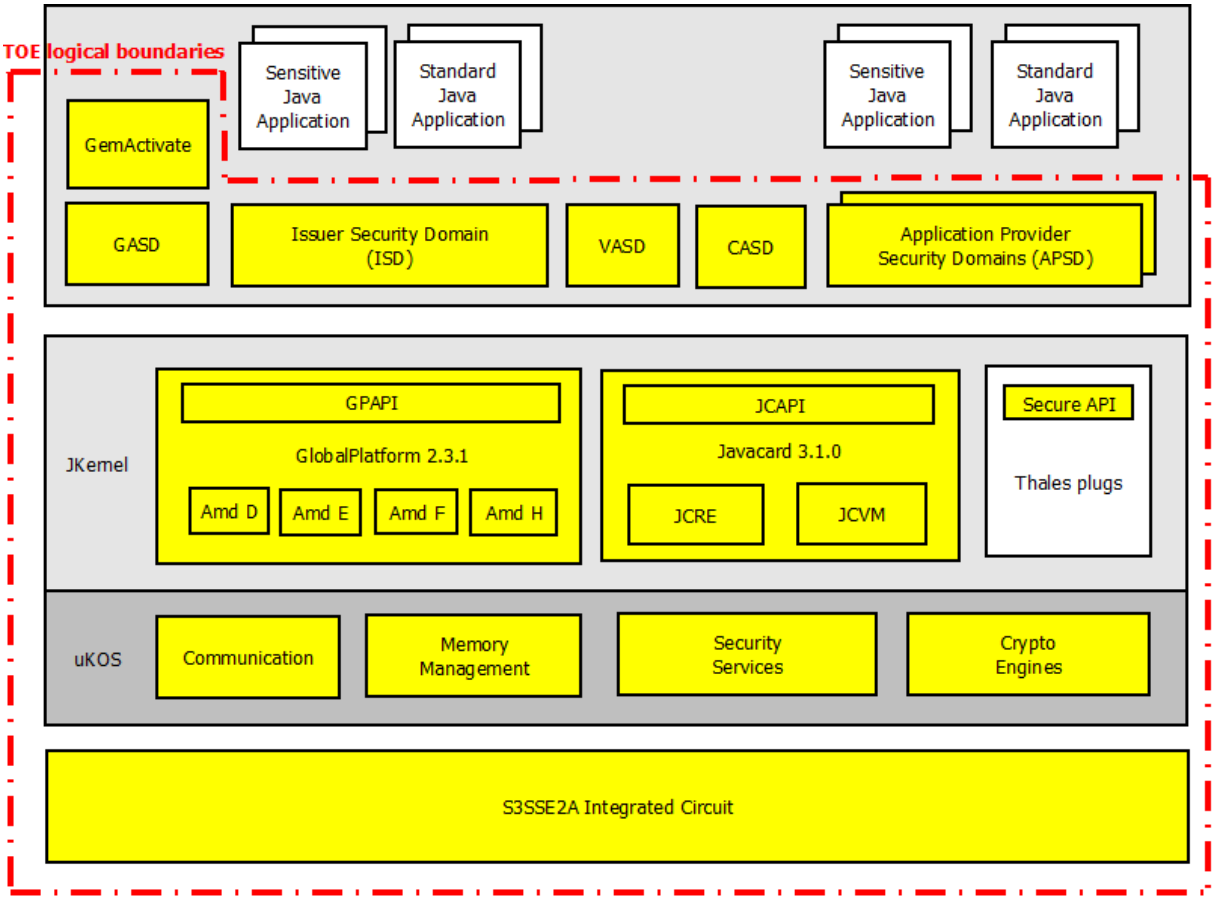

Figure 3: TOE logical boundaries

4.3 TEQS V1.0 PLATFORM DESCRIPTION

The TEQS V1.0 platform implements two major industry standards:

- Oracle's Java Card 3.1.0 [Javacard], which consists of the Java Card 3.1.0 Virtual Machine, Java Card 3.1.0 Runtime Environment and the Java Card 3.1.0 Application Programming Interface.
- Global Platform 2.3.1 [GP], SE Configuration.

It is an opened platform, meaning that additional applications - which may not be known at the time of the present evaluation - can be remotely loaded and installed on the eSE "post issuance", i.e. after the mobile device has been delivered to the end-user. Applications can also be installed "pre-issuance"

during the pre-personalization or personalization phases. Whatever the scenario (pre-issuance or post-issuance), application loading and installation are secured by the GlobalPlatform security mechanisms and verification processes.

The platform implements (at least) the following services:

- Management and control of the communication between the eSE and external entities
- Basic security services as follows:
  - Checking environmental operating conditions using information provided by the IC.
  - Checking life cycle consistency.
  - Providing secure cryptography primitives and algorithms.
  - Ensuring the security of the PIN and cryptographic key objects.
  - Generating random numbers.
  - Handling secure data object and backup mechanisms.
  - Managing memory content.
- Enforcement of the Javacard firewall mechanism.
- Standard Application Programming Interfaces (APIs) such as the Javacard API (JCAPI) and the Global Platform API (GPAPI). TEQS V1.0 also provides the following Java Card System augmentation packages: Sensitive Array, Sensitive Result, Monotonic counters, Cryptographic certificate management, Key Derivation Functions, and System Time.
- Proprietary Thales API: Secure API which provides security services to applications.
- Initialization of the Issuer Security Domain (ISD) and management of the eSE life cycle.
- Creation and management of Supplementary Security Domains (SSD).
- SCP02, SCP03 and SCP11 support.
- Secure loading, installation and deletion of applications within each SD.
- Secure loading of software patches (GemActivate).

Table 1 is instantiated from [GP-PP] and provides a complete view of the mandatory (M) and optional GlobalPlatform features implemented by the TOE (marked as 'Yes' in the table). Accordingly, the three rightmost columns indicate the corresponding selections from [GP-PP]:

- A cross ('X') indicates that the related privilege is covered by the core part of [GP-PP]
- PP packages taken into account for the present evaluation are: DAP, MDAP, DM, CVM, CLFDB, GS
- PP modules taken into account for the present evaluation are: ELFU, OS Update
- PP modules not taken into account for the present evaluation (as the corresponding feature is not supported by the TOE) are: CCCM, CTL, SEMS.

Supported	M	Yes	M	Selections in [PP-GP]		
Privilege	**ISD**	**SSD**	**Application**	**Core**	**Package**	**PP-Module**
Security Domain	M	M	NA	X		
Card Lock	M	Yes	Yes	X		
Card Terminate	M	Yes	Yes	X		
Card Reset	Yes	Yes	Yes	X		
Trusted Path	M	Yes	No	X		
Global Delete	M	Yes	NA	X		
Global Lock	M	Yes	NA	X		
Global Registry	M	Yes	NA	X		
Final Application	Yes	Yes	Yes	X		
Authorized Management (AM)	M	Yes	NA	X		
DAP Verification	No	Yes	NA		DAP	
Mandated DAP Verification	No	Yes	NA		MDAP	
Delegated Management (DM)	NA	Yes	NA		DM	
Token Verification	M	Yes	NA		DM	
Receipt Generation	M	Yes	NA		DM	
CVM Management	Yes	Yes	Yes		CVM	
Contactless Activation	No	No	No			~~CTL~~
Contactless Self Activation	No	No	No			~~CTL~~
Ciphered Load File Data Block (CLFDB)	No	Yes	No		CLFDB	
Global Service (GS) (optional)	No	No	Yes		GS	
						ELFU
						~~CCCM~~
						~~SEMS~~
						OS Update

Table 1: GlobalPlatform privileges and features supported by the TOE

4.4 APPLICATION LAYER DESCRIPTION

Applications can be split in two categories:

- Secure applications: these are sensitive applications, such as e.g. banking applets, whose security is assessed and certified through international schemes (Common Criteria, EMVCo etc.)
- Standard applications, also called “basic” applications: these are the other applications. Although they do not face a formal security evaluation, assurance has to be provided that they do not threaten the sensitive applications and their assets. This assurance is provided through a verification process. Security mechanisms are in place at platform level to ensure that applications which are loaded post issuance have been verified.

4.5 TOE LIFE-CYCLE

The product and TOE life cycle is composed of 7 phases which are described in Table 2. The table also mentions the actor(s) involved in each phase, as well as the associated location(s).

The loading of the TEQS V1.0 software occurs during phase 5, after which the IC loading service is locked and no more available. The TOE delivery point - which determines the boundary between the ALC and AGD Common Criteria assurance classes - is put at the end of phase 5, as illustrated in Figure 4.

As described, at the end of phase 6 Samsung LSI delivers personalized product to the Original Equipment Manufacturer (OEM). At this stage, the TOE is entirely built and protects itself through the security mechanisms implemented in the operating system and the underlying IC.

Notes related to applications development

The basic and secure applets development is part of the product life cycle, but is outside the scope of the present evaluation (since applications are out of the TOE).

The Thales applications will be verified using the evaluated Thales verification process prior to be loaded in Pre-Issuance.

In the same way, but to protect the supplier intellectual property, the applications provided by external Application Providers must be verified and signed by the Verification Authority (VA) prior to be loaded in Pre-Issuance. Application signature will be checked prior to load these applications on the Secure Element in Pre-Issuance.

Note related to patch development

No patch is present within the TOE for the present evaluation. Indeed, should a patch be needed in the future, it would require at least a maintenance of the CC certificate, as required by the CC scheme rules. However the patch mechanism is part of the TOE and as such its security is assessed within the present evaluation.

Phase	Designation	Description / comments		Actor	Location
1	TEQS V1.0 software development	TEQS V1.0 platform development	Platform development & tests	**Thales** DIS MCS R&D team - secure environment -	**Thales La Ciotat** site **Thales Singapore** site
				Thales SL Crypto team - secure environment -	**Thales Singapore** site
		Patch development	Patch development and tests	**Thales** DIS MCS R&D and SL Crypto teams - secure environment -	**Thales La Ciotat** site **Thales Singapore** site
		Basic and secure applets development	Applet development & tests	**Thales or any other accredited Application Provider (AP)** - secure environment -	**Thales Singapore** site **Thales La Ciotat** site or **Application Providers' development sites**
		Industrialization	Production scripts and tools development for phase 5.	**Thales** Product Engineering Team - secure environment -	**Thales Gémenos** site **Thales Singapore** site
			Personalization scripts development for phase 6.	**Thales** CPC team - secure environment -	**Thales Tczew** site
			Personalization Data Generation Secure delivery of TEQS V1.0 platform to Samsung LSI, together with scripts and personalization data.	**Thales** Data Generation team - secure environment -	**Thales Pont-Audemer** site
		IT Support (datacenters, monitoring…)		**Sopra Steria** - secure environment -	**Sopra Steria Noida** site **Sopra Steria Chennai** site
				Telehouse - secure environment -	**Telehouse Paris** site
				Verizon - secure environment -	**Thales Gémenos** site **Thales Calamba** site
2	IC development	Development of the S3SSE2A and associated tools.		**Samsung LSI** - Secure environment -	Development site(s) stated in the S3SSE2A CC certificate
3	IC manufacturing	Manufacturing of virgin S3SSE2A integrated circuits.		**Samsung LSI** - Secure environment -	Manufacturing site(s) stated in the S3SSE2A CC certificate
4	IC packaging	IC packaging & testing.		**Samsung LSI** - Secure environment -	Packaging site(s) stated in the S3SSE2A CC certificate

Phase	Designation	Description / comments	Actor	Location
5	Composite Product integration	Loading of the TEQS V1.0 software within the IC. Pre-personalization and Testing.	**Samsung LSI** - Secure environment -	Production site(s) stated in the S3SSE2A CC certificate
6	Personalization	Personalization and final tests.	**Samsung LSI** - Secure environment -	Personalization site(s) stated in the S3SSE2A CC certificate
7	End-usage	End-usage for the Original Equipment Manufacturer (OEM) and accredited business partners (Application Providers). The OEM, who is the issuer of the TEQS V1.0 product, is responsible for the secure element administration during the end-usage phase and the end of life process. The OEM also grants administration privileges to Application Providers on their respective Security Domains (APSD). Applets may be loaded onto the chip, and OS updates may also be triggered at this stage.	**Original Equipment Manufacturer and accredited business partners (Application Providers)**	Field
		End-usage for mobile phone holder The end-user accesses the OEM related services and performs secure transactions with his mobile phone, thanks to the TEQS V1.0 secure element hosting the sensitive applications and related assets.	**Final User (Mobile phone holder)**	Field

Table 2: Product and TOE life-cycle phases

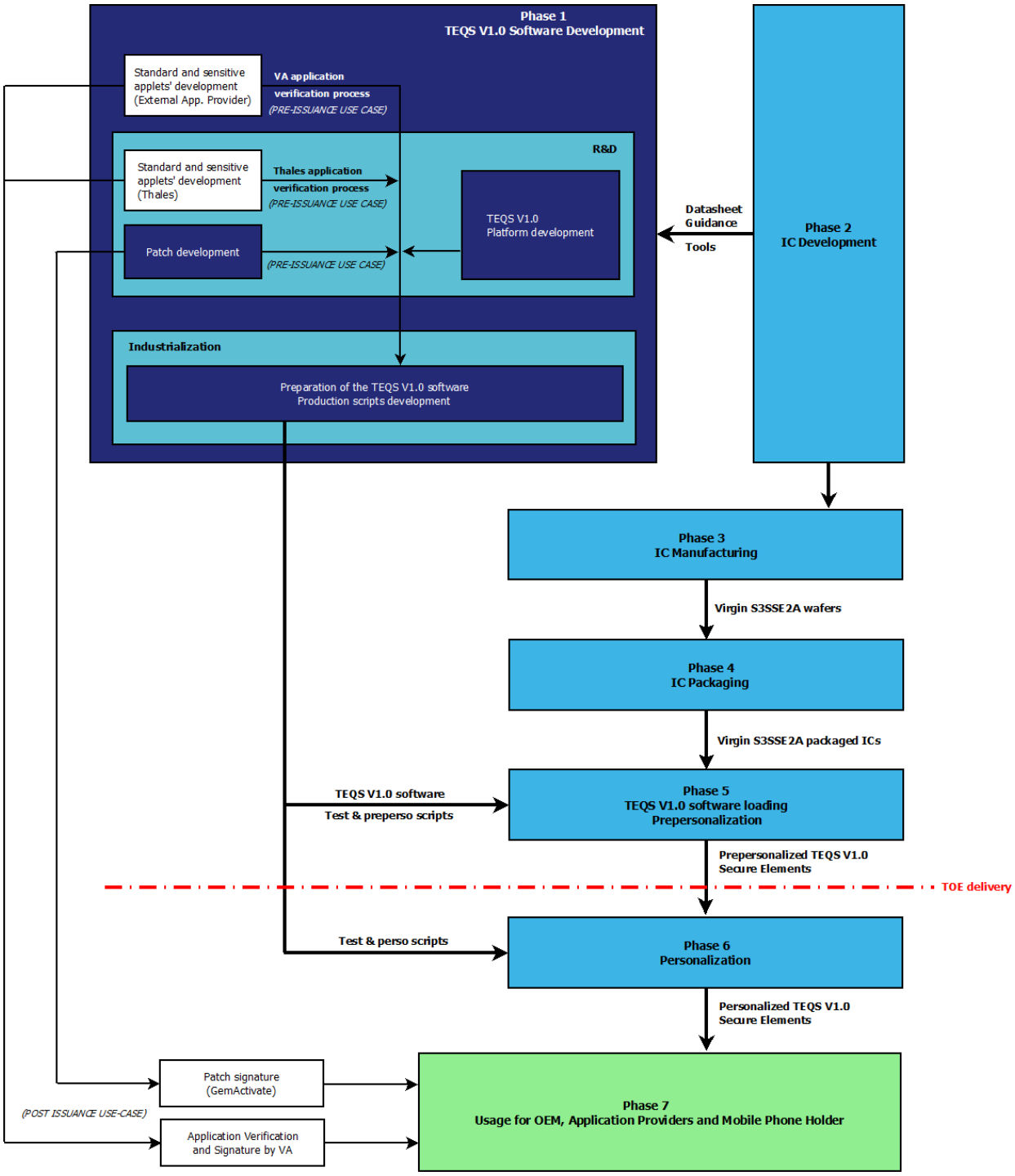

Figure 4: Product and TOE life-cycle

4.6 TOE ACTORS

The actors involved in the TOE life-cycle are listed in table 2.

Note that the following actors are directly represented within the TOE:

- ***The Original Equipment Manufacturer (OEM)***, who is the issuer of the TEQS V1.0 secure element and owner of the TOE. The TOE authorizes the OEM, once authenticated, to manage the loading, instantiation or deletion of applications.
- ***The Application Providers (AP)*** are entities or institutions responsible for their applications and associated services. It may be for example a financial institution (a bank) or a transport operator.
- ***The Controlling Authority (CA)***, optional entity independent from the OEM represented on the TOE and responsible for securing the keys creation and personalization of the Application Provider Security Domains (APSD).
- ***The Verification Authority (VA)***, trusted third party represented on the TOE, acts on behalf of the OEM and is responsible for the verification of applications signatures (DAP) during the loading process. These applications shall be validated for the standard applications or certified for the secure ones.
- ***The GemActivate Administrator*** (usually Thales), represented on the TOE by the GemActivate application and associated keys, is responsible for the remote installation of platform patches (if needed) and the activation of optional platform services on the field (post-issuance).

Note regarding [PP-GP] terminology:

- The term “Issuer” used in [PP-GP] corresponds to the OEM actor. Both names are equivalent in the present ST.
- As described in table 2, the TOE personalization requires two steps:
  - Personalization data generation, performed by Thales Data Generation team.
  - Loading of the personalization data by Samsung LSI during phase 6.

  The term “Personaliser” used in [PP-GP] is present in some OSPs, Assumptions and Objectives for the Operational Environment described in sections 6 and 7. It has to be read in the context of personalization data generation, meaning that the corresponding actor is the Thales Data Generation team mentioned in table 2.

Note regarding applet development:

As mentioned in table 2, applet development can be done by Thales or by Application Providers. This is referred to in some guidance documents as the Application Developer actor.

Note regarding the Final User actor:

As mentioned in table 2, the Final User is the holder of the mobile phone. The terms ‘Final User’, ‘Mobile phone holder’ or ‘End-user’ all designate this same actor.

4.7 TOE DELIVERY

The TEQS V1.0 embedded software is ciphered by Thales Trust Center and delivered from Thales Data Processing Configuration development site (Tczew) to Thales data generation site (Pont-Audemer) via Thales PDM tool.

It is then securely sent from Thales data generation site (Pont-Audemer) to Samsung LSI using Thales Allynis Connect platform (Thales’ secure platform for data transfer with external parties).

Samsung LSI is in charge of the TEQS V1.0 embedded software loading, pre-personalization and personalization in its own premises and proceeds to the delivery of the product directly to customers.
The different guides accompanying the TOE and parts of the TOE are the ones specified in [AGD] section. They are delivered by Thales Technical representative, in form of electronic documents (*.PDF), via secure email (PGP ciphered).

Item type	Item	Reference/Version	Form of delivery
Software	TEQS V1.0	Refer to paragraph §3.2	Enciphered TOE via Allynis Connect (Thales secure transmission tool)
Document	[AGD]	Refer to paragraph §1.2	Electronic document (PDF) via secure email

5 Conformance claims

Common criteria Version: This ST conforms to CC Version 3.1 Revision 5 [CC-1][CC-2][CC-3].

Conformance to CC part 2 and 3:

- This ST is CC part 2 extended with the FCS_RNG.1 and FCS_CKM.5 families. All the other SFRs have been drawn from the catalogue of requirements in CC part 2 [CC-2].
- This ST is CC part 3 conformant. It means that all SARs in that ST are based only upon assurance components in CC part 3 [CC-3].

Assurance package conformance: EAL4 augmented (EAL4+)
This ST conforms to the assurance package EAL4 augmented by ALC_DVS.2 and AVA_VAN.5.

Evaluation type
This is a composite evaluation, which relies on the S3SSE2A chip certificate and evaluation results:

- Certification done under the ANSSI scheme
- Certificate ANSSI-CC-2024/26
- Security Target [ST_IC] strictly conformant to IC Protection Profile [PP/0084]
- Common criteria version: 3.1 Revision 5
- Assurance level: EAL6+ (ASE_TSS.2 augmentation)

Consequently, the composite product evaluation (i.e. the present evaluation) includes the additional composition tasks defined in the CC supporting document “Composite product evaluation for smart cards and similar devices” [CCDB].

Protection Profile (PP) conformance claims:

- This Security Target claims conformance to the [PP-GP] protection profile. As mentioned in section 4.3, in addition to the core part of the PP, the following PP packages and PP modules are taken into account for the present evaluation:
  - PP packages: DAP, MDAP, DM, CLFDB, GS, CVM
  - PP modules: ELFU, OS Update

  The following PP modules are not taken into account for the present evaluation, as the corresponding features are not supported by the TOE: CCCM, CTL, SEMS.

  The conformance type is demonstrable.

  Notes:
  - OE.SCP.IC, OE.SCP.RECOVERY and OE.SCP.SUPPORT from [PP-JCS] have become security objectives for the TOE in the present security target. The reason is that [PP-JCS] considers that the SCP (encompassing the IC and the low-level OS modules) is within the TOE environment. As the TOE considered for the present evaluation includes the SCP, these SCP objectives must be TOE security objectives.
  - The following augmentation packages from [PP-JCS] Appendix 2 are included in the present Security Target document, as the corresponding optional Javacard features are supported by the TOE: Sensitive Array, Sensitive Result, Monotonic Counters, Cryptographic Certificate Management, KDF, System Time.

6 Security problem definition

6.1 ASSETS

6.1.1 [PP-GP] Protection Profile

The following assets are listed in [PP-GP] and shall be considered for the present evaluation.

From core part	
D.ISD_KEYS	Refinement of D.APP_KEYS of [PP-JCS]. ISD cryptographic keys needed to perform card management operations on the card. To be protected from unauthorized disclosure and modification.
D.APSD_KEYS	Refinement of D.APP_KEYS of [PP-JCS]. APSD cryptographic keys needed to establish Secure Channels with the AP. These keys can be used to load and install applications on the card if the Security Domain has the appropriate privileges. To be protected from unauthorized disclosure and modification.
D.CASD_KEYS	Refinement of D.APP_KEYS of [PP-JCS]. CASD cryptographic keys needed to establish Secure Channels with the CA and to decrypt confidential content for APSDs. To be protected from unauthorized disclosure and modification.
D.GP_REGISTRY	The information resource for Card Content management. The GlobalPlatform Registry contains information for managing the card, as well as Executable Load Files, Applications, SD associations, privileges, Identifiers, life cycle states, and memory resource quotas. To be protected from unauthorized modification.
D.GP_CODE	The code of the GlobalPlatform Framework on the card. To be protected from unauthorized modification.
D.TOE_IDENTIFIER	TOE Identification Data to identify the TOE. To be protected from unauthorized modification.
From package ‘Ciphered Load File Data Block (CLFDB)’	
D.CLFDB-DK	Symmetric key to be used to decrypt Load File Data Blocks. To be protected from unauthorized disclosure and modification. Application Note: See [GPCS] section C.1.3.
From package ‘Global Services (GS)’	
D.GS-PARAMETERS	Global Service Parameters are the service family and the service ID within that family. To be protected from unauthorized modification. Application Note: As defined in [GPCS] section 8.1.3. This asset is an extension of D.GP_REGISTRY.
From package ‘Cardholder Verification Method (CVM)’	
D.CVM_PIN	A single global PIN used to authenticate the Cardholder, which can be shared by all the application instances in the card. To be protected from unauthorized modification and disclosure.
D.CVM_MGMT_STATE	The CVM management data include: ▪ CVM value and state (e.g. to determine if the CVM value has been submitted, verified, or blocked) ▪ CVM Retry Limit: The maximum number of presentations of invalid CVM values, until the CVM handler rejects further presentation attempts. ▪ CVM Retry Counter: A counter, used in conjunction with the Retry Limit, to determine when attempts for presenting CVM values shall be rejected. To be protected from unauthorized modification.

From package 'Delegated Management (DM)'	
D.TOKEN-VERIFICATION-KEY	The symmetric key or the public asymmetric key to be used for token verification. To be protected from unauthorized modification and disclosure.
D.RECEIPT-GENERATION-KEY	The symmetric key or the private asymmetric key to be used for receipt generation. To be protected from unauthorized modification and disclosure.
D.CONFIRMATION-DATA	The confirmation Data generated by an SD with the Receipt Generation Privilege. To be protected from unauthorized modification. Application Note: See [GPCS] section 11.1.6.
From package 'DAP Verification'	
D.DAP_BLOCK	Authentication data present in the Load File and generated by an off-card entity (an Application Provider or a Verification Authority). The authentication data contains the SD AID and the Load File Data Block Signature of the Load File Data Block Hash. To be protected from unauthorized modification.
D.APSD_DAP_KEYS	Refinement of D.APP_KEYS of [PP-JCS]. The APSD cryptographic keys which are required for verification of the Load File Block signatures. To be protected from unauthorized disclosure and modification.
From package 'Mandated DAP Verification'	
D.CASD_DAP_KEYS	Refinement of D.APP_KEYS of [PP-JCS]. The CASD cryptographic keys which are required for verification of the Load File Data Block signatures. To be protected from unauthorized disclosure and modification.
From PP-module 'Amendment H: Executable Load File Upgrade (ELFU)'	
D.OLD_ELF	The ELF being upgraded. It is referred to as the "old ELF version". To be protected from unauthorized modification.
D.NEW_ELF	The ELF upgrading the old ELF version. It is referred to as the "new ELF version". To be protected from unauthorized modification.
D.ELF_AID	The ELF AIDs defined in the old and new ELF versions. To be protected from unauthorized modification.
D.ELF_SESSION_ST	The ELF Upgrade Session Status as described in [Amd H] Table 4 8. To be protected from unauthorized modification.
D.ELF_APP_INS	The application instances. To be protected from unauthorized modification and disclosure.
D.ELF_RG_DATA	The registry data including any persistent on-card information related to the application instance which would not be stored or modified by the application instance. To be protected from unauthorized modification.
From PP-module 'OS Update'	
D.OS-UPDATE_SGNVER-KEY	Refinement of D.APP_KEYS. A symmetric cryptographic key, owned by the OS Developer, and used by the TOE to verify the signature of the additional code to be loaded. To be protected from unauthorized disclosure and modification.
D.OS-UPDATE_DEC-KEY	Refinement of D.APP_KEYS. A symmetric cryptographic key, owned by the OS Developer, and used by the TOE to decrypt the additional code to be loaded. To be protected from unauthorized disclosure and modification.
D.OS-UPDATE_ADDITIONAL CODE	The code to be added to the OS after TOE issuance. The additional code has to be signed by the OS Developer. After successful verification of the signature by the Initial TOE, the additional code is loaded and installed through an atomic activation (to create an Updated TOE). To be protected from unauthorized disclosure and modification.
D.OS-UPDATE-CODE-ID	The identification data associated with the additional code. It is loaded and/or updated in the same atomic operation as additional code loading. To be protected from unauthorized modification.

6.1.2 [PP-JCS] Protection Profile

The following assets are listed in [PP-JCS]. According to [PP-GP] they shall also be considered for the present evaluation.

From core part	
D.APP_CODE	The code of the applets and libraries loaded on the card. To be protected from unauthorized modification.
D.APP_C_DATA	Confidentiality - sensitive data of the applications, like the data contained in an object, an array view, a static field, a local variable of the currently executed method, or a position of the operand stack. To be protected from unauthorized disclosure.
D.APP_I_DATA	Integrity sensitive data of the applications, like the data contained in an object, an array view and the PIN security attributes (PIN Try limit, PIN Try counter and State). To be protected from unauthorized modification.
D.APP_KEYS	Cryptographic keys owned by the applets. To be protected from unauthorized disclosure and modification. Note: D.APP_KEYS has been further refined in [PP-GP] as mentioned in section 6.1.1.
D.PIN	Any end-user's PIN. To be protected from unauthorized disclosure and modification.
D.API_DATA	Private data of the API, like the contents of its private fields. To be protected from unauthorized disclosure and modification.
D.CRYPTO	Cryptographic data used in runtime cryptographic computations, like a seed used to generate a key. To be protected from unauthorized disclosure and modification.
D.JCS_CODE	The code of the Java Card System. To be protected from unauthorized disclosure and modification.
D.JCS_DATA	The internal runtime data areas necessary for the execution of the Java Card VM, such as, for instance, the frame stack, the program counter, the class of an object, the length allocated for an array, any pointer used to chain data-structures. To be protected from unauthorized disclosure or modification.
D.SEC_DATA	The runtime security data of the Java Card RE, like, for instance, the AIDs used to identify the installed applets, the currently selected applet, the current context of execution and the owner of each object. To be protected from unauthorized disclosure and modification.

Application Notes:

- The scope of D.APP_I_DATA is widened in view of the Monotonic counters functionality, i.e. the asset described in this section now also addresses and covers monotonic counters.
- The scope of D.APP_I_DATA is widened in view of the Cryptographic Certificate Management functionality, i.e. the asset described in this section now also addresses and covers cryptographic certificates.
- For System Time package, System time is part of D.JCS_DATA therefore the asset to be protected is D.JCS_DATA.

6.2 USERS / SUBJECTS

6.2.1 [PP-GP] and [PP-JCS] Protection Profiles

Subjects are active components of the TOE that (essentially) act on the behalf of users. Users of the TOE include people or institutions (like the AP and the VA), hardware (like the CAD where the card is inserted) and software components (like the application packages installed on the card).

In this Security Target, relevant subjects are those mentioned in [PP-JCS] (i.e. S.ADEL, S.APPLET, S.BCV, S.CAD, S.INSTALLER, S.JCRE, S.JCVM, S.LOCAL, S.MEMBER and S.CAP_FILE)[1] plus the following ones:

S.SD	A GlobalPlatform SD representing an off-card entity on the card. This entity can be the Issuer, an Application Provider, the Controlling Authority, or the Validation Authority.
S.OPEN	It represents the GlobalPlatform Environment (OPEN) on the card. The main responsibility of the S.OPEN is to provide an API to applications, command dispatch, Application selection, (optional) logical channel management, Card Content management, memory management, and Life Cycle management. Note: S.ADEL and S.INSTALLER from [PP-JCS] are parts of S.OPEN.
S.GEMACTIVATE	GemActivate Security Domain representing a Thales administrator on the card. This entity can authorize the activation of optional services and the loading of additional code (i.e. patch) post issuance. Note: this subject corresponds to 'S.OS-DEVELOPER' in the PP-Module 'OS Update' of [PP-GP]. S.GEMACTIVATE and S.OS-DEVELOPER are aliases of the same subject.

6.3 THREATS

6.3.1 [PP-GP] Protection Profile

The following threats are listed in [PP-GP] and shall be considered for the present evaluation.

From core part	
T.UNAUTHORISED-CARD-MGMT	Threat agent: Attacker Adverse action: The attacker performs unauthorised card management operations (for instance impersonates one of the actors represented on the card) in order to take benefit of the privileges or services granted to this actor on the card and perform fraudulent operations: ▪ Load of a package file ▪ Installation of a package file ▪ Extradition of a package file or an applet ▪ Personalisation of an applet or an SD ▪ Deletion of a package file or an applet ▪ Privileges update of an applet or an SD Directly threatened asset(s): D.ISD_KEYS, D.APSD_KEYS, D.APP_C_DATA, D.APP_I_DATA, D.APP_CODE, D.SEC_DATA, D.PIN, and D.GP_REGISTRY (any other asset may be jeopardised should this attack succeed, depending on the virulence of the installed application).

[1] For the description of these [PP-JCS] subjects, see the table at the beginning of section §9.1.3.

T.LIFE-CYCLE	Threat agent: Attacker Adverse action: An attacker accesses an application outside of its expected availability range thus violating irreversible life cycle phases of the application (for instance, an attacker re-personalises the application). Directly threatened asset(s): D.APP_I_DATA, D.APP_C_DATA, and D.GP_REGISTRY.
T.COM-EXPLOIT	Threat agent: Attacker Adverse action: An attacker remotely exploits the communication channels established between a third party and the TOE in order to modify or disclose confidential data. Directly threatened asset(s): All assets are threatened.
T.BRUTE-FORCE-SCP	Threat agent: Attacker Adverse action: APDU commands/API methods can be repeatedly transmitted/invoked to search the entire space of secret values such as cryptographic keys and attempt their brute force extraction. Directly threatened asset(s): All assets are threatened.
From package 'Ciphered Load File Data Block (CLFDB)'	
T.CLFDB-DISC	Threat agent: Attacker Adverse action: The attacker discloses a Ciphered Load File Data Block when it is transmitted to the SE for decryption prior to installation. Directly threatened asset(s): All assets are threatened. Note: This threat refines T.COM-EXPLOIT to address the CLFDB.
From package 'Cardholder Verification Method (CVM)'	
T.CVM-IMPERSONATE	Threat agent: Attacker Adverse action: An attacker could try to impersonate the Cardholder for disclosing or guessing the PIN stored in the CVM, in order to access the services the SE offers. Directly threatened asset(s): D.CVM_PIN
T.CVM-UPDATE	Threat agent: Attacker Adverse action: An attacker could try executing an application that tries to modify (reset/update) the CVM management data (Retry Limit, retry Counter, CVM value and state). Directly threatened asset(s): D.CVM_MGMT_STATE
T.BRUTE-FORCE-CVM	Threat agent: Attacker Adverse action: APDU commands/API methods could be repeatedly transmitted/invoked to attempt the brute force extraction of secrets such as PINs. Directly threatened asset(s): D.CVM_PIN, D.CVM_MGMT_STATE
From package 'Delegated Management (DM)'	
T.RECEIPT	Threat agent: Attacker Adverse action: The attacker may generate fake receipts in order to hide or falsify completion proofs of card management operations. Directly threatened asset(s): D.RECEIPT-GENERATION-KEY,

	D.CONFIRMATION-DATA
T.TOKEN	Threat agent: Attacker Adverse action: The attacker may try to impersonate the Card Manager in order to gain access to the card and perform illegitimate card management operations. Directly threatened asset(s): D.TOKEN-VERIFICATION-KEY
From PP-Module 'Amendment H: Executable Load File Upgrade (ELFU)'	
T.ELF-UNAUTHORISED	Threat agent: Attacker Adverse action: An attacker tries to load an ELF without authorisation. Directly threatened asset(s): T D.OLD_ELF, D.NEW_ELF, D.ELF_AID
T.ELF-VERSION	Threat agent: Attacker Adverse action: An attacker tries to modify the application version in order to prevent the loading of a new ELF. Directly threatened asset(s): T D.OLD_ELF, D.NEW_ELF, D.ELF_AID
T.ELF-DATA-ACCESS	Threat agent: Attacker Adverse action: An attacker tries to access confidential application instance data. Directly threatened asset(s): D.ELF_APP_INS
T.ELF-DATA-INTEGRITY	Threat agent: Attacker Adverse action: An attacker tries to change application instance data. Directly threatened asset(s): D.ELF_APP_INS
T.ELF-SESSION	Threat agent: Attacker Adverse action: An attacker tries to perturb the Session Status to recognize an incomplete upgrade as being complete. Directly threatened asset(s): D.ELF_SESSION_ST
T.ELF-ILL-COMMAND	Threat agent: Attacker Adverse action: An attacker tries to execute forbidden commands during the ELF upgrade session. Directly threatened asset(s): All ELFU PP-Module assets are threatened.
T.ELF-RES-DATA	Threat agent: Attacker Adverse action: An attacker tries to reallocate TOE resources from a user or process to another for gaining unauthorised access to ELF data. Directly threatened asset(s): All ELFU PP-Module assets are threatened.
From PP-Module 'OS Update'	
T.UNAUTHORISED-TOE-CODE-UPDATE	Threat agent: Attacker Adverse action: An attacker loads malicious additional code in order to compromise the security features of the TOE. Directly threatened asset(s): D.OS-UPDATE_ADDITIONALCODE, D.JCS_CODE, D.JCS_DATA.
T.FAKE-SGNVER-KEY	Threat agent: Attacker Adverse action: An attacker modifies the signature verification key used

	by the TOE to verify the signature of the additional code. Hence, the attacker is able to sign and successfully load malicious additional code inside the TOE. Directly threatened asset(s): D.OS-UPDATE_SGNVER-KEY, D.OS UPDATE_ADDITIONALCODE.
T.WRONG-UPDATE-STATE	Threat agent: Attacker Adverse action: An attacker prevents the OS Update operation to be performed atomically, resulting in an inconsistency between the resulting TOE code and the identification data: ▪ The additional code is not loaded within the TOE, but the identification data is updated to mention that the additional code is present. ▪ The additional code is loaded within the TOE, but the identification data is not updated to indicate the change. Directly threatened asset(s): D.OS-UPDATE-CODE-ID.
T.INTEG-OS-UPDATE-LOAD	Threat agent: Attacker Adverse action: The attacker modifies (part of) the additional code when it is transmitted to the TOE for installation. Directly threatened asset(s): D.OS-UPDATE_ADDITIONALCODE, D.JCS_CODE, D.JCS_DATA.
T.CONFID-OS-UPDATE-LOAD	Threat agent: Attacker Adverse action: The attacker discloses (part of) the additional code when it is transmitted to the TOE for installation. Directly threatened asset(s): D.OS-UPDATE_ADDITIONALCODE, D.JCS_CODE, D.JCS_DATA.

6.3.2 [PP-JCS] Protection Profile

According to [PP-GP], the threats listed in [PP-JCS] shall also be considered for the present evaluation. The following table gathers elements extracted from [PP-JCS] which will be referred to in some of the threats mentioned in this section.

#.CONFID-APPLI-DATA	*Application data must be protected against unauthorized disclosure. This concerns logical attacks at runtime in order to gain read access to other application's data.*
#.CONFID-JCS-CODE	*Java Card System code must be protected against unauthorized disclosure. Knowledge of the Java Card System code may allow bypassing the TSF. This concerns logical attacks at runtime in order to gain a read access to executable code, typically by executing an application that tries to read the memory area where a piece of Java Card System code is stored.*
#.CONFID-JCS-DATA	*Java Card System data must be protected against unauthorized disclosure. This concerns logical attacks at runtime in order to gain a read access to Java Card System data. Java Card System data includes the data managed by the Java Card RE, the Java Card VM and the internal data of Java Card platform API classes as well.*
#.INTEG-APPLI-CODE	*Application code must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to the memory zone where executable code is stored. In post-issuance application loading, this threat also concerns the modification of application code in transit to the card.*
#.INTEG-APPLI-DATA	*Application data must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain unauthorized write access to application data. In post-issuance application loading, this threat also concerns the modification of application data contained in a CAP file in transit to the card. For instance, a CAP file contains the values to be used for initializing the static fields of the CAP file.*
#.INTEG-JCS-CODE	*Java Card System code must be protected against unauthorized modification. This*

	concerns logical attacks at runtime in order to gain write access to executable code.
#.INTEG-JCS-DATA	*Java Card System data must be protected against unauthorized modification. This concerns logical attacks at runtime in order to gain write access to Java Card System data. Java Card System data includes the data managed by the Java Card RE, the Java Card VM and the internal data of Java Card API classes as well.*
#.EXE-APPLI-CODE	*Application (byte)code must be protected against unauthorized execution. This concerns (1) invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([JAVASPEC], §6.6); (2) jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code; (3) unauthorized execution of a remote method from the CAD (if the TOE provides JCRMI functionality).*
#.EXE-JCS-CODE	*Java Card System bytecode must be protected against unauthorized execution. Java Card System bytecode includes any code of the Java Card RE or API. This concerns (1) invoking a method outside the scope of the accessibility rules provided by the access modifiers of the Java programming language ([JAVASPEC], §6.6); (2) jumping inside a method fragment or interpreting the contents of a data memory area as if it was executable code. Note that execute access to native code of the Java Card System and applications is the concern of #.NATIVE.*
#.FIREWALL	*The Firewall shall ensure controlled sharing of class instances, and isolation of their data and code between CAP files (that is, controlled execution contexts) as well as between CAP files and the JCRE context. An applet shall not read, write, compare a piece of data belonging to an applet that is not in the same context, or execute one of the methods of an applet in another context without its authorization.*
#.NATIVE	*Because the execution of native code is outside of the JCS TSF scope, it must be secured so as to not provide ways to bypass the TSFs of the JCS. Loading of native code, which is as well outside those TSFs, is submitted to the same requirements. Should native software be privileged in this respect, exceptions to the policies must include a rationale for the new security framework they introduce.*
#.VERIFICATION	*Bytecode must be verified prior to being executed. Bytecode verification includes (1) how well-formed CAP file is and the verification of the typing constraints on the bytecode, (2) binary compatibility with installed CAP files and the assurance that the export files used to check the CAP file correspond to those that will be present on the card when loading occurs.*
#.INSTALL	*(1) The TOE must be able to return to a safe and consistent state when the installation of a CAP file or an applet fails or be cancelled (whatever the reasons). (2) Installing an applet must have no effect on the code and data of already installed applets. The installation procedure should not be used to bypass the TSFs. In short, it is an atomic operation, free of harmful effects on the state of the other applets. (3) The procedure of loading and installing a CAP file shall ensure its integrity and authenticity. In case of Extended CAP files, installation of a CAP shall ensure installation of all the packages in the CAP file.*
#.SID	*(1) Users and subjects of the TOE must be identified. (2) The identity of sensitive users and subjects associated with administrative and privileged roles must be particularly protected; this concerns the Java Card RE, the applets registered on the card, and especially the default applet and the currently selected applet (and all other active applets in Java Card System 2.2.x). A change of identity, especially standing for an administrative role (like an applet impersonating the Java Card RE), is a severe violation of the Security Functional Requirements (SFR). Selection controls the access to any data exchange between the TOE and the CAD and therefore, must be protected as well. The loading of a CAP file or any exchange of data through the APDU buffer (which can be accessed by any applet) can lead to disclosure of keys, application code or data, and so on.*
#.OBJ-DELETION	*(1) Deallocation of objects should not introduce security holes in the form of references pointing to memory zones that are not longer in use, or have been reused for other purposes. Deletion of collection of objects should not be maliciously used to circumvent the TSFs. (2) Erasure, if deemed successful, shall ensure that the deleted class instance is no longer accessible.*
#.DELETION	*(1) Deletion of installed applets (or CAP files) should not introduce security holes in the form of broken references to garbage collected code or data, nor should they alter integrity or confidentiality of remaining applets. The deletion procedure should not be maliciously used to bypass the TSFs. (2) Erasure, if deemed successful, shall ensure that any data owned by the deleted applet is no longer accessible (shared objects shall either prevent deletion or be made inaccessible). A deleted applet cannot be selected or receive APDU commands. CAP file deletion shall make the code of the CAP file is no longer available for execution. In case of Extended CAP files, deletion of a CAP shall ensure that code and data for all the packages in the CAP file is no longer available for execution. (3) Power failure or other failures during the process shall be taken into account in the implementation so as to preserve the SFRs. This does not mandate, however, the*

	process to be atomic. For instance, an interrupted deletion may result in the loss of user data, as long as it does not violate the SFRs.
#.RESOURCES	*The TOE controls the availability of resources for the applications in order to prevent unauthorized denial of service or malfunction of the TSFs. This concerns both execution (dynamic memory allocation) and installation (static memory allocation) of applications and CAP files.*
#.INTEG-APPLI-DATA-PHYS	*Integrity-sensitive application data must be protected against unauthorized modification by physical attacks.*

Application note: Specific threats against System Time refer to security aspect #.INTEG-JCS-DATA.

The following threats are derived from the here-above security aspects:

T.CONFID-APPLI-DATA	The attacker executes an application to disclose data belonging to another application. See #.CONFID-APPLI-DATA for details. Directly threatened asset(s): D.APP_C_DATA, D.PIN and D.APP_KEYs.
T.CONFID-JCS-CODE	The attacker executes an application to disclose the Java Card System code. See #.CONFID-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.
T.CONFID-JCS-DATA	The attacker executes an application to disclose data belonging to the Java Card System. See #.CONFID-JCS-DATA for details. Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.
T.INTEG-APPLI-CODE	The attacker executes an application to alter (part of) its own code or another application's code. See #.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.
T.INTEG-APPLI-CODE.LOAD	The attacker modifies (part of) its own or another application code when an application CAP file is transmitted to the card for installation. See #.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.
T.INTEG-APPLI-DATA	The attacker executes an application to alter (part of) another application's data. See #.INTEG-APPLI-DATA for details. Directly threatened asset(s): D.APP_I_DATA, D.PIN, and D.APP_KEYs.
T.INTEG-APPLI-DATA.LOAD	The attacker modifies (part of) the initialization data contained in an application CAP file when the CAP file is transmitted to the card for installation. See #.INTEG-APPLI-DATA for details. Directly threatened asset(s): D.APP_I_DATA and D_APP_KEY.
T.INTEG-JCS-CODE	The attacker executes an application to alter (part of) the Java Card System code. See #.INTEG-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.
T.INTEG-JCS-DATA	The attacker executes an application to alter (part of) Java Card System or API data. See #.INTEG-JCS-DATA for details. Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.

Other attacks are in general related to one of the above, and aimed at disclosing or modifying on-card information. Nevertheless, they vary greatly on the employed means and threatened assets, and are thus covered by quite different objectives in the sequel. That is why a more detailed list is given hereafter.

T.SID.1	An applet impersonates another application, or even the Java Card RE, in order to gain illegal access to some resources of the card or with respect to the end user or the terminal. See #.SID for details.

	Directly threatened asset(s): D.SEC_DATA (other assets may be jeopardized should this attack succeed, for instance, if the identity of the JCRE is usurped), D.PIN and D.APP_KEYs.
T.SID.2	The attacker modifies the TOE's attribution of a privileged role (e.g. default applet and currently selected applet), which allows illegal impersonation of this role. See #.SID for further details. Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on whose identity was forged).
T.EXE-CODE.1	An applet performs an unauthorized execution of a method. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.
T.EXE-CODE.2	An applet performs an execution of a method fragment or arbitrary data. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.
T.NATIVE	An applet executes a native method to bypass a TOE Security Function such as the firewall. See #.NATIVE for details. Directly threatened asset(s): D.JCS_DATA.
T.RESOURCES	An attacker prevents correct operation of the Java Card System through consumption of some resources of the card: RAM or NVRAM. See #.RESOURCES for details. Directly threatened asset(s): D.JCS_DATA.
T.DELETION	The attacker deletes an applet or a CAP file already in use on the card, or uses the deletion functions to pave the way for further attacks (putting the TOE in an insecure state). See #.DELETION for details). Directly threatened asset(s): D.SEC_DATA and D.APP_CODE. Note: T.DELETION is a sub-threat of the T.UNAUTHORISED-CARD-MGMT threat mentioned in [PP-GP] and listed in section 6.3.1.
T.INSTALL	The attacker fraudulently installs post-issuance of an applet on the card. This concerns either the installation of an unverified applet or an attempt to induce a malfunction in the TOE through the installation process. See #.INSTALL for details. Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on the virulence of the installed application). Note: T.INSTALL is a sub-threat of the T.UNAUTHORISED-CARD-MGMT threat mentioned in [PP-GP] and listed in section 6.3.1.
T.OBJ-DELETION	The attacker keeps a reference to a garbage collected object in order to force the TOE to execute an unavailable method, to make it to crash, or to gain access to a memory containing data that is now being used by another application. See #.OBJ-DELETION for further details. Directly threatened asset(s): D.APP_C_DATA, D.APP_I_DATA and D.APP_KEYs.
T.PHYSICAL	The attacker discloses or modifies the design of the TOE, its sensitive data or application code by physical (opposed to logical) tampering means. This threat includes IC failure analysis, electrical probing, unexpected tearing, and DPA. That also includes the modification of the runtime execution of Java Card System or SCP software through alteration of the intended execution order of (set of) instructions through physical tampering techniques. This threatens all the identified assets. Application note: as sensitive array and sensitive result are supported by the TOE, this threat also covers the following sub-threat exploiting specifically the listed assets below: ▪ The attacker performs a physical manipulation to alter (part of) an application's integrity-sensitive data. See #.INTEG-APPLI-DATA-PHYS for details. ▪ Directly threatened asset(s): D.APP_I_DATA, D.PIN, and D.APP_KEYs.

6.4 ORGANISATIONAL SECURITY POLICIES

6.4.1 [PP-GP] Protection Profile

The following OSP are listed in [PP-GP] and shall be considered for the present evaluation.

From core part	
OSP.AID-MANAGEMENT	When loading an application that uses shareable object interface, to make its services available to other applications, the VA shall verify that the AID of the application being loaded does not impersonate the AID known by another application on the card for the use of shareable services.
OSP.LOADING	Application code, validated or certified depending on the application, is loaded onto the SE Platform using any kind of CCM servers (e.g. OTA or other kinds of servers used to perform card content management) and protocols with contactless or contact (e.g. USB) connectivity. If needed, the Issuer can pre authorize content loading operation through delegated management privilege to an individual on-card representative of APs. In that case the application code is loaded in the APSD. Once loaded, the application is personalized using the appropriate SD keys.
OSP.SERVERS	A security policy shall be employed by the Issuer to ensure the security of the applications stored on its CCM servers (e.g. OTA or other kinds of servers used to perform card content management).
OSP.APSD-KEYS	The APSD keys personalization can rely either on the key escrow if the APSD has been created before the usage phase of the SE card, or on the CA if the APSD has been created during the usage phase. In the first case, the APSD keys are generated and stored in a secure way by the personalizer. Then, these keys are transmitted to the AP, via the key escrow. In the second case, one of the following must occur: ▪ The APSD keys are generated and stored in a secure way by the APSD, then securely transmitted to the AP using the CASD. ▪ Or the APSD keys are created by the AP and securely transferred to the APSD using the CASD.
OSP.ISD-KEYS	The security of the ISD keys shall be ensured by a well-defined security policy that covers generation, storage, distribution, destruction, and recovery. This policy is enforced by the Issuer in collaboration with the personaliser.
OSP.KEY-GENERATION	The personaliser shall enforce a policy ensuring that generated keys cannot be accessed in plaintext.
OSP.CASD-KEYS	The CASD keys shall be securely generated and stored in the SE card during the personalization process. These keys are not modifiable after card issuance.
OSP.KEY-CHANGE	The AP shall change its initial keys before any operation on its APSD.
OSP.SECURITY-DOMAINS	SDs can be dynamically created, deleted, and blocked during usage phase, i.e. post issuance.
OSP.APPLICATIONS	The applications intending to be used with the TOE shall follow the TOE's security guidance and recommendations.
From package 'Ciphered Load File Data Block (CLFDB)'	
OSP.CLFDB-ENC-PR	The Load File Data Block must be encrypted securely by a trusted SD provider. Application Note: See [GPCS] section C.6.

From package 'Delegated Management (DM)'	
OSP.TOKEN-GEN	The Token must be generated securely by a trusted entity according to the signature algorithms defined in GlobalPlatform specifications. Application Note: See [GPCS] sections B.1, B.2, B.3, B.4, and C.4.
OSP.RECEIPT-VER	The Receipt must be verified securely by a trusted entity according to the methods defined in GlobalPlatform specifications. Application Note: See [GPCS] sections B.1, B.2, B.3, B.4, and C.5.
From packages 'DAP Verification' and 'Mandated DAP Verification'	
OSP.DAP_BLOCK_GEN	The DAP Block must be generated securely by a trusted entity that verifies the content of the Load File Data Block linked to the hash.
From PP-Module 'Amendment H: Executable Load File Upgrade (ELFU)'	
OSP.ELF_DELE_OP	The TOE shall provide the possibility to perform the deletion operation of the Application instances and ELF(s) in one transaction, so that either a full operation or no operation can occur (atomic and irreversible operation).
From PP-Module 'OS Update'	
OSP.ATOMIC_ACTIVATION	Additional code has to be loaded and installed on the Initial TOE through an atomic activation to create the Updated TOE. Each additional code shall be identified with unique Identification Data. During such atomic activation, identification Data of the Initial TOE have to be updated to clearly identify the Updated TOE. In case of interruption or incident during activation, the TOE shall remain in its initial state or fail secure.
OSP.TOE_IDENTIFICATION	Identification Data of the resulting Updated TOE shall identify the Initial TOE and the activated additional code. Identification Data shall be protected in integrity.
OSP.ADDITIONAL_CODE_SIGNING	The additional code has to be signed with a cryptographic key according to relevant standards, and the generated signature is associated with the additional code. The additional code signature must be verified during loading to assure its authenticity and integrity and to assure that loading is authorized on the TOE. The cryptographic key used to sign the additional code shall be of sufficient quality and its generation shall be appropriately secured to ensure the authenticity, integrity, and confidentiality of the key.
OSP.ADDITIONAL_CODE_ENCRYPTION	The additional code has to be encrypted according to the relevant standard in order to ensure its confidentiality when it is transmitted to the TOE for loading and installation. The encryption key shall be of sufficient quality and its generation shall be appropriately secured to ensure the confidentiality, authenticity, and integrity of the key.

6.4.2 [PP-JCS] Protection Profile

According to [PP-GP], the OSP listed in [PP-JCS] shall also be considered for the present evaluation.

OSP.VERIFICATION	This policy shall ensure the consistency between the export files used in the verification and those used for installing the verified file. The policy must also ensure that no modification of the file is performed in between its verification and the signing by the verification authority. See #.VERIFICATION for details. If the application development guidance provided by the platform developer contains recommendations related to the isolation property of the platform, this policy shall also ensure that the verification authority checks that these recommendations are applied in the application code.

6.5 SECURE USAGE ASSUMPTIONS

6.5.1 [PP-GP] Protection Profile

The following assumptions are listed in [PP-GP] and shall be considered for the present evaluation.

From core part	
A.ISSUER	This is the entity that owns the SE and is ultimately responsible for the behavior of the SE.
A.ADMIN	These administrators of the CCM servers (e.g. OTA or other kinds of servers) used to perform card content management are trusted actors. They are trained to use and administrate those servers securely. They have the means and the equipment to perform their tasks. They are aware of the sensitivity of the assets they manage and the responsibilities associated with the administration of CCM servers. Administrators obey the security policies and constitute, by this assumption, no source of an inside attack.
A.APPS-PROVIDER	The AP is a trusted actor that provides applications. APs are responsible for their APSD keys.
A.VERIFICATION-AUTHORITY	The VA is a trusted actor with the capability to check and validate the digital signature of an application.
A.KEY-ESCROW	The key escrow is a trusted actor in charge of the secure storage of the initial APSD keys generated by the TOE personaliser during the initial personalisation.
A.PERSONALISER	The personaliser is in charge of the TOE personalisation process, which ensures the security of the keys loaded in the SE: ▪ Issuer Security Domain keys (ISD keys) ▪ Application Provider Security Domains keys (APSD keys) ▪ Controlling Authority Security Domain keys (CASD keys)
A.CONTROLLING-AUTHORITY	The CA is a trusted actor different from the issuer responsible for the CASD keys and associated services.
A.PRODUCTION	Security procedures are used after TOE Delivery up to delivery to the end consumer to maintain the confidentiality and integrity of the TOE and its data (to prevent any possible copy, modification, retention, theft, or unauthorised use).
A.SCP-SUPP	The operational environment supports and uses the SCPs offered by the TOE.
A.KEYS-PROT	The keys stored outside the TOE and applied for secure communication and authentication between the SE and the external entities are confidentiality and integrity protected in their storage environment. This covers D.APSD_KEYS and D.ISD_KEYS.
From PP-Module 'OS Update'	
A.OS-UPDATE-EVIDENCE	For additional code loaded pre-issuance, it is assumed that evaluated technical and/or audited organisational measures have been implemented to ensure that the additional code: 1. has been issued by the genuine OS Developer 2. has not been altered since it was issued by the genuine OS Developer.

	For additional code loaded post-issuance, it is assumed that the OS Developer provides digital evidence to the TOE in order to prove the following: 1. he is the genuine developer of the additional code and 2. the additional code has not been modified since it was issued by the genuine OS Developer.
A.SECURE_ACODE_MANAGEMENT	It is assumed that: ▪ The Key management process related to the OS Update capability takes place in a secure and audited environment. ▪ The cryptographic keys used by the cryptographic operations are of strong quality and appropriately secured to ensure confidentiality, authenticity, and integrity of those keys.

6.5.2 [PP-JCS] Protection Profile

The following assumptions from [PP-JCS] shall also be considered for the present evaluation.

A.CAP_FILE	CAP Files loaded post-issuance do not contain native methods. The Java Card specification explicitly "does not include support for native methods" ([JCVM310], §3.3) outside the API.
A.VERIFICATION	All the bytecodes are verified at least once, before the loading, before the installation or before the execution, depending on the card capabilities, in order to ensure that each bytecode is valid at execution time.

6.6 COMPOSITION TASKS – SECURITY PROBLEM DEFINITION PART

6.6.1 Statement of Compatibility – Threats part

The following table (see next page) lists the relevant threats of the security target [ST_IC], and provides the link to the threats on the composite-product, showing that there is no contradiction between the two.

IC relevant threat label	IC relevant threat title	IC relevant threat content	Link to the composite-product threats
T.Leak-Inherent	Inherent Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential user data as part of the assets.	T.PHYSICAL
T.Phys-Probing	Physical Probing	An attacker may perform physical probing of the TOE in order (i) to disclose user data while stored in protected memory areas, (ii) to disclose/reconstruct the user data while processed or (iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.	T.PHYSICAL
T.Malfunction	Malfunction due to Environmental Stress	An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to (i) modify security services of the TOE or (ii) modify functions of the Security IC Embedded Software (iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software. This may be achieved by operating the Security IC outside the normal operating conditions.	T.PHYSICAL
T.Phys-Manipulation	Physical Manipulation	An attacker may physically modify the Security IC in order to (i) modify user data of the Composite TOE, (ii) modify the Security IC Embedded Software, (iii) modify or deactivate security services of the TOE, or (iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.	T.PHYSICAL
T.Leak-Forced	Forced Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential user data of the Composite TOE as part of the assets even if the information leakage is not inherent but caused by the attacker.	T.PHYSICAL
T.Abuse-Func	Abuse of Functionality	An attacker may use functions of the TOE which may not be used after TOE Delivery in order to (i) disclose or manipulate user data of the Composite TOE, (ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or (iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or (iv) enable an attack disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.	T.LIFE-CYCLE
T.RND	Deficiency of Random Numbers	An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.	Analysis of the composite-product threats does not reveal any contradiction with this IC threat.
T.Masquerade_TOE	Masquerade the TOE	An attacker may threaten the property being a genuine TOE by producing a chip which is not a genuine TOE but wrongly identifying itself as genuine TOE sample.	Analysis of the composite-product threats does not reveal any

IC relevant threat label	IC relevant threat title	IC relevant threat content	Link to the composite-product threats
			contradiction with this IC threat.
T.Mem-Access	Memory Access Violation	Parts of the IC Smartcard Embedded Software may cause security violations by accidentally or deliberately accessing restricted data (which may include code). Any restrictions are defined by the security policy of the specific application context and must be implemented by the Smartcard IC Embedded Software.	T.CONFID-APPLI-DATA T.CONFID-JCS-DATA T.INTEG-APPLI-DATA T.INTEG-JCS-DATA T.SID.1 T.SID.2 T.EXE-CODE.1
T.Open_Samples_Diffusion	Diffusion of open samples	An attacker may get access to open samples of the TOE and use them to gain information about the TSF (loader, memory management unit, ROM code…). He may also use the open samples to characterize the behavior of the IC and its security functionalities (for example: characterization of side channel profiles, perturbation cartography…). The execution of a dedicated security features (for example: execution of a DES computation without countermeasures or by de-activating countermeasures) through the loading of an adequate code would allow this kind of characterization and the execution of enhanced attacks on the IC.	T.PHYSICAL

6.6.2 *Statement of compatibility – OSPs part*

The following table lists the relevant OSPs of the security target [ST_IC], and provides the link to the OSPs related to the composite-product, showing that there is no contradiction between the two.

IC OSP label	IC OSP content	Link to the composite product
P.Process-TOE	Identification during TOE Development and Production: an accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.	No contradiction with the present evaluation; the chip traceability information participates to the composite TOE identification.
P.Crypto-Service	Cryptographic Services provided by the TOE: The TOE shall provide the following cryptographic services to the IC Embedded Software: - Triple Data Encryption Standard (TDES) - Advanced Encryption Standard (AES)	The TDES and AES hardware accelerators are used by the composite TOE cryptographic library, to provide respectively TDES and AES encryption and decryption.
P.Lim_Block_Loader	Limiting and Blocking the Loader functionality: The composite manufacturer uses the Loader for loading of Security IC Embedded Software, user data of the Composite Product or IC Dedicated Support Software in charge of the IC Manufacturer. He limits the capability and blocks the availability of the Loader in order to protect stored data from disclosure and manipulation.	As mentioned in section §4.5, the TEQS V1.0 software is loaded during phase 5 of the composite TOE life cycle, after which the IC loading service is locked and no more available.
P.Ctlr_Loader	Controlled usage to Loader Functionality: authorized user controls the usage of the Loader functionality in order to protect stored and loaded user data from disclosure and manipulation.	As mentioned in section §4.5, the TEQS V1.0 software is loaded during phase 5 of the composite TOE life cycle. Access to the Loader is done in a secured environment, under Samsung LSI authority, and is conditioned by a successful authentication.

6.6.3 *Statement of compatibility – Assumptions part*

The following table (see next page) lists the relevant assumptions of the security target [ST_IC], and provides the link to the assumptions related to the composite-product, showing that there is no contradiction between the two.

IC assumption label	IC assumption title	IC assumption content	IrPA	CfPA	SgPA	Link to the composite product
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 unauthorized use).		X	X	• During phases 4 and 5: CfPA Fulfilled by the ALC composite-SARs • During phases 6 and 7: SgPA A.PRODUCTION.
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.		X		O.KEY-MNGT O.PIN-MNGT
A.Key-Function	Usage of Key-dependent Functions	Key-dependent functions (if any) shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced).		X		O.SCP.IC O.SCP.SUPPORT O.CIPHER O.KEY-MNGT

7 Security objectives

7.1 SECURITY OBJECTIVES FOR THE TOE

7.1.1 [PP-GP] Protection Profile

The following TOE security objectives are listed in [PP-GP] and shall be considered for the present evaluation.

From core part	
O.CARD-MANAGEMENT	The TOE shall provide the card manager as defined in [GPCS]. The card manager shall control the access to card management functions such as the installation, update, or deletion of applets. It shall also implement the Issuer's policy on the card. The card manager is an application with specific rights (e.g. ISD), which is responsible for the administration of the SE. Typically, the card manager shall be in charge of the life cycle of the whole card, as well as that of the installed applications (applets). The card manager shall prevent card content management operations (loading, installation, deletion) from being carried out, for instance, at invalid states of the card or by unauthorised actors. It shall also enforce security policies established by the Issuer.
O.DOMAIN-RIGHTS	The Issuer shall not access or change personalised APSD keys, which belong exclusively to the AP. Modification of an SD key set is restricted to the AP owning the SD.
O.APPLI-AUTH	The card manager shall enforce the application security policies established by the Issuer. The enforcement shall be implemented by requiring application authentication during application loading on the card.
O.SECURITY-DOMAINS	SDs can be dynamically created, deleted, and blocked during the end use phase.
O.COMM-AUTH	The TOE shall authenticate the origin of the card management requests received by the card, and authenticate itself to the remote actor.
O.COMM-INTEGRITY	The TOE shall verify the integrity of the (card management) requests that the card receives.
O.COMM-CONFIDENTIALITY	The TOE shall be able to process card management requests containing encrypted data.
O.NO-KEY-REUSE	The TOE shall ensure that session keys can be used only once.
O.PRIVILEGES-MANAGEMENT	The TOE shall provide Privileges assignment and management functionalities for the on-card entities ISD, SSD, and Applications. The TOE shall control the access to the Privileges assignment and management functions.
O.LC-MANAGEMENT	The TOE shall provide a state machine that enforces the TOE's life cycle, keeps track of the TOE's current state, and controls that the operations required by the users are consistent with the current life cycle state of the TOE. The TOE shall provide Life Cycle (LC) management functionalities for the Card, ELFs, SDs, and Applications.

From package 'Ciphered Load File Data Block (CLFDB)'	
O.CLFDB-DECIPHER	If the SD to be associated with the Executable Load File has the Ciphered Load File Data Block privilege, then the card shall support encryption schemes as defined by GlobalPlatform specifications and the SD shall be able to decipher the Ciphered Load File Data Blocks. *Application Note*: See [GPCS] section C.6.
From package 'Cardholder Verification Method (CVM)'	
O.GLOBAL-CVM	The TOE shall restrict the modification of the security attributes of the CVM only to defined privileged applications appointed by the Card Manager. Any SD allowed to perform CVM can grant the CVM privilege to an Application.
O.CVM-BLOCK	If the maximum number of attempts has been reached, further Cardholder authentication attempts are blocked. The blocking can be removed by special action of the Card Manager or a privileged user.
O.CVM-MGMT	The TOE shall provide means to securely manage CVM objects. Secure management of CVM objects includes: • Atomic update of PIN code and of the try counter, • No rollback of the number of unsuccessful authentication attempts, • Protection of confidentiality of the PIN value, • Protection of the PIN comparison process against observation.
From package 'Delegated Management (DM)'	
O.RECEIPT	The TOE shall generate non-repudiable receipts of the completion of card management operations. The generation of the receipt shall be performed by an SD with 'Receipt Generation' Privilege.
O.TOKEN	The TOE shall verify tokens during the processing of card management operations. The verification of the token shall be performed by an SD with 'Token Verification' Privilege.
From PP-Module 'Amendment H: Executable Load File Upgrade (ELFU)'	
O.ELF_AUTHORISED	Only authorised entities shall be able to load ELFs.
O.ELF_INTEGRITY	The ELF integrity shall be preserved during the loading process – (confidentiality maintained if required).
O.ELF_APP_DATA	The application instance data shall be securely stored when saved. The OPEN shall maintain the integrity & consistency of Registry data.
O.ELF_SESSION	The session status shall be consistent throughout the upgrade process. Forbidden commands shall be rejected during the upgrade process.
O.ELF_DELE_IRR	The TOE must be able to provide an atomic and irreversible deletion operation of the Application instances and ELF(s).
O.ELF_DATA_PRO	The TOE must ensure that any ELF information contained in a protected resource is not inappropriately disclosed when the resource is reallocated.
From PP-Module 'OS Update'	
O.SECURE_LOAD_ACODE	The TOE shall check an evidence of authenticity and integrity of the additional code to be loaded. The TOE enforces that only an allowed version of the additional code can be loaded. The TOE shall forbid the loading of an additional code not intended to be

	assembled with the TOE. During the loading of the additional code, the TOE shall remain secure.
O.SECURE_AC_ACTIVATION	Activation of the additional code and update of the Identification Data shall be performed at the same time in an atomic way. All the operations needed for the code to be able to operate as in the Updated TOE shall be completed before activation. If the atomic activation is successful, then the resulting product is the Updated TOE, otherwise (in case of interruption or incident which prevents the forming of the Updated TOE), the TOE shall preserve a secure state.
O.TOE_IDENTIFICATION	The TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data. After atomic activation of the additional code, the Identification Data of the Updated TOE allows identifications of both the Initial TOE and additional code. The user must be able to uniquely identify Initial TOE and additional code(s) which are embedded in the Updated TOE.
O.CONFID-OS-UPDATE.LOAD	The TOE shall decrypt the additional code prior installation. *Application Note*: Confidentiality protection must be enforced when the additional code is transmitted to the TOE for loading (See OE.OS-UPDATE-ENCRYPTION later in this table). Confidentiality protection can be achieved either through direct encryption of the additional code, or by means of a trusted path ensuring the confidentiality of the communication to the TOE.

7.1.2 [PP-JCS] Protection Profile

The following TOE security objectives from [PP-JCS] shall also be considered for the present evaluation.

From core part	
O.SID	The TOE shall uniquely identify every subject (applet, or CAP file) before granting it access to any service.
O.FIREWALL	The TOE shall ensure controlled sharing of data containers owned by applets of different CAP files or the JCRE and between applets and the TSFs. See #.FIREWALL for details.
O.GLOBAL_ARRAYS_CONFID	The TOE shall ensure that the APDU buffer that is shared by all applications is always cleared upon applet selection. The TOE shall ensure that the global byte array used for the invocation of the install method of the selected applet is always cleared after the return from the install method.
O.GLOBAL_ARRAYS_INTEG	The TOE shall ensure that no application can store a reference to the APDU buffer, a global byte array created by the user through makeGlobalArray method and the byte array used for invocation of the install method of the selected applet.
O.ARRAY_VIEWS_CONFID	The TOE shall ensure that no application can read elements of an array view not having array view security attribute ATTR_READABLE_VIEW. The TOE shall ensure that an application can only read the elements of the array view within the bounds of the array view.
O.ARRAY_VIEWS_INTEG	The TOE shall ensure that no application can write to an array view not having array view security attribute ATTR_WRITABLE_VIEW. The TOE shall ensure that an application can only write within the bounds of the array view.
O.NATIVE	The only means that the Java Card VM shall provide for an application to execute

	native code is the invocation of a method of the Java Card API, or any additional API. See #.NATIVE for details.
O.OPERATE	The TOE must ensure continued correct operation of its security functions. See #.OPERATE for details.
O.REALLOCATION	The TOE shall ensure that the re-allocation of a memory block for the runtime areas of the Java Card VM does not disclose any information that was previously stored in that block.
O.RESOURCES	The TOE shall control the availability of resources for the applications. See #.RESOURCES for details.
O.ALARM	The TOE shall provide appropriate feedback information upon detection of a potential security violation. See #.ALARM for details.
O.CIPHER	The TOE shall provide a means to cipher sensitive data for applications in a secure way. In particular, the TOE must support cryptographic algorithms consistent with cryptographic usage policies and standards. See #.CIPHER for details.
O.RNG	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.
O.KEY-MNGT	The TOE shall provide a means to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys. See #.KEY-MNGT.
O.PIN-MNGT	The TOE shall provide a means to securely manage PIN objects (including the PIN try limit, PIN try counter and states). If the PIN try limit is reached, no further PIN authentication must be allowed. See #.PIN-MNGT for details. Application Note: PIN objects may play key roles in the security architecture of client applications. The way they are stored and managed in the memory of the smart card must be carefully considered, and this applies to the whole object rather than the sole value of the PIN. For instance, the try limit and the try counter's value are as sensitive as that of the PIN and the TOE must restrict their modification only to authorized applications such as the card manager.
O.TRANSACTION	The TOE must provide a means to execute a set of operations atomically. See #.TRANSACTION for details.
O.OBJ-DELETION	The TOE shall ensure the object deletion shall not break references to objects. See #.OBJ-DELETION for further details.
O.DELETION	The TOE shall ensure that both applet and CAP file deletion perform as expected. See #.DELETION for details.
O.LOAD	The TOE shall ensure that the loading of a CAP file into the card is safe. Besides, for code loaded post-issuance, the TOE shall verify the integrity and authenticity evidences generated during the verification of the application CAP file by the verification authority. This verification by the TOE shall occur during the loading or later during the install process. Application Note: Usurpation of identity resulting from a malicious installation of an applet on the card may also be the result of perturbing the communication channel linking the CAD and the card. Even if the CAD is placed in a secure environment, the attacker may try to capture, duplicate, permute or modify the CAP files sent to the card. He may also try to send one of its own applications as if it came from the card issuer. Thus, this objective is intended to ensure the integrity and authenticity of loaded CAP files.
O.INSTALL	The TOE shall ensure that the installation of an applet performs as expected (See #.INSTALL for details). Besides, for code loaded post-issuance, the TOE shall verify the integrity and authenticity evidences generated during the verification of the application CAP file by the verification authority. If not performed during the loading process, this verification by the TOE shall occur during the install process.

O.SCP.IC	The SCP shall provide all IC security features against physical attacks. This security objective refers to the point (7) of the security aspect #.SCP: It is required that the IC is designed in accordance with a well-defined set of policies and Standards (likely specified in another protection profile), and will be tamper resistant to actually prevent an attacker from extracting or altering security data (like cryptographic keys) by using commonly employed techniques (physical probing and sophisticated analysis of the chip). This especially matters to the management (storage and operation) of cryptographic keys.
O.SCP.RECOVERY	If there is a loss of power, or if the smart card is withdrawn from the CAD while an operation is in progress, the SCP must allow the TOE to eventually complete the interrupted operation successfully, or recover to a consistent and secure state. This security objective refers to the security aspect #.SCP (1): The smart card platform must be secure with respect to the SFRs. Then after a power loss or sudden card removal prior to completion of some communication protocol, the SCP will allow the TOE on the next power up to either complete the interrupted operation or revert to a secure state.
O.SCP.SUPPORT	The SCP shall support the TSFs of the TOE. This security objective refers to the security aspects 2, 3, 4 and 5 of #.SCP: (2) It does not allow the TSFs to be bypassed or altered and does not allow access to other low-level functions than those made available by packages of the API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System. (3) It provides secure low-level cryptographic processing to the Java Card System. (4) It supports the needs for any update to a single persistent object or class field to be atomic, and possibly a low-level transaction mechanism. (5) It allows the Java Card System to store data in "persistent technology memory" or in volatile memory, depending on its needs (for instance, transient objects must not be stored in non-volatile memory). The memory model is structured and allows for low-level control accesses (segmentation fault detection).
From ‘Sensitive Array’ package	
O.SENSITIVE_ARRAYS_INTEG	The TOE shall ensure that only the currently selected applications may have a write access to the integrity-sensitive array object (javacard.framework.SensitiveArrays) created by that application. Any unauthorized modification through physical attacks to that integrity-sensitive array must be detected by the TOE and notified to the application.
From ‘Sensitive Result’ package	
O.SENSITIVE_RESULTS_INTEG	The TOE shall ensure that the sensitive results (javacardx.security.SensitiveResults) of sensitive operations executed by applications through the Java Card API are protected in integrity specifically against physical attacks.
From ‘Monotonic Counters’ package	
O.MTC-CTR-MNGT	The TOE shall provide a means to securely manage value of the monotonic counter. This concerns the optional package javacardx.security.util of the Java Card platform.
From ‘Cryptographic Certificate Management’ package	
O.CRT-MNGT	The TOE shall provide a means to securely manage cryptographic certificates. This concerns the optional package javacardx.security.cert of the Java Card platform.

Application note: Security Objectives O.OPERATE, O.RESOURCES are relevant security objectives for System Time package, as system time API extension package is implemented.

7.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

7.2.1 [PP-GP] Protection Profile

The following security objectives for the operational environment are listed in [PP-GP] and shall be considered for the present evaluation.

From core part	
OE.ISSUER	The Issuer shall be a trusted actor responsible for the behaviour of the SE.
OE.ADMIN	The administrators of the CCM servers (e.g. OTA or other kinds of servers) shall be trusted actors. They shall be trained to use and administrate those servers. They have the means and the equipment to perform their tasks. They must be aware of the sensitivity of the assets they manage and the responsibilities associated with the administration of CCM servers. Administrators obey the security policies and constitute, by this OE, no source of an inside attack.
OE.APPS-PROVIDER	The AP shall be a trusted actor that provides applications. The AP must be responsible for the APSD keys.
OE.VERIFICATION-AUTHORITY	The VA shall be a trusted actor with the capability to check and validate the digital signature attached to an application.
OE.KEY-ESCROW	The key escrow shall be a trusted actor in charge of the secure storage of the AP initial keys generated by the personaliser.
OE.PERSONALISER	The personaliser shall be a trusted actor in charge of the personalisation process. The personaliser shall ensure the security of the keys managed and loaded into the card: ▪ Issuer Security Domain keys (ISD keys) ▪ Application Provider Security Domain keys (APSD keys) ▪ Controlling Authority Security Domain keys (CASD keys).
OE.CONTROLLING-AUTHORITY	The CA shall be a trusted actor responsible for securing the creation and personalisation of APSD keys. The CA must be responsible for the CASD keys.
OE.SCP-SUPP	Secure Communication Protocols shall be supported and used by the operational environment.
OE.KEYS-PROT	During the TOE's use, the terminal in interaction with the TOE shall ensure the protection (integrity and confidentiality) of the applied keys by operational means and/or procedures.
OE.PRODUCTION	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 data (to prevent any possible copy, modification, retention, theft, or unauthorized use).
OE.APPLICATIONS	Developers and Validators shall comply with the security guidance and ensure that the rules are enforced.
OE.AID-MANAGEMENT	The VA shall verify that the AID of the application being loaded does not impersonate the AID known by another application on the card for the use of shareable services.
OE.LOADING	Application code, validated or certified depending on the application, is loaded onto the SE Platform using any kind of CCM servers (e.g. OTA or other kinds of servers used to perform card content management) and protocols with contactless or contact (e.g. USB) connectivity.
OE.SERVERS	The Issuer must enforce a policy to ensure the security of the applications stored on its CCM servers (e.g. OTA or other kinds of servers used to perform card content management).
OE.AP-KEYS	The SD-key-personaliser, the AP, and the key escrow must enforce a security policy securing the transmissions.
OE.ISD-KEYS	The security of the ISD keys must be ensured in the environment of the TOE.
OE.KEY-GENERATION	The personaliser must ensure that the generated keys cannot be accessed by unauthorised users.
OE.CA-KEYS	The CASD keys must be securely generated prior to storage in the SE card.
OE.KEY-CHANGE	The AP must change the initial keys of APSD before any operation on it.
From package 'Ciphered Load File Data Block (CLFDB)'	
OE.CLFDB-ENC-PR	The Load File Data Block shall be encrypted securely by a trusted SD provider. Application Note: See [GPCS] section C.6.
From package 'Delegated Management (DM)'	
OE.TOKEN-GEN	The Token shall be generated securely by a trusted entity according to the

	signature algorithms defined in GlobalPlatform specifications. Application Note: See [GPCS] sections B.1, B.2, B.3, B.4, and C.4.
OE.RECEIPT-VER	The Receipt shall be verified securely by a trusted entity according to the methods defined in GlobalPlatform specifications. Application Note: See [GPCS] sections B.1, B.2, B.3, B.4, and C.5.
From packages 'DAP Verification' and 'Mandated DAP Verification'	
OE.DAP_BLOCK_GEN	The DAP Block shall be generated securely by a trusted entity that verifies the content of the Load File Data Block linked to the hash.
From PP-Module 'OS Update'	
OE.OS-UPDATE-EVIDENCE	For additional code loaded pre issuance, evaluated technical measures implemented by the TOE or audited organisational measures must ensure that the additional code (1) has been issued by the genuine OS Developer and (2) has not been altered since it was issued by the genuine OS Developer. For additional code loaded post issuance, the OS Developer shall provide digital evidence to the TOE that (1) he is the genuine developer of the additional code and (2) the additional code has not been modified since it was issued by the genuine OS Developer.
OE.OS-UPDATE-ENCRYPTION	For additional code loaded post issuance, the OS Developer shall encrypt the additional code so that its confidentiality is ensured when it is transmitted to the TOE for loading and installation.
OE.SECURE_ACODE_MANAGEMENT	Key management processes related to the OS Update capability shall take place in a secure and audited environment. The key generation processes shall guarantee that cryptographic keys are of sufficient quality and appropriately secured to ensure confidentiality, authenticity, and integrity of the keys.

7.2.2 [PP-JCS] Protection Profile

The following security objectives for the operational environment are listed in [PP-JCS] and shall also be considered for the present evaluation.

OE.CAP_FILE	No CAP file loaded post-issuance shall contain native methods.
OE.VERIFICATION	All the bytecodes shall be verified at least once, before the loading, before the installation or before the execution, depending on the card capabilities, in order to ensure that each bytecode is valid at execution time. See #.VERIFICATION for details. Additionally, the applet shall follow all the recommendations, if any, mandated in the platform guidance for maintaining the isolation property of the platform. Application Note: constraints to maintain the isolation property of the platform are provided by the platform developer in application development guidance. The constraints apply to all application code loaded in the platform.
OE.CODE-EVIDENCE	For application code loaded pre-issuance, evaluated technical measures implemented by the TOE or audited organizational measures must ensure that loaded application has not been changed since the code verifications required in OE.VERIFICATION. For application code loaded post-issuance and verified off-card according to the requirements of OE.VERIFICATION, the verification authority shall provide digital evidence to the TOE that the application code has not been modified after the code verification and that he is the actor who performed code verification. For application code loaded post-issuance and partially or entirely verified on-card, technical measures must ensure that the verification required in OE.VERIFICATION are performed. On-card bytecode verifier is out of the scope of this Protection Profile. Application Note: for application code loaded post-issuance and verified off-card, the integrity and authenticity evidence can be achieved by electronic signature of the application code, after code verification, by the actor who performed verification.

7.3 SECURITY OBJECTIVES RATIONALE

7.3.1 Threats, OSPs and Assumptions coverage – Mapping tables from [PP-GP] Protection Profile

Threat	Security objectives
T.UNAUTHORISED-CARD-MGMT	O.CARD-MANAGEMENT, O.COMM-AUTH, O.COMM-INTEGRITY, O.COMM-CONFIDENTIALITY, O.APPLI-AUTH, O.PRIVILEGES-MANAGEMENT, O.LC-MANAGEMENT, O.DOMAIN-RIGHTS
T.LIFE-CYCLE	O.CARD-MANAGEMENT, O.DOMAIN-RIGHTS
T.COM-EXPLOIT	O.COMM-AUTH, O.COMM-INTEGRITY,O.COMM-CONFIDENTIALITY
T.BRUTE-FORCE-SCP	O.NO-KEY-REUSE
T.CLFDB-DISC	O.CLFDB-DECIPHER
T.CVM-IMPERSONATE	O.GLOBAL-CVM, O.CVM-BLOCK, O.CVM-MGMT
T.CVM-UPDATE	O.CVM-BLOCK, O.CVM-MGMT
T.BRUTE-FORCE-CVM	O.CVM-BLOCK, O.CVM-MGMT
T.RECEIPT	O.RECEIPT
T.TOKEN	O.TOKEN
T.ELF-UNAUTHORISED	O.ELF_AUTHORISED, O.CARD-MANAGEMENT, O.DOMAIN-RIGHTS, O.COMM-AUTH
T.ELF-VERSION	O.ELF_INTEGRITY, O.COMM-CONFIDENTIALITY, O.COMM-INTEGRITY
T.ELF-DATA-ACCESS	O.ELF_APP_DATA
T.ELF-DATA-INTEGRITY	O.ELF_APP_DATA
T.ELF-SESSION	O.ELF_SESSION
T.ELF-ILL-COMMAND	O.ELF_SESSION
T.ELF-RES-DATA	O.ELF_DATA_PRO
T.UNAUTHORISED-TOE-CODE-UPDATE	O.SECURE_LOAD_ACODE
T.FAKE-SGNVER-KEY	O.SECURE_LOAD_ACODE
T.WRONG-UPDATE-STATE	O.SECURE_AC_ACTIVATION, O.TOE_IDENTIFICATION
T.INTEG-OS-UPDATE-LOAD	O.SECURE_LOAD_ACODE
T.CONFID-OS-UPDATE-LOAD	O.CONFID-OS-UPDATE.LOAD
T.CONFID-APPLI-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.ARRAY_VIEWS_CONFID, O.ALARM, O.TRANSACTION, O.CIPHER, O.RNG, O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION
T.CONFID-JCS-CODE	OE.VERIFICATION, O.CARD-MANAGEMENT, O.NATIVE
T.CONFID-JCS-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.ALARM
T.INTEG-APPLI-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE
T.INTEG-APPLI-CODE.LOAD	O.LOAD, O.CARD-MANAGEMENT, OE.CODE-EVIDENCE
T.INTEG-APPLI-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.GLOBAL_ARRAYS_INTEG, O.ARRAY_VIEWS_INTEG, O.ALARM, O.TRANSACTION, O.CIPHER, O.RNG, O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION, OE.CODE-EVIDENCE, O.MTC-CTR-MNGT, O.CRT-MNGT
T.INTEG-APPLI-DATA.LOAD	O.LOAD, O.CARD-MANAGEMENT, OE.CODE-EVIDENCE
T.INTEG-JCS-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE,

	OE.CODE-EVIDENCE
T.INTEG-JCS-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.ALARM, OE.CODE-EVIDENCE
T.SID.1	O.CARD-MANAGEMENT, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.INSTALL, O.SID
T.SID.2	O.SCP.RECOVERY, O.SCP.SUPPORT, O.SID, O.OPERATE, O.FIREWALL, O.INSTALL
T.EXE-CODE.1	OE.VERIFICATION, O.FIREWALL
T.EXE-CODE.2	OE.VERIFICATION
T.NATIVE	OE.VERIFICATION, OE.CAP_FILE , O.NATIVE
T.RESOURCES	O.INSTALL, O.OPERATE, O.RESOURCES, O.SCP.RECOVERY, O.SCP.SUPPORT
T.DELETION	O.DELETION, O.CARD-MANAGEMENT
T.INSTALL	O.INSTALL, O.LOAD, O.CARD-MANAGEMENT
T.OBJ-DELETION	O.OBJ-DELETION
T.PHYSICAL	O.SCP.IC, O.SENSITIVE_ARRAYS_INTEG, O.SENSITIVE_RESULTS_INTEG

Table 3: Threats coverage by security objectives – Mapping table [PP-GP]

OSP	Security objectives
OSP.AID-MANAGEMENT	OE.AID-MANAGEMENT
OSP.LOADING	OE.LOADING
OSP.SERVERS	OE.SERVERS
OSP.APSD-KEYS	OE.AP-KEYS
OSP.ISD-KEYS	OE.ISD-KEYS
OSP.KEY-GENERATION	OE.KEY-GENERATION
OSP.CASD-KEYS	OE.CA-KEYS
OSP.KEY-CHANGE	OE.KEY-CHANGE
OSP.SECURITY-DOMAINS	O.SECURITY-DOMAINS
OSP.APPLICATIONS	OE.APPLICATIONS
OSP.CLFDB-ENC-PR	OE.CLFDB-ENC-PR
OSP.TOKEN-GEN	OE.TOKEN-GEN
OSP.RECEIPT-VER	OE.RECEIPT-VER
OSP.DAP_BLOCK_GEN	OE.DAP_BLOCK_GEN
OSP.ELF_DELE_OP	O.ELF_DELE_IRR
OSP.ATOMIC_ACTIVATION	O.SECURE_AC_ACTIVATION
OSP.TOE_IDENTIFICATION	O.TOE_IDENTIFICATION
OSP.ADDITIONAL_CODE_SIGNING	O.SECURE_LOAD_ACODE
OSP.ADDITIONAL_CODE_ENCRYPTION	O.CONFID-OS-UPDATE.LOAD, OE.OS-UPDATE-ENCRYPTION
OSP.VERIFICATION	OE.VERIFICATION, O.LOAD, OE.CODE-EVIDENCE

Table 4: OSP coverage by security objectives – Mapping table [PP-GP]

Assumption	Security objectives
A.ISSUER	OE.ISSUER
A.ADMIN	OE.ADMIN
A.APPS-PROVIDER	OE.APPS-PROVIDER
A.VERIFICATION-AUTHORITY	OE.VERIFICATION-AUTHORITY
A.KEY-ESCROW	OE.KEY-ESCROW
A.PERSONALISER	OE.PERSONALISER
A.CONTROLLING-AUTHORITY	OE.CONTROLLING-AUTHORITY
A.PRODUCTION	OE.PRODUCTION
A.SCP-SUPP	OE.SCP-SUPP

A.KEYS-PROT	OE.KEYS-PROT
A.OS-UPDATE-EVIDENCE	OE.OS-UPDATE-EVIDENCE
A.SECURE_ACODE_MANAGEMENT	OE.SECURE_ACODE_MANAGEMENT
A.CAP_FILE	OE.CAP_FILE
A.VERIFICATION	OE.VERIFICATION, OE.CODE-EVIDENCE

Table 5: Assumptions coverage by security objectives – Mapping table [PP-GP]

7.3.2 Threats coverage – Rationale from [PP-GP] Protection Profile

T.UNAUTHORISED-CARD-MGMT is covered by:
- O.CARD-MANAGEMENT controls the access to card management functions such as the loading, installation, extradition, or deletion of applets.
- O.COMM-AUTH prevents unauthorized users from initiating a malicious card management operation.
- O.COMM-INTEGRITY protects the integrity of the card management data while it is in transit to the card.
- O.COMM-CONFIDENTIALITY prevents disclosure of encrypted data transiting to the card.
- O.APPLI-AUTH requires that each application be authenticated before loading.
- O.DOMAIN-RIGHTS restricts the modification of an AP security domain key set to the AP owning it.
- O.PRIVILEGES-MANAGEMENT enforces the Privileges assignment and management functionalities for the on-card entities ISD, SSD, and Applications.
- O.LC-MANAGEMENT enforces the Life Cycle management for the Card, ELFs, SDs, and Applications.

T.LIFE-CYCLE is covered by:
- O.CARD-MANAGEMENT controls the access to the card management functions of loading, installation, extradition, and deletion of applets. Attacks for modification or exploitation of the current life cycle of applications are thus rendered impractical.
- O.DOMAIN-RIGHTS restricts the use of an AP security domain key set and thereby restricts the management of applications to the affected SD and to the AP owning the key set.

T.COM-EXPLOIT is covered by:
- O.COMM-AUTH prevents unauthorized users from initiating a malicious card management operation.
- O.COMM-INTEGRITY protects the integrity of the card management data while it is in transit to the card.
- O.COMM-CONFIDENTIALITY prevents disclosure of encrypted data transiting to the card.

T.BRUTE-FORCE-SCP is covered by O.NO-KEY-REUSE which ensures that session keys can be used only once.

T.CLFDB-DISC is covered by O.CLFDB-DECIPHER which protects the Ciphered Load File Data Block when it is transmitted to the SE for decryption prior to installation.

T.CVM-IMPERSONATE is covered by:
- O.GLOBAL-CVM restricts the modification of the security attributes of the CVM only to defined privileged applications appointed by the Card Manager.
- O.CVM-BLOCK blocks the global PIN used to authenticate the Cardholder if the maximum number of attempts has been reached.
- O.CVM-MGMT securely manages CVM objects.

T.CVM-UPDATE is covered by:
- O.CVM-BLOCK
- O.CVM-MGMT

T.BRUTE-FORCE-CVM is covered by:
- O.CVM-BLOCK blocks the global PIN used to authenticate the Cardholder if the maximum number of attempts has been reached.
- O.CVM-MGMT securely manages CVM objects.

T.RECEIPT is covered by O.RECEIPT which generates non repudiable receipts of the completion of card management operations.

T.TOKEN is covered by O.TOKEN which verifies tokens during the processing of card management operations.

T.ELF-UNAUTHORISED is covered by:
- O.ELF_AUTHORISED ensures that only authorized entities are able to load ELFs.
- O.CARD-MANAGEMENT controls the access to card management functions such as the loading, installation, extradition, or deletion of applets.
- O.DOMAIN-RIGHTS restricts the use of an AP security domain key set and therewith the management of applications to the affected SD and to the AP owning the key set.
- O.COMM-AUTH prevents unauthorized users from initiating a malicious card management operation.

T.ELF-VERSION is covered by:
- O.ELF_INTEGRITY preserves the ELF integrity and confidentiality (if required) during the loading process.
- O.COMM-CONFIDENTIALITY prevents disclosure of encrypted data transiting to the card.
- O.COMM-INTEGRITY protects the integrity of the card management data while it is in transit to the card.

T.ELF-DATA-ACCESS is covered by O.ELF_APP_DATA which maintains the integrity & consistency of Registry data.

T.ELF-DATA-INTEGRITY is covered by O.ELF_APP_DATA which maintains the integrity & consistency of Registry data.

T.ELF-SESSION is covered by O.ELF_SESSION which ensures that the upgrade process is performed securely.

T.ELF-ILL-COMMAND is covered by O.ELF_SESSION which ensures that the upgrade process is performed securely.
T.ELF-RES-DATA is covered by O.ELF_DATA_PRO which protects ELF information when the resource is reallocated.

T.UNAUTHORISED-TOE-CODE-UPDATE is covered by O.SECURE_LOAD_ACODE which ensures that only an allowed version of the additional code can be loaded.

T.FAKE-SGNVER-KEY is covered by O.SECURE_LOAD_ACODE which ensures that only an allowed version of the additional code can be loaded.

T.WRONG-UPDATE-STATE is covered by:
- O.SECURE_AC_ACTIVATION ensures that the activation of the additional code and update of the Identification Data are performed at the same time in an atomic way.
- O.TOE_IDENTIFICATION guarantees the integrity of the stored Identification Data in its non-volatile memory.

T.INTEG-OS-UPDATE-LOAD is covered by O.SECURE_LOAD_ACODE which ensures that only an allowed version of the additional code can be loaded.

T.CONFID-OS-UPDATE-LOAD is covered by O.CONFID-OS-UPDATE.LOAD which performs the decryption of the additional code prior installation.

T.CONFID-APPLI-DATA is countered by the security objective for the operational environment regarding bytecode verification (OE.VERIFICATION). It is also covered by the isolation commitments stated in the (O.FIREWALL) objective. It relies in its turn on the correct identification of applets stated in (O.SID). Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (O.OPERATE) objective. As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken. The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively. The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter. As applets may need to share some data or communicate with the CAD, cryptographic functions are required to actually protect the exchanged information (O.CIPHER, O.RNG). Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them. Keys and PIN's are particular cases of an application's sensitive data (the Java Card System may possess keys as well) that ask for appropriate management (O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION). If the PIN class of the Java Card API is used, the objective (O.FIREWALL) shall contribute in covering this threat by controlling the sharing of the global PIN between the applets. Other application data that is sent to the applet as clear text arrives to the APDU buffer, which is a resource shared by all applications. The disclosure of such data is prevented by the security objective O.GLOBAL_ARRAYS_CONFID. An applet might share data buffer with another applet using array views without the array view security attribute ATTR_READABLE_VIEW. The disclosure of data of the applet creating the array view is prevented by the security object O.ARRAY_VIEWS_CONFID. Finally, any attempt to read a piece of information that was previously used by an application but has been logically deleted is countered by the O.REALLOCATION objective. That objective states that any information that was formerly stored in a memory block shall be cleared before the block is reused.

T.CONFID-JCS-CODE is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of those instructions enables reading a piece of code, no Java Card applet can therefore be executed to disclose a piece of code. Native applications are also harmless because of the objective O.NATIVE, so no application can be run to disclose a piece of code. The (#.VERIFICATION) security aspect is addressed in this PP by the objective for the environment OE.VERIFICATION. The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively.

T.CONFID-JCS-DATA is covered by bytecode verification (OE.VERIFICATION) and the isolation commitments stated in the (O.FIREWALL) security objective. This latter objective also relies in its turn on the correct identification of applets stated in (O.SID). Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (O.OPERATE) objective. As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken. The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively. The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

T.INTEG-APPLI-CODE is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code. Native applications are also harmless because of the objective O.NATIVE, so no application can run to modify a piece of code. The (#.VERIFICATION) security aspect is addressed in this configuration by the objective for the environment OE.VERIFICATION. The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively. The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that integrity and authenticity evidences exist for the application code loaded into the platform.

T.INTEG-APPLI-CODE.LOAD is countered by the security objective O.LOAD which ensures that the loading of CAP files is done securely and thus preserves the integrity of CAP files' code. The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. By controlling the access to card management functions such as the installation, update or deletion of applets the objective O.CARD-MANAGEMENT contributes to cover this threat.

T.INTEG-APPLI-DATA is countered by bytecode verification (OE.VERIFICATION) and the isolation commitments stated in the (O.FIREWALL) objective. This latter objective also relies in its turn on the correct identification of applets stated in (O.SID). Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (O.OPERATE) objective. As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken. The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively. The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter. Concerning the confidentiality and integrity of application sensitive data, as applets may need to share some data or communicate with the CAD, cryptographic functions are required to actually protect the exchanged information (O.CIPHER, O.RNG). Remark that even if the TOE shall provide access to the appropriate TSFs, it is still the responsibility of the applets to use them. Keys and PIN's are particular cases of an application's sensitive data (the Java Card System may possess keys as well) that ask for appropriate management (O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION). If the PIN class of the Java Card API is used, the objective (O.FIREWALL) is also concerned. Other application data that is sent to the applet as clear text arrives to the APDU buffer, which is a resource shared by all applications. The integrity of the information stored in that buffer is ensured by the objective O.GLOBAL_ARRAYS_INTEG. An applet might share data buffer with another applet using array views without the array view security attribute ATTR_WRITABLE_VIEW. The integrity of data of the applet creating the array view is ensured by the security objective O.ARRAY_VIEWS_INTEG. Additionally, any attempt to read a piece of information that was previously used by an application but has been logically deleted is countered by the O.REALLOCATION objective. That objective states that any information that was formerly stored in a memory block shall be cleared before the block is reused. Finally, this threat is countered by the security objective for monotonic counter management (O.MTC-CTR-MNGT) such that value of the monotonic counter will be protected against any unauthorized change, and by the security objective for certificate management (O.CRT-MNGT) such that certificate data will be protected against any unauthorized change.

T.INTEG-APPLI-DATA.LOAD is countered by the security objective O.LOAD which ensures that the loading of CAP files is done securely and thus preserves the integrity of applications data. The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. By controlling the access to card management functions such as the installation, update or deletion of applets the objective O.CARD-MANAGEMENT contributes to cover this threat.

T.INTEG-JCS-CODE is countered by the list of properties described in the (#.VERIFICATION) security aspect. Bytecode verification ensures that each of the instructions used on the Java Card platform is used for its intended purpose and in the intended scope of accessibility. As none of these instructions enables modifying a piece of code, no Java Card applet can therefore be executed to modify a piece of code. Native applications are also harmless because of the objective O.NATIVE, so no application can be run to modify a piece of code. The (#.VERIFICATION) security aspect is addressed in this configuration by the objective for the environment OE.VERIFICATION. The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively. The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity.

T.INTEG-JCS-DATA is countered by bytecode verification (OE.VERIFICATION) and the isolation commitments stated in the (O.FIREWALL) objective. This latter objective also relies in its turn on the correct identification of applets stated in (O.SID). Moreover, as the firewall is dynamically enforced, it shall never stop operating, as stated in the (O.OPERATE) objective. As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, so that the appropriate counter-measure can be taken. The objectives O.CARD-MANAGEMENT and OE.VERIFICATION contribute to cover this threat by controlling the access to card management functions and by checking the bytecode, respectively. The objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring that the application code loaded into the platform has not been changed after code verification, which ensures code integrity and authenticity. The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE and O.ALARM objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

As impersonation is usually the result of successfully disclosing and modifying some assets, **T.SID.1** is mainly countered by the objectives concerning the isolation of application data (like PINs), ensured by the (O.FIREWALL). Uniqueness of subject-identity (O.SID) also participates to face this threat. It should be noticed that the AIDs, which are used for applet identification, are TSF data. In this configuration, usurpation of identity resulting from a malicious installation of an applet on the card is covered by the objective O.INSTALL. The installation parameters of an applet (like its name) are loaded into a global array that is also shared by all the applications. The disclosure of those parameters (which could be used to impersonate the applet) is countered by the objectives O.GLOBAL_ARRAYS_CONFID and O.GLOBAL_ARRAYS_INTEG. The objective O.CARD-MANAGEMENT contributes, by preventing usurpation of identity resulting from a malicious installation of an applet on the card, to counter this threat.

T.SID.2 is covered by integrity of TSF data, subject-identification (O.SID), the firewall (O.FIREWALL) and its good working order (O.OPERATE). The objective O.INSTALL contributes to counter this threat by ensuring that installing an applet has no effect on the state of other applets and thus can't change the TOE's attribution of privileged roles. The objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE objective of the TOE, so they are indirectly related to the threats that this latter objective contributes to counter.

T.EXE-CODE.1 coverage: unauthorized execution of a method is prevented by the objective OE.VERIFICATION. This threat particularly concerns the point (8) of the security aspect #.VERIFICATION (access modifiers and scope of accessibility for classes, fields and methods). The O.FIREWALL objective is also concerned, because it prevents the execution of non-shareable methods of a class instance by any subject apart from the class instance owner.

T.EXE-CODE.2 coverage: unauthorized execution of a method fragment or arbitrary data is prevented by the objective OE.VERIFICATION. This threat particularly concerns those points of the security aspect related to control flow confinement and the validity of the method references used in the bytecodes.
T.NATIVE is countered by O.NATIVE which ensures that a Java Card applet can only access native methods indirectly that is, through an API. OE.CAP_FILE also covers this threat by ensuring that no CAP files containing native code shall be loaded in post-issuance. In addition to this, the bytecode verifier also prevents the program counter of an applet to jump into a piece of native code by confining the control flow to the currently executed method (OE.VERIFICATION).

T.RESOURCES is directly countered by objectives on resource-management (O.RESOURCES) for runtime purposes and good working order (O.OPERATE) in a general manner. Consumption of resources during installation and other card management operations are covered, in case of failure, by O.INSTALL. It should be noticed that, for what relates to CPU usage, the Java Card platform is single-threaded and it is possible for an ill-formed application (either native or not) to monopolize the CPU. However, a smart card can be physically interrupted (card removal or hardware reset) and most CADs implement a timeout policy that prevent them from being blocked should a card fails to answer. That point is out of scope of this Protection Profile, though. Finally, the objectives O.SCP.RECOVERY and O.SCP.SUPPORT are intended to support the O.OPERATE and O.RESOURCES objectives of the TOE, so they are indirectly related to the threats that these latter objectives contribute to counter.

T.DELETION is covered by is covered by the O.DELETION security objective which ensures that both applet and CAP file deletion perform as expected. The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

T.INSTALL is covered by the security objective O.INSTALL which ensures that the installation of an applet performs as expected and the security objectives O.LOAD which ensures that the loading of a CAP file into the card is safe. The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

T.OBJ-DELETION is covered by the O.OBJ-DELETION security objective which ensures that object deletion shall not break references to objects.

T.PHYSICAL is covered by O.SCP.IC, as physical protections rely on the underlying platform. It is also partially covered by O.SENSITIVE_ARRAYS_INTEG which requires the TOE to detect and notify the application if any unauthorized modification of the integrity-sensitive array object through physical attacks occurred, and by O.SENSITIVE_RESULTS_INTEG which ensures that sensitive results are protected against unauthorized modification by physical attacks.

7.3.3 OSP coverage – Rationale from [PP-GP]

OSP.AID-MANAGEMENT is directly enforced by the security objective for the operational environment of the TOE OE.AID-MANAGEMENT.

OSP.LOADING is enforced by the security objective for the operational environment of the TOE OE.LOADING.

OSP.SERVERS is enforced by the security objective for the operational environment of the TOE OE.SERVERS.

OSP.APSD-KEYS is enforced by the security objective for the operational environment of the TOE OE.AP-KEYS.

OSP.ISD-KEYS is enforced by the security objective for the operational environment of the TOE OE.ISD-KEYS.
OSP.KEY-GENERATION is enforced by the security objective for the operational environment of the TOE OE.KEY-GENERATION.

OSP.CASD-KEYS is enforced by the security objective for the operational environment of the TOE OE.CA-KEYS.

OSP.KEY-CHANGE is enforced by the security objective for the operational environment of the TOE OE.KEY-CHANGE.

OSP.SECURITY-DOMAINS is enforced by the security objective for the TOE O.SECURITY-DOMAINS.

OSP.APPLICATIONS is enforced by the security objective for the operational environment of the TOE OE.APPLICATIONS.

OSP.CLFDB-ENC-PR is enforced by the security objective for the operational environment of the TOE OE.CLFDB-ENC-PR.

OSP.TOKEN-GEN is enforced by the security objective for the operational environment of the TOE OE.TOKEN-GEN.

OSP.RECEIPT-VER is enforced by the security objective for the operational environment of the TOE OE.RECEIPT-VER.

OSP.DAP_BLOCK_GEN is enforced by the security objective for the operational environment of the TOE OE.DAP_BLOCK_GEN.

OSP.ELF_DELE_OP is covered by O.ELF_DELE_IRR which provides an atomic and irreversible deletion operation of the Application instances and ELF(s).

OSP.ATOMIC_ACTIVATION is covered by O.SECURE_AC_ACTIVATION which ensures that the activation of the additional code and update of the Identification Data are performed at the same time in an atomic way.

OSP.TOE_IDENTIFICATION is covered by O.TOE_IDENTIFICATION which guarantees the integrity of the stored Identification Data in its non-volatile memory.

OSP.ADDITIONAL_CODE_SIGNING is covered by O.SECURE_LOAD_ACODE ensures that only an allowed version of the additional code can be loaded.

OSP.ADDITIONAL_CODE_ENCRYPTION is covered by:

- O.CONFID-OS-UPDATE.LOAD performs the decryption of the additional code prior installation.
- OE.OS-UPDATE-ENCRYPTION requires confidentiality protection measures on the additional code loaded when it is transmitted to the TOE for loading and installation.

OSP.VERIFICATION is upheld by the security objective of the environment OE.VERIFICATION which guarantees that all the bytecodes shall be verified at least once, before the loading, before the installation or before the execution in order to ensure that each bytecode is valid at execution time. This policy is also upheld by the security objective of the environment OE.CODE-EVIDENCE which ensures that evidences exist that the application code has been verified and not changed after verification, and by the security objective for the TOE O.LOAD which shall ensure that the loading of a CAP file into the card is safe.

7.3.4 Assumptions coverage – Rationale from [PP-GP]

A.ISSUER is directly upheld by OE.ISSUER.

A.ADMIN is directly upheld by OE.ADMIN.

A.APPS-PROVIDER is directly upheld by OE.APPS-PROVIDER.

A.VERIFICATION-AUTHORITY is directly upheld by OE.VERIFICATION-AUTHORITY.

A.KEY-ESCROW is directly upheld by OE.KEY-ESCROW.

A.PERSONALISER is directly upheld by OE.PERSONALISER.

A.CONTROLLING-AUTHORITY is directly upheld by OE.CONTROLLING-AUTHORITY.

A.PRODUCTION is directly upheld by OE.PRODUCTION.

A.SCP-SUPP is directly upheld by OE.SCP-SUPP.

A.KEYS-PROT is directly upheld by OE.KEYS-PROT.

A.OS-UPDATE-EVIDENCE is covered by OE.OS-UPDATE-EVIDENCE which requires integrity protection measures on the additional code loaded.

A.SECURE_ACODE_MANAGEMENT is covered by OE.SECURE_ACODE_MANAGEMENT ensures that a key management process related to the OS Update capability is in place in a secure and audited environment.

A.CAP_FILE is upheld by the security objective for the operational environment OE.CAP_FILE which ensures that no CAP file loaded post-issuance shall contain native methods.

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

7.4 COMPOSITION TASKS – OBJECTIVES PART

7.4.1 Statement of compatibility – TOE Objectives part

The following table (see next page) lists the relevant TOE security objectives of the security target [ST_IC], and provides the link to the composite-product TOE security objectives, showing that there is no contradiction between the two sets of objectives.

Label of the chip TOE security objective	Title of the chip TOE security objective	Content of the chip TOE security objective	Linked Composite-product TOE security objectives
O.Leak-Inherent	Protection against Inherent Information Leakage	The TOE must provide protection against disclosure of confidential data (User Data or TSF data) stored and/or processed in the Smartcard IC - By measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and - By measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines).	O.SCP.SUPPORT O.SCP.IC
O.Phys-Probing	Protection against Physical Probing	The TOE must provide protection against disclosure/reconstruction of user data while stored in protected memory areas and processed or against the disclosure of other critical information about the operation of the TOE. This includes protection against - measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or - measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis) with a prior reverse-engineering to understand the design and its properties and functions.	O.SCP.SUPPORT O.SCP.IC
O.Malfunction	Protection against Malfunctions	The TOE must ensure its correct operation. The TOE must indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields.	O.OPERATE
O.Phys-Manipulation	Protection against Physical Manipulation	The TOE must provide protection against manipulation of the TOE (including its software and TSF data), the Smartcard Embedded Software and the user data of the Composite TOE. This includes protection against - Reverse-engineering (understanding the design and its properties and functions), - Manipulation of the hardware and any data, as well as - Undetected manipulation of memory contents.	O.SCP.SUPPORT O.SCP.IC
O.Leak-Forced	Protection against Forced Information Leakage	The Security IC must be protected against disclosure of confidential data processed in the Security IC (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker - By forcing a malfunction (refer to “Protection against Malfunction due to Environmental Stress (O.Malfunction)” and/or - By a physical manipulation (refer to “Protection against Physical Manipulation (O.Phys-Manipulation)”. If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.	O.SCP.SUPPORT O.SCP.IC
O.Abuse-Func	Protection against Abuse of Functionality	The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order to (i) disclose critical user data of the Composite TOE, (ii) manipulate critical user data of the Composite TOE, (iii) manipulate Security IC Embedded Software or (iv) bypass, deactivate, change or explore security features or security services of the TOE. Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Test Software which are not specified here.	O.SCP.SUPPORT
O.Identification	TOE Identification	The TOE must provide means to store Initialization Data and Pre-personalization Data in its non-volatile memory. The Initialization Data (or parts of them) are used for TOE identification.	No direct link to the composite-product TOE objectives, however chip traceability information

Label of the chip TOE security objective	Title of the chip TOE security objective	Content of the chip TOE security objective	Linked Composite-product TOE security objectives
			stored in NVM is used by the TOE to answer identification CC requirements.
O.RND	Random Numbers	The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have a sufficient entropy. The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.	O.RNG
O.Mem-Access	Area based Memory Access Control	The TOE must provide the Smartcard Embedded Software with the capability to define restricted access memory areas. The TOE must then enforce the partitioning of such memory areas so that access of software to memory areas is controlled as required, for example, in a multi-application environment.	O.SCP.SUPPORT
O.Cap_Avail_Loader	Capability and availability of the Loader	The TSF provides limited capability of the Loader functionality and irreversible termination of the Loader in order to protect stored user data from disclosure and manipulation.	O.LC-MANAGEMENT O.SCP.SUPPORT
O.Ctrl_Auth_Loader	Access control and authenticity for the Loader	The TSF provides trusted communication channel with authorized user, supports confidentiality protection and authentication of the user data to be loaded and access control for usage of the Loader functionality.	This IC security objective supports the loading of the TEQS V1.0 software during phase 5 (under Samsung LSI authority).
O.TDES	Cryptographic service Triple-DES	The TOE provides secure hardware based cryptographic services implementing the Triple-DES for encryption and decryption.	O.CIPHER
O.AES	Cryptographic service AES	The TOE provides secure hardware based cryptographic services for the AES for encryption and decryption.	O.CIPHER
O.Authentication	Authentication to external entities	The TOE shall be able to authenticate itself to external entities. The Initialization Data (or parts of them) are used for TOE authentication verification data.	This IC security objective supports the loading of the TEQS V1.0 software during phase 5 (under Samsung LSI authority).
O.Prot_TSF_Confidentiality	Protection of the confidentiality of the TSF	The TOE must provide protection against disclosure of confidential operations of the Security IC (loader, memory management unit…) through the use of a dedicated code loaded on open samples.	No direct link to the composite TOE security objectives, nevertheless it supports the IC global robustness and thus

Label of the chip TOE security objective	Title of the chip TOE security objective	Content of the chip TOE security objective	Linked Composite-product TOE security objectives
			participates to the composite TOE resistance to attacks.

7.4.2 Statement of compatibility – ENV Objectives part

The following table lists the relevant ENV security objectives of the security target [ST_IC], and provides the link to the composite-product, showing that they have been taken into account and that no contradiction has been introduced.

IC ENV security objective label	IC ENV security objective title	IC ENV security objective content	Link to the composite-product
OE.Resp-Appl	Treatment of user data of the Composite TOE	Security relevant user data of the Composite TOE (especially cryptographic keys) are treated by the Security IC Embedded Software as required by the security needs of the specific application context. For example the Security IC Embedded Software will not disclose security relevant user data of the Composite TOE to unauthorized users or processes when communicating with a terminal.	Covered by TOE Security Objectives: O.COMM-AUTH, O.COMM-INTEGRITY O.COMM-CONFIDENTIALITY O.KEY-MNGT, O.PIN-MNGT
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 unauthorized use). This means that Phases after TOE Delivery up to the end of Phase 6 must be protected appropriately.	• During phases 4 and 5: covered by the ALC composite-SARs • During phases 6 and 7, covered by OE.PRODUCTION.
OE.Lim_Block_Loader	Limitation of capability and blocking the Loader	The Composite Product Manufacturer will protect the Loader functionality against misuse, limit the capability of the Loader and terminate irreversibly the Loader after intended usage of the Loader and before the end of phase 5. Note: To maintain the confidentiality of the data of the composite TOE, the intended usage of the Loader is limited to the phase 5 of the life cycle.	No contradiction with the composite TOE objectives, the Loader is deactivated before the end of phase 5.

OE.Loader_Usage	Secure communication and usage of the Loader	The authorized user must support the trusted communication channel with the TOE by confidentiality protection and authenticity proof of the data to be loaded and fulfilling the access conditions required by the Loader.	No contradiction with composite TOE objectives.
OE.TOE_Auth	External entities authenticating of the TOE	The operational environment shall support the authentication verification mechanism and know authentication reference data of the TOE.	Fulfilled by Samsung LSI during phases 4 & 5. No contradiction with composite TOE objectives.

8 Extended components definition

8.1 EXTENDED COMPONENT FCS_RNG.1

8.1.1 Description

To define the IT security functional requirements of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

This family also defines quality requirements for the generation of random numbers which are intended to be used for cryptographic purposes.

8.1.2 Definition

Hierarchical to: No other components.
Dependencies: No dependencies.
Management: No management activities are foreseen.
Audit: No actions are defined to be auditable.

FCS_RNG.1 Random Number Generation

FCS_RNG.1.1 The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid, 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].

8.2 EXTENDED COMPONENT FCS_CKM.5

8.2.1 Description

This chapter describes a component of the family Cryptographic key management (FCS_CKM) for key derivation as process by which one or more keys are calculated from either a pre-shared key or a shared secret and other information. Key derivation is the deterministic repeatable process by which one or more keys are calculated from both a pre-shared key or shared secret, and other information, while key generation required by FCS_CKM.1 uses internal random numbers.

The component FCS_CKM.5 is on the same level as the other components of the family FCS_CKM.

8.2.2 Definition

FCS_CKM.5 Requires the TOE to provide key derivation.

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
Management: No management activities are foreseen.

Audit: The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Minimal: Success and failure of the activity.
b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).

FCS_CKM.5 Cryptographic key derivation

FCS_CKM.5.1 The TSF shall derive cryptographic keys [assignment: key type] from [assignment: input parameters] in accordance with a specified cryptographic key derivation algorithm [assignment: cryptographic key derivation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

9 Security requirements

9.1 SECURITY FUNCTIONAL REQUIREMENTS

9.1.1 Typographical conventions

The following conventions are used in the definitions of the SFRs:

- Selections, assignments and refinements that have already been made in the [PP-GP] and [PP-JCS] Protection Profiles are **in bold**, and the original text on which the selection, assignment or refinement has been made is not reminded.
- Selections, assignments and refinements made in this ST are **in bold and underlined**, and the PP original text on which the selection or assignment has been made is indicated in a footnote.
- Iteration operations on SFR components are denoted by showing a slash "/" and the iteration indicator after the SFR component identifier.

9.1.2 [PP-GP] Protection Profile

GlobalPlatform Card Management - Security Functional Requirements

Application note: patch management is an extension of the card management defined in GlobalPlatform since a patch is managed as a JavaCard Package, loaded as a standard executable load file and registered with specific attributes handled with GemActivate.

ELF loading

FDP_IFC.2/GP-ELF Complete information flow control

FDP_IFC.2.1/GP-ELF The TSF shall enforce the **ELF Loading information flow control SFP** on

- **Subjects: S.SD, S.CAD, S.OPEN**
- **Information: APDU commands INSTALL and LOAD, GlobalPlatform APIs for loading and installing ELF**

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

FDP_IFC.2.2/GP-ELF 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.

Application Note:

- This SFR replaces FDP_IFC.2/CM of [PP-JCS].
- The subject S.SD can be the ISD, an APSD, or the CASD.
- GlobalPlatform's card content management APDU commands and API methods are described in [GPCS] Chapter 11 and Appendix A.1, respectively.

FDP_IFF.1/GP-ELF Complete information flow control

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

- **Subjects: S.SD, S.OPEN**
- **Information: APDU commands INSTALL and LOAD, GlobalPlatform APIs for loading and installing ELF**

- **Security attributes: Card Life Cycle state, ELF signature verification status, ELF AID, SD privileges, Secure Channel Security Level**[2].

FDP_IFF.1.2/GP-ELF The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **S.SD implements one or more Secure Channel Protocols, namely SCP02, SCP03, SCP11**[3]**, each with a complete Secure Channel Key Set.**
- **S.SD has all of the cryptographic keys required by its privileges (e.g. CLFDB, DAP, DM).**
- **On receipt of INSTALL or LOAD commands, S.OPEN checks that the card Life Cycle State is not CARD_LOCKED or TERMINATED.**
- **S.OPEN accepts an ELF only if its integrity and authenticity has been verified.**
- **S.OPEN accepts an ELF only if its AID is not already registered by the TSF**[4]

FDP_IFF.1.3/GP-ELF The TSF shall enforce the **none**[5].

FDP_IFF.1.4/GP-ELF The TSF shall explicitly authorize an information flow based on the following rules: **none**[6].

FDP_IFF.1.5/GP-ELF The TSF shall explicitly deny an information flow based on the following rules:

- **S.OPEN fails to verify the integrity and request verification of the authenticity for ELFs**
- **S.OPEN fails to verify the Card Life Cycle state**
- **S.OPEN fails to verify the SD privileges.**
- **S.SD fails to verify the security level applied to protect INSTALL or LOAD commands.**
- **S.SD fails to set the security level (integrity and/or confidentiality), to apply to the next incoming command and/or next outgoing response.**
- **S.SD fails to unwrap INSTALL or LOAD commands.**
- **The ELF AID is already registered within the card**[7]

Application Note:

- This SFR refines and replaces FDP_IFF.1/CM of [PP-JCS].
- APDUs belonging to the policy ELF Loading information flow control SFP are described in the following references:
  - For INSTALL, see [GPCS] section 11.5.
  - For LOAD, see [GPCS] section 11.6.
- The INSTALL and LOAD commands must only be issued within a Secure Channel Session; the levels of security for these commands depend on the security level defined in the EXTERNAL AUTHENTICATE command.
- The Minimum Security Level of INSTALL and LOAD is 'AUTHENTICATED' as defined in [GPCS] section 10.6.
- For more details about the rules to be applied to each role of INSTALL command, refer to [GPCS] sections 9.3 and 3.4.

FDP_ITC.2/GP-ELF Import of user data with security attributes

FDP_ITC.2.1/GP-ELF The TSF shall enforce the **ELF Loading information flow control SFP** when importing user data, controlled under the SFP, from outside of the TOE.

---

[2] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]
[3] [selection: SCP02, SCP03, SCP10, SCP11, SCP22, SCP80, SCP81]
[4] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]
[5] [assignment: additional information flow control SFP rules]
[6] [assignment: rules, based on security attributes, that explicitly authorize information flows]
[7] [assignment: rules, based on security attributes, that explicitly deny information flows]

FDP_ITC.2.2/GP-ELF The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/GP-ELF The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/GP-ELF The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/GP-ELF The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:

- **Referring to Java Card rules defined in [JCVM310] and [JCRE310]: ELF loading is allowed only if, for each dependent ELF, its AID attribute is equal to a resident ELF AID attribute, and the major (minor) Version attribute associated with the dependent ELF is less than or equal to the major (minor) Version attribute associated with the resident ELF.**
- **None**[8]

Application Note:

- This SFR corresponds to FDP_ITC.2/Installer of [PP-JCS].
- Java Card rules are defined in [JCVM310] sections 4.4 and 4.5 and [JCRE310] section 11.
- The TSF shall use the INSTALL data format and the LOAD data format when interpreting the user data from outside the TOE.

Data & Key Loading

FDP_IFC.2/GP-KL Complete information flow control

FDP_IFC.2.1/GP-KL The TSF shall enforce the **Data & Key Loading information flow control SFP** on

- **Subjects: S.SD, S.CAD, S.OPEN, Application**
- **Information: GlobalPlatform APDU commands STORE DATA and PUT KEY, GlobalPlatform APIs for loading and storing data and keys**

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

FDP_IFC.2.2/GP-KL 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.

Application Note:

- GlobalPlatform's card content management APDU commands and API methods are described in [GPCS] Chapter 11 and Appendix A.1, respectively.
- The subject S.SD can be the ISD, an APSD, or the CASD.

FDP_IFF.1/GP-KL Complete information flow control

FDP_IFF.1.1/GP-KL The TSF shall enforce the **Data & Key Loading information flow control SFP** based on the following types of subject and information security attributes:

- **Subjects: S.SD, S.OPEN**
- **GlobalPlatform APDU commands STORE DATA and PUT KEY, GlobalPlatform APIs for loading and storing data and keys**
- **Security attributes: card Life Cycle State, Application and SD Life Cycle states, Secure Channel Security Level, SD and Application privileges**[9].

[8] [assignment: additional importation control rules]

[9] [assignment: list of subjects and information controlled under the indicated SFP, and for each, the security attributes]

FDP_IFF.1.2/GP-KL The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:

- **S.SD implements one or more Secure Channel Protocols, namely SCP02, SCP03, SCP11[10], each equipped with a complete Secure Channel Key Set.**
- **S.SD has all of the cryptographic keys required by its privileges (e.g. CLFDB, DAP, DM).**
- **An Application accepts a message only if it comes from the S.SD it belongs to.**
- **On receipt of a request to forward STORE DATA or PUT KEY commands to an Application, S.OPEN checks that the card Life Cycle State is not CARD_LOCKED or TERMINATED.**
- **On receipt of a request to forward STORE DATA or PUT KEY commands to an Application, the S.OPEN checks that the requesting S.SD has no restrictions for personalization.**
- **S.SD unwraps STORE DATA or PUT KEY according to the Current Security Level of the current Secure Channel Session and prior to the command forwarding to the targeted Application or SD.**
- **S.OPEN verifies that the targeted application implements a personalization interface[11]**

FDP_IFF.1.3/GP-KL The TSF shall enforce the **none[12]**.

FDP_IFF.1.4/GP-KL The TSF shall explicitly authorize an information flow based on the following rules: **none[13]**.

FDP_IFF.1.5/GP-KL The TSF shall explicitly deny an information flow based on the following rules:

- **S.OPEN fails to verify the Card Life Cycle, Application and SD Life Cycle states.**
- **S.OPEN fails to verify the privileges belonging to an SD or an Application.**
- **S.SD fails to unwrap STORE DATA or PUT KEY.**
- **S.SD fails to verify the security level applied to protect APDU commands.**
- **S.SD fails to set the security level (integrity and/or confidentiality), to apply to the next incoming command and/or next outgoing response.**
- **S.OPEN fails to verify that the targeted application implements a personalization interface.[14]**

Application Note:

- APDUs belonging to the Data & Key Loading information flow control SFP are described in the following references:
  - For PUT KEY, see [GPCS] section 11.8.
  - For STORE DATA, see [GPCS] section 11.11.
- The PUT KEY and STORE DATA commands must only be issued within a Secure Channel Session; the levels of security for these commands depend on the security level defined in the EXTERNAL AUTHENTICATE command.
- The Minimum Security Level of PUT KEY and STORE DATA is 'AUTHENTICATED' as defined in [GPCS] section 10.6.
- For more details about Key Access Conditions, Data and Key Management, refer to [GPCS] sections 7.5.2 and 7.6.

FDP_ITC.2/GP-KL Import of user data with security attributes

FDP_ITC.2.1/GP-KL The TSF shall enforce the **Data & Key Loading information flow control SFP** when importing user data, controlled under the SFP, from outside of the TOE.

[10] [selection: SCP02, SCP03, SCP10, SCP11, SCP22, SCP80, SCP81]
[11] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]
[12] [assignment: additional information flow control SFP rules]
[13] [assignment: rules, based on security attributes, that explicitly authorize information flows]
[14] [assignment: rules, based on security attributes, that explicitly deny information flows]

FDP_ITC.2.2/GP-KL The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/GP-KL The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/GP-KL The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/GP-KL The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:

- **The algorithms and key sizes of the imported keys shall be supported by the SE**
- **The Key Identifier (Key ID) of the imported keys shall be in an allowed range as specified in section 4 of [CIC]**[15]

Application Note:

- The algorithms and key sizes of the imported keys shall be supported by the Card as specified in [GPCS] Appendices B and C.
- PUT KEY and STORE DATA are described in [GPCS] sections 11.8 and 11.11.

Life Cycle Management

FMT_MTD.1/GP-LC Management of TSF Data

FMT_MTD.1.1/GP-LC The TSF shall restrict the ability to **change_default, query**[16] the **TSF data listed in *Table 6***[17] to **the authorized identified roles mentioned in *Table 6***[18].

Operations (APDUs or APIs)	**List of TSF Data: (Life Cycle State and Transitions)**	**Authorised Identified Roles**
Query (GET STATUS)	Card Life Cycle State information	ISD on behalf of the Issuer
	Application or SSD Life Cycle State information	ISD on behalf of the Issuer, AP owning the corresponding SSD or Application
	Executable Load Files Life Cycle State information	ISD on behalf of the Issuer, AP owning the corresponding ELF
	Executable Load Files and Executable Modules Life Cycle State information	ISD on behalf of the Issuer, AP owning the corresponding ELF and Modules
Change_default (SET STATUS)	Card Life Cycle State information and transitions as defined in [GPCS]	ISD on behalf of the Issuer
	Application or SSD Life Cycle State information and transitions as defined in [GPCS]	AP owning the corresponding SSD or Application
	SD and its associated Applications Life Cycle State information	AP owning the corresponding SSD and its Applications

Table 6: Life Cycle Management Operations, Data, and Roles

Application Note: Refer to the following sections in [GPCS] for additional details about Life Cycle:

[15] [assignment: additional importation control rules]
[16] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[17] [assignment: list of TSF data]
[18] [assignment: the authorized identified roles]

- Card Life Cycle states and transitions are described in [GPCS] section 5.1.
- The Executable Load File/ Executable Module Life Cycle is described in [GPCS] section 5.2.
- Application and Security Domain Life Cycle states and transitions are described in [GPCS] section 5.3.
- Authorised commands per Card Life Cycle state are detailed in [GPCS] Table 11-1.
- The GET STATUS APDU command used to query Life Cycle state information of an ISD, Executable Load File, Executable Module, Application, or SD is described in [GPCS] section 11.4.
- The SET STATUS APDU command used to change the Life Cycle state information of an ISD, Supplementary SD, or Application is described in [GPCS] section 11.10.
- The minimum security level for SET STATUS and GET STATUS is 'AUTHENTICATED' as defined in [GPCS] section 10.6.

Privileges Management

FMT_MTD.1/GP-PR Management of TSF Data

FMT_MTD.1.1/GP-PR The TSF shall restrict the ability to **modify**[19] the **TSF data listed in *Table 7***[20] to **the authorized identified roles mentioned in *Table 7***[21].

Operations (APDUs or APIs)	List of TSF Data: Privileges	Authorised Identified Roles
Modify (INSTALL [for registry update])	Privileges of an Application or SSD	SD processing the command shall be an ancestor SD with the AM privilege, or an SD with DM privilege under an ancestor SD with AM privilege
	Privileges of ISD	Only ISD

Table 7: Privileges Management Operations, Data, and Roles

Application Note: The 'Privileges Management' requirements cover all Privileges Assignment, Management, and Transition as defined in [CIC] section 3.1.1 and [GPCS] section 6.6.

Secure Communication

The purpose of an SCP is to authenticate the on-card and off-card entities and to protect the data exchanged between them with regard to Authenticity, Integrity, and/or Confidentiality.

The Secure Communication requirements cover all the SCPs defined by GlobalPlatform which are supported by the TOE:
- The symmetric key Secure Channel Protocol '02' defined in [GPCS], using 3DES cryptography
- The symmetric key Secure Channel Protocol '03' defined in [Amd D] includes services similar to SCP02; however, it uses AES rather than DES cryptography.
- The asymmetric key Secure Channel Protocol '11' defined in [Amd F] offers authentication services using an ECC-based Public Key Infrastructure (PKI) and secure messaging protection of commands and responses based on SCP03.

APDU commands belonging to SCPs are defined in the following references:
- SCP02: [GPCS] Annex E
- SCP03: [Amd D] section 7
- SCP11: [Amd F] section 6

[19] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]
[20] [assignment: list of TSF data]
[21] [assignment: the authorized identified roles]

The following references give details about the rules to be applied to SCPs:
- Rules that apply to all Secure Channel Protocols as defined in [GPCS] Chapter 10.
- Rules for handling Security Levels in [GPCS] section 10.6
- SCP02 protocol rules as defined in [GPCS] section E.1.6
- SCP03 protocol rules as defined in [Amd D] section 5.6
- SCP11 protocol rules as defined in [Amd F] section 4.8

FCS_CKM.1/GP-SCP Cryptographic key generation

FCS_CKM.1.1/GP-SCP The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **as listed in *Table 8***[22] and specified cryptographic key sizes **as listed in *Table 8***[23] that meet the following: **the standards listed in *Table 8***[24].

SCP protocol	Cryptographic algorithm	Cryptographic key sizes	Standard
SCP02	TDES 2-keys	112 bits[25]	[GPCS] section E.4.1
SCP03	AES	128, 192, 256 bits	[Amd D] section 6.2.1
SCP11	AES	128, 192, 256 bits	[Amd F] section 5.2

Table 8: Session key generation covering the supported SCPs

Application note: this SFR deals with the generation of the session keys which are used by the SCPs supported by the TOE.

FCS_COP.1/GP-SCP Cryptographic operation

FCS_COP.1.1/GP-SCP The TSF shall perform **the cryptographic operations listed in *Table 9***[26] in accordance with a specified cryptographic algorithm **as listed in *Table 9***[27] and cryptographic key sizes **as listed in *Table 9***[28] that meet the following: **the standards listed in *Table 9***[29].

SCP Protocol	Operation	Algorithm	Key Sizes	Standards
SCP02	MAC Generation/ Verification	HMAC, CMAC using TDES	112 bits[25]	FIPS 198
SCP02	Symmetric Encryption/ Decryption	TDES in CBC mode	112 bits[25]	NIST 800 67 NIST 800 38A
SCP02	Key Derivation	HMAC-based KDF, CMAC-based KDF using TDES	112 bits[25]	NIST 800 108 FIPS 198
SCP03, SCP11	Symmetric Encryption/ Decryption	AES in CBC mode	128, 192, or 256 bits	FIPS 197 NIST 800 38A

[22] [assignment: cryptographic key generation algorithm]
[23] [assignment: cryptographic key sizes]
[24] [assignment: list of standards]
[25] From 2026, symmetric keys must be configured with a minimum of 128 bits.
[26] [assignment: list of cryptographic operations]
[27] [assignment: cryptographic algorithm]
[28] [assignment: cryptographic key sizes]
[29] [assignment: list of standards]

SCP Protocol	Operation	Algorithm	Key Sizes	Standards
SCP03	MAC Generation/ Verification	CMAC AES	128, 192, or 256 bits	NIST 800 38B
SCP03	Key Derivation	CMAC-based KDF using AES	128, 192, or 256 bits	NIST 800 108 NIST 800 38B
SCP02, SCP03, SCP11	Hash Computing	SHA-256, SHA-384, SHA-512	-	ISO 10118 3 FIPS 180 4

Table 9: Cryptographic Operations covering the supported SCPs

Trusted Framework

FTP_TRP.1/GP-TF Trusted Path

FTP_TRP.1.1/GP-TF The TSF shall provide a communication path between itself and **the Target Application and the Receiving SD** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **modification and disclosure**[30].

FTP_TRP.1.2/GP-TF The TSF shall permit **the Receiving SD with the Trusted Path privilege, the Trusted Framework, and the Target Application** to initiate communication via the trusted path.

FTP_TRP.1.3/GP-TF The TSF shall require the use of the trusted path for **Application personalization: the GlobalPlatform Trusted Framework for inter-application communication forwards the unwrapped command (STORE DATA) to the Target Application indicated by the Receiving SD through its GlobalPlatform Application interface**.

GlobalPlatform Card Management: Common SFRs

FMT_MSA.1/GP Management of security attributes

FMT_MSA.1.1/GP The TSF shall enforce the **ELF Loading information flow control SFP and Data & Key Loading information flow control SFP** to restrict the ability to **perform the operations listed in Table 10 to Table 12 acting on**[31] the security attributes **mentioned in Table 10 to Table 12**[32] to **the authorized identified roles mentioned in Table 10 to Table 12**[33].

Operations (APDUs or APIs)	Security Attributes: Card Life Cycle State	Authorised Identified Roles with Privileges
DELETE Executable Load File	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD
DELETE Executable Load File and related Application(s)	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD

[30] [selection: modification, disclosure, [assignment: other types of integrity or confidentiality violation]]
[31] [selection: change_default, query, modify, delete, [assignment: other operations]]
[32] [assignment: list of security attributes]
[33] [assignment: the authorized identified roles]

Operations (APDUs or APIs)	Security Attributes: Card Life Cycle State	Authorised Identified Roles with Privileges
DELETE Application	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD
DELETE Key	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD, SD
INSTALL	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD
INSTALL [for personalisation]	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD, SD
LOAD	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD
PUT KEY	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD, SD
SELECT	OP_READY, INITIALIZED, SECURED, or CARD_LOCKED (If an SD does have the Final Application privilege)	ISD, AM SD, DM SD, SD with Final Application privilege
SET STATUS	OP_READY, INITIALIZED, SECURED, or CARD_LOCKED	ISD, AM SD, DM SD, SD
STORE DATA	OP_READY, INITIALIZED, or SECURED	ISD, AM SD, DM SD, SD
GET DATA	OP_READY, INITIALIZED, SECURED, CARD_LOCKED, or TERMINATED	ISD, AM SD, DM SD, SD
GET STATUS	OP_READY, INITIALIZED, SECURED, or CARD_LOCKED	ISD, AM SD, DM SD, SD

Table 10: GlobalPlatform Common Operations, Security Attributes, and Roles

Operations: SCP11 Commands	Used by	Security Attributes: Card Life Cycle State	Security Attributes: Minimum Security Level	Authorised Identified Roles with Privileges
GET DATA (ECKA Certificate)	SCP11a SCP11b SCP11c	OP_READY, INITIALIZED, SECURED, or CARD_LOCKED	None	ISD, AM SD, DM SD, SD
PERFORM SECURITY OPERATION	SCP11a SCP11c		None	
MUTUAL AUTHENTICATE	SCP11a SCP11c		AUTHENTICATED or ANY_AUTHENTICATED	
INTERNAL AUTHENTICATE	SCP11b		AUTHENTICATED or ANY_AUTHENTICATED	
STORE DATA (ECKA Certificate)	SCP11a SCP11b SCP11c		None	
STORE DATA (Whitelist)	SCP11a SCP11c		None	
VERIFY PIN	SCP11b		None	

Table 11: SCP11 Operations, Security Attributes, and Roles

Operations: SCP02 Commands	Security Attributes: Card Life Cycle State	Security Attributes: Minimum Security Level	Authorised Identified Roles with Privileges
INITIALIZE UPDATE	OP_READY, INITIALIZED, SECURED, or CARD_LOCKED	None	ISD, AM SD, DM SD, SD
EXTERNAL AUTHENTICATE		C-MAC	

Table 12: SCP02 Operations, Security Attributes, and Roles

Legend for Table 10 to Table 12:
ISD: Issuer Security Domain
AM SD: Security Domain with Authorized Management privilege
DM SD: Security Domain with Delegated Management privilege
SD: Other Security Domain

Application Note:
- This SFR refines and replaces FMT_MSA.1/CM of [PP-JCS]. It is extended to cover Data and Key loading Policy.
- The authorized identified roles could be off-card or on-card entities as defined in FMT_SMR.1/GP.

FMT_MSA.3/GP Security attribute initialization

FMT_MSA.3.1/GP The TSF shall enforce the **ELF Loading information flow control SFP and Data & Key Loading information flow control SFP** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/GP The TSF shall allow the **None**[34] to specify alternative initial values to override the default values when an object or information is created.

Application Note:
- This SFR refines and replaces FMT_MSA.3/CM of [PP-JCS]. It is extended to cover the Data and Key loading Policy.
- The authorized identified roles could be off-card or on-card entities as defined in FMT_SMR.1/GP.

FMT_SMR.1/GP Security roles

FMT_SMR.1.1/GP The TSF shall maintain the roles:
- **On-card: S.OPEN, S.SD (e.g. ISD, APSD, CASD), Application**
- **Off-card: Issuer, Users (e.g. VA, AP, CA) owning SDs**

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

Application Note: this SFR refines and replaces FMT_SMR.1/Installer and FMT_SMR.1/CM of [PP-JCS], applied to roles involved in card content management operations.

FMT_SMF.1/GP Specification of Management Functions

FMT_SMF.1.1/GP The TSF shall be capable of performing the following management functions **specified in [GPCS]:**

[34] [assignment: authorized identified roles]

- **Card and Application Security Management as defined in [GPCS]: Life Cycle, Privileges, Application/SD Locking and Unlocking, Card Locking and Unlocking, Card Termination, Application Status interrogation, Card Status Interrogation, command dispatch, Operational Velocity Checking.**
- **Management functions (Secure Channel Initiation/Operation/Termination) related to SCPs as defined in [GPCS].**

Application Note:
- This SFR refines and replaces FMT_SMF.1/CM of [PP-JCS].
- Management functions related to SCPs are defined in [GPCS] Chapter 10.

FPT_RCV.3/GP Automated recovery without undue loss

FPT_RCV.3.1/GP When automated recovery from **none, see application note below**[35] 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/GP For **detection of a potential loss of integrity during the transmission of an Executable Load File to the card, abortion of the installation process of an Executable Load File, or any fatal error occurred during the linking of an Executable Load File to the Executable Files already installed on the card**[36] the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3/GP The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding **the loss of the Executable Load File being loaded or installed**[37] for loss of TSF data or objects under the control of the TSF.

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

Application Note:
- This SFR refines and replaces FPT_RCV.3/Installer of [PP-JCS], applied to card content management operations
- There is no maintenance mode implemented within the TOE. Recovery is always enforced automatically as stated in FPT_RCV.3.2/GP

FPT_FLS.1/GP Failure with preservation of secure state

FPT_FLS.1.1/GP The TSF shall preserve a secure state when the following types of failures occur:
- **S.OPEN fails to load/install an Executable Load File / Application instance.**
- **S.SD fails to load SD/Application data and keys.**
- **S.OPEN fails to verify/change the Card Life Cycle, Application and SD Life Cycle states.**
- **S.OPEN fails to verify the privileges belonging to an SD or an Application.**
- **S.SD fails to verify the security level applied to protect APDU commands.**
- **None**[38]

Application Note:
- This SFR extends FPT_FLS.1/Installer of [PP-JCS] to include the failures that may occur during the loading of SD/Application keys and data.
- Refer to [JCRE310] section 11.1.5 and [GPCS] sections 11.5, 11.6, 11.8, and 11.11 for additional details.

[35] [assignment: list of failures/service discontinuities during card content management operations]
[36] [assignment: list of failures/service discontinuities during card content management operations]
[37] [assignment: quantification]
[38] [assignment: list of additional types of failures]

FPT_TDC.1/GP Inter-TSF basic TSF data consistency

FPT_TDC.1.1/GP The TSF shall provide the capability to consistently interpret **ELFs, SD/Application data and keys, data used to implement a Secure Channel, None**[39] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2/GP The TSF shall use **the list of interpretation rules to be applied by the TSF when processing the INSTALL, LOAD, PUT KEY, and STORE DATA commands sent to the card, None**[40] when interpreting the TSF data from another trusted IT product.

Application Note: the list of interpretation rules to be applied by the TSF when processing the INSTALL, LOAD, PUT KEY, and STORE DATA commands sent to the card are defined in [GPCS] sections 11.5, 11.6, 11.8, and 11.11.

FTP_ITC.1/GP Inter-TSF trusted channel

FTP_ITC.1.1/GP 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/GP The TSF shall permit **another trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3/GP The TSF shall initiate communication via the trusted channel for:

- **APDU commands sent to the card within a Secure Channel Session**
- **When loading/installing a new ELF on the card**
- **When transmitting and loading sensitive data to the card using STORE DATA or PUT KEY commands**
- **When deleting ELFs, Applications, or Keys**
- **None**[41]

Application Note: this SFR corresponds to FTP_ITC.1/CM of [PP-JCS], applied where APDU command and response integrity and/or confidentiality protection through a Secure Channel are required.

FCO_NRO.2/GP Enforced proof of origin

FCO_NRO.2.1/GP The TSF shall enforce the generation of evidence of origin for transmitted **Executable Load Files, SD/Application data and keys** [42] at all times.

Refinement: the TSF shall be able to generate an evidence of origin at all times for 'Executable Load Files, SD/Application data and keys' received from the off-card entity (originator of transmitted data) that communicates with the card.

FCO_NRO.2.2/GP The TSF shall be able to relate the **identity**[43] of the originator of the information, and the **Executable Load Files, SD/Application data and keys**[44] of the information to which the evidence applies.

Refinement: the TSF shall be able to load 'Executable Load Files, SD/Application data and keys' to the card with associated security attributes (the identity of the originator, the destination) such that the evidence of origin can be verified.

[39] [assignment: list of TSF data types]
[40] [assignment: list of interpretation rules to be applied by the TSF]
[41] [assignment: list of functions for which a trusted channel is required]
[42] [assignment: list of information types]
[43] [assignment: list of attributes]
[44] [assignment: list of information fields]

FCO_NRO.2.3/GP The TSF shall provide a capability to verify the evidence of origin of information to **the off card entity (recipient of the evidence of origin) who requested that verification** given **at the time the ELF, SD/Application data and keys are received**[45].

Application Note:
- This SFR extends FCO_NRO.2/CM of [PP-JCS] to cover the SD/Application data and keys transmitted and loaded to the card via STORE DATA and PUT KEY commands.

FIA_UID.1/GP Timing of identification

FIA_UID.1.1/GP The TSF shall allow **SD selection, Application selection, initializing a Secure Channel with the card, requesting data that identifies the card or off-card entities**[46] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2/GP The TSF shall require each user to be successfully identified before allowing any other TSF mediated actions on behalf of that user.

Application Note:
- This SFR refines and replaces FIA_UID.1/CM of [PP-JCS].

FDP_UIT.1/GP Basic data exchange integrity

FDP_UIT.1.1/GP The TSF shall enforce the **ELF Loading information flow control SFP and Data & Key Loading information flow control SFP** to **receive**[47] user data in a manner protected from **modification, deletion, insertion, replay** errors.

FDP_UIT.1.2/GP The TSF shall be able to determine on receipt of user data, whether **modification, deletion, insertion, replay** has occurred.

Application Note:
- This SFR extends FDP_UIT.1/CM of [PP-JCS] to cover the integrity protection of SD/Application data and keys.
- This SFR applies where APDU command and response integrity protection is required (e.g. INSTALL, LOAD, STORE DATA and PUT KEY commands).

FDP_ROL.1/GP Basic rollback

FDP_ROL.1.1/GP The TSF shall enforce **ELF Loading information flow control SFP and Data & Key Loading information flow control SFP** to permit the rollback of the **installation, loading, or removal operation** on the **executable files, application instances, SD/Application data and keys**.

FDP_ROL.1.2/GP The TSF shall permit operations to be rolled back within the **boundary limit**:
- **Until the Executable File or application instance has been added to or removed from the applet's registry.**
- **Until SD/Application data or keys have been added to or removed from SD or Application.**

FDP_UCT.1/GP Basic data exchange confidentiality

FDP_UCT.1.1/GP The TSF shall enforce the **ELF Loading information flow control SFP and Data & Key Loading information flow control SFP** to **receive**[48] user data in a manner protected from unauthorized disclosure.

[45] [assignment: limitations on the evidence of origin]
[46] [assignment: list of TSF-mediated actions]
[47] [selection: transmit, receive]
[48] [selection: transmit, receive]

Application Note: this SFR applies where APDU command and response confidentiality protection is required. For example, the sensitive data (e.g. secret keys) shall always be transmitted as confidential data.

FPR_UNO.1/GP Unobservability

FPR_UNO.1.1/GP The TSF shall ensure that **SDs and Applications** are unable to observe the operation: **keys or data import (PUT KEY or STORE DATA), encryption, decryption, signature generation and verification, none**[49] on **keys and data** by **the OPEN or any other SD or Application**.

FIA_UAU.1/GP Timing of authentication

FIA_UAU.1.1/GP The TSF shall allow **the TSF mediated actions listed in FIA_UID.1/GP** on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2/GP 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/GP Single-use authentication mechanisms

FIA_UAU.4.1/GP The TSF shall prevent reuse of authentication data related to **the authentication mechanism used to open a secure communication channel with the card**.

FIA_AFL.1/GP Authentication failure handling

FIA_AFL.1.1/GP The TSF shall detect when **1**[50] unsuccessful authentication attempt occur related to **the authentication of the origin of a card management operation command**.

FIA_AFL.1.2/GP When the defined number of unsuccessful authentication attempts has been **met or surpassed**, the TSF shall **close the Secure Channel**.

FMT_MTD.3/GP Secure TSF Data

FMT_MTD.3.1/GP The TSF shall ensure that only secure values are accepted for **Life Cycle states, Security Levels and Privileges in the GlobalPlatform Registry**.

Package 'Ciphered Load File Data Block (CLFDB)' - Security Functional Requirements

FCS_COP.1/GP-CLFDB Cryptographic operation

FCS_COP.1.1/GP-CLFDB The TSF shall perform **Decryption of Ciphered Load File Data Blocks** in accordance with a specified cryptographic algorithm **as mentioned in Table 13**[51] and cryptographic key sizes **as mentioned in Table 13**[52] that meet the following: **standards mentioned in Table 13**[53].

[49] [assignment: list of operations]
[50] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[51] [assignment: cryptographic algorithm]
[52] [assignment: cryptographic key sizes]
[53] [assignment: list of standards]

Algorithm	Key sizes	Standards
TDES with CBC mode	112[54] bits	[ISO/IEC 9797-1]
AES with CBC mode with a null ICV	128, 192, or 256 bits	[FIPS 197]

Table 13: Algorithms used to decrypt CLFDB

Application note: See [GPCS] section C.6.

Package 'Global Services (GS)' - Security Functional Requirements

FDP_ACC.1/GP-GS Subset access control

FDP_ACC.1.1/GP-GS The TSF shall enforce the **GlobalPlatform Services access control policy** on **the following list of subjects, objects and operations**:

- **Subject: S.OPEN, Applications with 'Global Service' privilege, other Applications.**
- **Objects:**
  - **Global Service Privilege**
  - **Service name**
  - **GlobalPlatform Registry**
  - **AID**
- **Operation controlled by the policy:**
  - **Registration of a Global Service with a unique service name**
  - **Deregistration of a Global Service with a unique service name**
  - **Access of a uniquely registered Global Service or a specific Global Services Application**

FDP_ACF.1/GP-GS Security attribute based access control

FDP_ACF.1.1/GP-GS The TSF shall enforce the **GlobalPlatform Services access control policy** to objects based on the following **Security Attributes:**

- **Global Service privilege: Assigned or Not assigned**
- **Service name: Recorded or Not recorded for an on-card entity (as provided in the INSTALL command)**
- **Service name: Registered or Not registered in the GlobalPlatform Registry**
- **AID: Associated or Not associated**

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

- **Registering/Deregistering Global Services:**
  - **S.OPEN is responsible for ensuring the uniqueness of each service name registered by Global Services Applications.**
  - **On receipt of unique service registration or deregistration request, S.OPEN checks that the requesting on-card entity has the 'Global Service' privilege.**
  - **On receipt of unique service registration request, S.OPEN checks that the requested service name is not registered in the GlobalPlatform Registry for another on-card entity.**
  - **On receipt of service deregistration request, S.OPEN checks that the requested service name is registered in GlobalPlatform Registry entry of the requesting on-card entity.**

[54] From 2026, symmetric keys must be configured with a minimum of 128 bits.

- **Application Accessing rules to Global Services: On receipt of service access request,**
  - **If the request indicates a specific service name without any associated AID, S.OPEN checks that the requested service name matches exactly with (one of) the service name(s) uniquely registered, or belongs to the same service family uniquely registered.**
  - **If the request indicates a specific AID, S.OPEN checks that the on-card entity identified in the request has the 'Global Service' privilege, and that the requested service name matches exactly with (one of) the service name(s) recorded for that on-card entity, or belongs to (one of) the same service family(ies) recorded for that on-card entity.**
  - **S.OPEN identifies the corresponding Global Services Application.**
  - **S.OPEN obtains the GlobalPlatform Service interface of the corresponding Global Services Application and forwards it to the requesting on-card entity.**
- **None**[55]

FDP_ACF.1.3/GP-GS The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **None**[56].

FDP_ACF.1.4/GP-GS The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **None**[57].

Application Note: Global Services Applications are described in [GPCS] section 8.1.

FMT_MSA.1/GP-GS Management of security attributes

FMT_MSA.1.1/GP-GS The TSF shall enforce the **GlobalPlatform Services access control policy** to restrict the ability to **query, modify** [58] the security attributes **defined in FDP_ACF.1.1/GP-GS** to the **S.OPEN**.

FMT_MSA.3/GP-GS Security attribute initialization

FMT_MSA.3.1/GP-GS The TSF shall enforce the **GlobalPlatform Services access control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/GP-GS The TSF shall allow the **S.OPEN** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMR.1/GP-GS Security roles

FMT_SMR.1.1/GP-GS The TSF shall maintain the roles **S.OPEN, Global Services Application**.

FMT_SMR.1.2/GP-GS The TSF shall be able to associate users with roles.

FMT_SMF.1/GP-GS Specification of Management Functions

FMT_SMF.1.1/GP-GS The TSF shall be capable of performing the following management functions:

- **Management of Global Services Applications (Registering, Deregistering, Accessing)**

[55] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[56] [assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects]
[57] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]
[58] [selection: change_default, query, modify, delete, [assignment: other operations]]

- **none**[59]

Application Note: Global Services Applications are described in [GPCS] section 8.1.

Package 'Cardholder Verification Method (CVM)' - Security Functional Requirements

FIA_AFL.1/GP-CVM Authentication failure handling

FIA_AFL.1.1/GP-CVM The TSF shall detect when **an administrator configurable positive integer within the [1-255]**[60] unsuccessful authentication attempts occur related to **user authentication using CVM**.

FIA_AFL.1.2/GP-CVM When the defined number of unsuccessful authentication attempts has been **met**[61], the TSF shall **block the usage of the Global PIN**[62].

FPR_UNO.1/GP-CVM Unobservability

FPR_UNO.1.1/GP-CVM The TSF shall ensure that **all users and subjects**[63] are unable to observe the operation **comparison** on **Global PIN** by **S.OPEN**[64].

Package 'Delegated Management (DM)' - Security Functional Requirements

FCO_NRR.1/GP-RECEIPT Selective proof of receipt

FCO_NRR.1.1/GP-RECEIPT The TSF shall be able to generate evidence of receipt for received **card management operation requests** at the request of the **originator**.

FCO_NRR.1.2/GP-RECEIPT The TSF shall be able to relate the **Confirmation Data** of the recipient of the information, and the parameters of **the card management operation request** of the information to which the evidence applies.

FCO_NRR.1.3/GP-RECEIPT The TSF shall provide a capability to verify the evidence of receipt of information to **recipient** given **none**.

Application Note:
- The confirmation data are described in [GPCS] section 11.1.6.
- The parameters of the card management operation request are described in [GPCS] section C.5.

[59] [assignment: list of management functions to be provided by the TSF]
[60] [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
[61] [selection: met, surpassed]
[62] [assignment: list of actions]
[63] [assignment: list of users and/or subjects]
[64] [assignment: list of protected users and/or subjects]

FCO_NRO.2/GP-TOKEN Enforced proof of origin

FCO_NRO.2.1/GP-TOKEN The TSF shall enforce the generation of evidence of origin for transmitted **'ELF with Token Verification', as mentioned in the refinement below**[65] at all times.

Refinement: The TSF shall be able to generate an evidence of origin at all times for 'ELF with Token Verification' received from the off-card entity (originator of transmitted data) that communicates with the card.

FCO_NRO.2.2/GP-TOKEN The TSF shall be able to relate the **token present in the card management operation request, as mentioned in the refinement below**[66] of the originator of the information, and the **'ELF with Token Verification', as mentioned in the refinement below**[67] of the information to which the evidence applies.

Refinement: the TSF shall be able to load 'ELF with Token Verification' to the card with associated security attributes (token present in the card management operation request) such that the authenticity of transmitted data can be verified.

FCO_NRO.2.3/GP-TOKEN The TSF shall provide a capability to verify the evidence of origin of information to **the off-card entity (recipient of the evidence of origin) requesting that verification** given **at the time the ELF with Token is received**.

Application Note: the parameters of the card management operation request are described in [GPCS] section C.4.

FCS_COP.1/GP-TOKEN Cryptographic operation

FCS_COP.1.1/GP-TOKEN The TSF shall perform **the verification of the Token signature attached to card management commands** in accordance with a specified cryptographic algorithm **as mentioned in Table 14**[68] and cryptographic key sizes **as mentioned in Table 14**[69] that meet the following: **standards mentioned in Table 14**[70].

Algorithm	Key sizes	Recommended Standards
TDES	112 bits[71]	[GPCS] section B.1.2.2, Annex C.4 'Tokens'
AES	128, 192, or 256 bits	[GPCS] section B.2.2, Annex C.4 'Tokens'
RSA	1024 or 2048 bits[72]	[GPCS] section B.3.1.1 or B3.2.1, Annex C.4 'Tokens'
ECC	256, 384, or 512 bits	[GPCS] section B.4.3, Annex C.4 'Tokens'

Table 14: Algorithms Used to Verify the Token Signature

[65] [assignment: list of information types]
[66] [assignment: list of attributes]
[67] [assignment: list of information fields]
[68] [assignment: cryptographic algorithm]
[69] [assignment: cryptographic key sizes]
[70] [assignment: list of standards]
[71] From 2026, symmetric keys must be configured with a minimum of 128 bits.
[72] The minimum length of the module must be configured to 2048 bits, until 2030. From 2031, the minimum size is 3072 bits.

FCS_COP.1/GP-RECEIPT Cryptographic operation

FCS_COP.1.1/GP-RECEIPT The TSF shall perform **the generation of the Receipt signature attached to responses to card management commands** in accordance with a specified cryptographic algorithm **as mentioned in Table 15**[73] and cryptographic key sizes **as mentioned in Table 15**[74] that meet the following: **standards mentioned in Table 15**[75].

Algorithm	Key sizes	Recommended Standards
TDES	112 bits[76]	[GPCS] section B.1.2.2, Annex C.5 'Receipts'
AES	128, 192, or 256 bits	[GPCS] section B.2.2, Annex C.5 'Receipts'
RSA	1024 or 2048 bits[77]	[GPCS] section B.3.1.1 or B3.2.1, Annex C.5 'Receipts'
ECC	256, 384, or 512 bits	[GPCS] section B.4.3, Annex C.5 'Receipts'

Table 15: Algorithms Used to Generate the Receipt Signature

Packages 'DAP Verification' & 'Mandated DAP Verification' - Security Functional Requirements

FCS_COP.1/GP-DAP_SHA Cryptographic operation

FCS_COP.1.1/GP-DAP_SHA The TSF shall perform **computation of a hash value for DAP Verification** in accordance with a specified cryptographic algorithm **SHA-1[78], SHA-256, SHA-384, or SHA-512**[79] and cryptographic key sizes **SHA-1, SHA-256, SHA-384, or SHA-512 hash lengths**[80] that meet the following: **[NIST 800 57]**[81].

Application Note: refer to the description in [GPCS] section C.3 for more details.

FCS_COP.1/GP-DAP_VER Cryptographic operation

FCS_COP.1.1/GP-DAP_VER The TSF shall perform **verification of the DAP signature attached to Load Files** in accordance with a specified cryptographic algorithm **as mentioned in Table 16**[82] and cryptographic key sizes **as mentioned in Table 16**[83] that meet the following: **standards mentioned in Table 16**[84].

[73] [assignment: cryptographic algorithm]
[74] [assignment: cryptographic key sizes]
[75] [assignment: list of standards]
[76] From 2026, symmetric keys must be configured with a minimum of 128 bits.
[77] The minimum length of the module must be configured to 2048 bits, until 2030. From 2031, the minimum size is 3072 bits.
[78] SHA-1 is vulnerable to collision attacks. Its usage must be limited to contexts with no risk of collision attacks.
[79] [assignment: cryptographic algorithm]
[80] [assignment: cryptographic key sizes]
[81] [assignment: list of standards]
[82] [assignment: cryptographic algorithm]
[83] [assignment: cryptographic key sizes]
[84] [assignment: list of standards]

Algorithm	Key sizes	Recommended Standards
TDES	112 bits[85]	[ISO/IEC 9797-1]
AES	128, 192, or 256 bits	[NIST 800 38B]
RSA	1024 or 2048 bits[86]	[PKCS#1]
ECC	256, 384, or 512 bits	[ANSI X9.62]

Table 16: Algorithms Used to Verify the DAP Signature

Application Note: refer to the description in [GPCS] section C.3 for more details.

FCO_NRO.2/GP-DAP Enforced proof of origin

FCO_NRO.2.1/GP-DAP The TSF shall enforce the generation of evidence of origin for transmitted **'ELF with DAP', as mentioned in the refinement below**[87] at all times.

Refinement: the TSF shall be able to generate an evidence of origin at all times for 'ELF with DAP' received from the off-card entity (originator of transmitted data) that communicates with the card.

FCO_NRO.2.2/GP-DAP The TSF shall be able to relate the **Load File Data Block Signature, as mentioned in the refinement below**[88] of the originator of the information, and the **'ELF with DAP', as mentioned in the refinement below**[89] of the information to which the evidence applies.

Refinement: the TSF shall be able to load 'ELF with DAP' to the card with associated security attributes (Load File Data Block Signature) such that the integrity and authenticity of transmitted data can be verified.

FCO_NRO.2.3/GP-DAP The TSF shall provide a capability to verify the evidence of origin of information to **the off-card entity (recipient of the evidence of origin) who requested that verification** given **at the time the ELF with DAP is received**.

Application Note: this SFR addresses the DAP verification as defined in [GPCS] sections 9.2.1, 11.6.2.3, and C.3.

PP-Module 'Amendment H: Executable Load File Upgrade (ELFU)' - Security Functional Requirements

FDP_ACC.1/GP-ELFU Subset access control

FDP_ACC.1.1/GP-ELFU The TSF shall enforce the **ELF Upgrade Access Control Policy** on **the following list of subjects, objects and operations**:

- **Subjects: S.OPEN, ELF Provider, S.SD**
- **Objects: Application instance data, ELF, ELF Registry data, ELF session data**

[85] From 2026, symmetric keys must be configured with a minimum of 128 bits.
[86] The minimum length of the module must be configured to 2048 bits, until 2030. From 2031, the minimum size is 3072 bits.
[87] [assignment: list of information types]
[88] [assignment: list of attributes]
[89] [assignment: list of information fields]

- **Operation controlled by the policy: APDUs 'MANAGE ELF UPGRADE', INSTALL [for load] and LOAD, and Upgrade API methods.**

Application Note:
- The APDU 'MANAGE ELF UPGRADE' is defined in [Amd H] section 4.1.
- The INSTALL [for load], LOAD commands, and Upgrade API methods are defined in [Amd H] Annex A.

FDP_ACF.1/GP-ELFU Security attribute based access control

FDP_ACF.1.1/GP-ELFU The TSF shall enforce the **ELF Upgrade Access Control Policy** to objects based on the following **Security Attributes: AIDs, ELF session status, ELF versions (old or new).**

FDP_ACF.1.2/GP-ELFU The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:
- **Only a single ELF Upgrade Session is processed at a time. No new ELF Upgrade Session may be started until the previous one (if any) has been completed or aborted.**
- **The MANAGE ELF UPGRADE [start] command is rejected with an error and the ELF Upgrade Process is aborted if any of the conditions defined in [Amd H] are satisfied.**
- **S.OPEN allows an ELF upgrade session to be initiated if no other ELF upgrade session is running.**
- **S.OPEN allows an ELF upgrade session to be initiated if processing S.SD has authorized management privilege or delegate management privilege**[90]

FDP_ACF.1.3/GP-ELFU The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none**[91].
FDP_ACF.1.4/GP-ELFU The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none**[92].

Application Note:
- AIDs, ELF session status are given in [Amd H] Table 4-8.
- Rules to be applied when starting the Upgrade session are described in [Amd H] section 3.2.1.
- Rules to be applied during the Saving phase are described in [Amd H] section 3.2.2.
- Rules to be applied during the Loading phase are described in [Amd H] section 3.2.3.
- Rules to be applied during the Restore phase are described in [Amd H] section 3.2.4.
- Card Content Management Operations described in [Amd H] section 3.4 shall always be rejected during an ELF Upgrade Session.

FDP_ROL.1/GP-ELFU Basic rollback

FDP_ROL.1.1/GP-ELFU The TSF shall enforce **ELF Upgrade Access Control Policy** to permit the rollback of the **deletion** on the **Application instances and ELF(s).**

FDP_ROL.1.2/GP-ELFU The TSF shall permit operations to be rolled back within the **boundary limit**:

[90] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[91] [assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects]
[92] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

- **If the deletion of the application instances and ELF(s) (atomic and irreversible operation) was started and then interrupted and/or disturbed by for example unexpected power-down, it shall automatically restart and complete at next power-up.**
- **If the interruption occurred during the Deletion Sequence and the latter did not complete automatically (i.e. the irreversible deletion operation did not start already), the Deletion Sequence shall restart.**

FMT_MSA.1/GP-ELFU	Management of security attributes

FMT_MSA.1.1/GP-ELFU The TSF shall enforce the **ELF Upgrade Access Control Policy** to restrict the ability to **set and maintain** the security attributes **defined in FDP_ACF.1.1/GP-ELFU** to the **S.OPEN**.

FMT_MSA.3/GP-ELFU	Security attribute initialization

FMT_MSA.3.1/GP-ELFU The TSF shall enforce the **ELF Upgrade Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/GP-ELFU The TSF shall allow the **S.OPEN** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/GP-ELFU	Specification of Management Functions

FMT_SMF.1.1/GP-ELFU The TSF shall be capable of performing the following management functions:

- **The Saving, Loading, Restore phases of the Executable Load File Process**
- **Management of the ELF upgrade session status**
- **Card management during the ELF upgrade session**
- **None**[93]

FPT_FLS.1/GP-ELFU	Failure with preservation of secure state

FPT_FLS.1.1/GP-ELFU The TSF shall preserve a secure state when the following types of failures occur:

- **The required minimum amount of memory is not available at the time the command MANAGE ELF UPGRADE is received,**
- **A fatal error occurs using the new ELF version during the Restore Phase**
- **The ELF Upgrade Recovery Procedure fails,**
- **The installation of an Application instance fails,**
- **An interruption occurred during the Installation, Saving, Restore, or Consolidation Sequences,**
- **none**[94]**.**

[93] [assignment: list of management functions to be provided by the TSF]
[94] [assignment: list of types of failures in the TSF]

PP-Module 'OS Update' - Security Functional Requirements

FDP_ACC.1/OS-UPDATE Subset access control

FDP_ACC.1.1/OS-UPDATE The TSF shall enforce the **OS Update Access Control Policy** on **the following list of subjects, objects, and operations**:

- **Subjects: S.OS-DEVELOPER is the representative of the OS Developer within the TOE, being responsible for signature verification and decryption of the additional code, before Loading, Installation and Activation are authorized.**
- **Objects: additional code and associated cryptographic signature**
- **Operations: loading, installation, and activation of additional code**

FDP_ACF.1/OS-UPDATE Security attribute based access control

FDP_ACF.1.1/OS-UPDATE The TSF shall enforce the **OS Update Access Control Policy** to objects based on the following Security Attributes:

- **The additional code cryptographic signature verification status**
- **The Identification Data verification status (between the Initial TOE and the additional code)**

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

- **The verification of the additional code cryptographic signature (using D.OS-UPDATE_SGNVER-KEY) by S.OS-DEVELOPER is successful.**
- **The decryption of the additional code prior installation (using D.OS-UPDATE_DEC-KEY) by S.OS-DEVELOPER is successful.**
- **The comparison between the identification data of both the Initial TOE and the additional code demonstrates that the OS Update operation can be performed.**
- **none**[95]

FDP_ACF.1.3/OS-UPDATE The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none**[96].

FDP_ACF.1.4/OS-UPDATE The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **none**[97].

Application Note:

- Identification data verification is necessary to ensure that the received additional code is actually targeting the TOE and that its version is compatible with the TOE version.
- Confidentiality protection must be enforced when the additional code is transmitted to the TOE for loading (See OE.OS-UPDATE-ENCRYPTION). Confidentiality protection is achieved through direct encryption of the additional code.

[95] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]
[96] [assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects]
[97] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FMT_MSA.3/OS-UPDATE	Security attribute initialization

FMT_MSA.3.1/OS-UPDATE The TSF shall enforce the **OS Update Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

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

Application Note: the additional code signature verification status must be set to “Fail” by default. This prevents installation of any additional code until the additional code signature is successfully verified by the TOE.

FMT_SMR.1/OS-UPDATE	Security roles

FMT_SMR.1.1/OS-UPDATE The TSF shall maintain the roles **OS Developer, Issuer**.

FMT_SMR.1.2/OS-UPDATE The TSF shall be able to associate users with roles.

FMT_SMF.1/OS-UPDATE	Specification of Management Functions

FMT_SMF.1.1/OS-UPDATE The TSF shall be capable of performing the following management functions: **activation of additional code**.

Application Note: once verified and installed, additional code needs to be activated to become effective.

FIA_ATD.1/OS-UPDATE	User attribute definition

FIA_ATD.1.1/OS-UPDATE The TSF shall maintain the following list of security attributes belonging to individual users: **additional code ID for each activated additional code**.

Refinement: "Individual users" stands for additional code.

FTP_TRP.1/OS-UPDATE	Trusted Path

FTP_TRP.1.1/OS-UPDATE The TSF shall provide a communication path between itself and **remote** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **none**[98].

FTP_TRP.1.2/OS-UPDATE The TSF shall permit **remote users** to initiate communication via the trusted path.

FTP_TRP.1.3/OS-UPDATE The TSF shall require the use of the trusted path for **the transfer of the additional code to the TOE**.

Application Note: during the transmission of the additional code to the TOE for loading, the confidentiality is ensured through direct encryption of the additional code, hence the ‘none’ selection in FTP_TRP.1.1/OS-UPDATE.

[98] [selection: disclosure, none]

FCS_COP.1/OS-UPDATE-DEC	Cryptographic operation

FCS_COP.1.1/OS-UPDATE-DEC The TSF shall perform **Decryption of the additional code prior installation** in accordance with a specified cryptographic algorithm **AES in CBC mode with null IV**[99] and cryptographic key sizes **128 bits**[100] that meet the following: **FIPS 197**[101].

FCS_COP.1/OS-UPDATE-VER	Cryptographic operation

FCS_COP.1.1/OS-UPDATE-VER The TSF shall perform **digital signature verification of the additional code to be loaded** in accordance with a specified cryptographic algorithm **AES-CMAC**[102] and cryptographic key sizes **128 bits**[103] that meet the following: **FIPS 197 and SP800-38B**[104].

FPT_FLS.1/OS-UPDATE	Failure with preservation of secure state

FPT_FLS.1.1/OS-UPDATE The TSF shall preserve a secure state when the following types of failures occur: **interruption or incident which prevents the forming of the Updated TOE**.

Application Note:

- The OS Update operation must either be successful or fail securely. There are 3 steps in an OS Update operation:
  - step 1: loading
  - step 2: activation
  - step 3: update of TOE identification data

  Steps 2 and 3 are performed atomically, so that the TOE active code and identification data always remain consistent.
- If a failure (interruption or incident) occurs during step 1 (loading), then the TOE remains in its initial state (no update, neither of code nor of the TOE identification data).
- If a failure (interruption or incident) occurs during the atomic sequence step 2 / step 3 (activation / update of TOE identification data), then the enforced behavior depends on the nature of the update:
  - For java code updates, the TOE remains in its initial state and the OS Update operation is aborted.
  - For native code updates, the TOE does some retries to complete the atomic sequence step 2 / step 3 (activation / update of TOE identification data) until it is successful.
  - In any case, only two possible secure states are possible at any given time:
    - Either activation is not done and the TOE identification data is not updated (i.e. initial state)
    - Or the atomic sequence completes successfully, i.e. the OS update is activated and the TOE identification data is updated accordingly.

9.1.3 [PP-JCS] Protection Profile

This section states the security functional requirements for the Java Card System - Open configuration. For readability, requirements are arranged into groups. All the groups defined in the table below come from [PP-JCS].

[99] [assignment: cryptographic algorithm]
[100] [assignment: cryptographic key sizes]
[101] [assignment: list of standards]
[102] [assignment: cryptographic algorithm]
[103] [assignment: cryptographic key sizes]
[104] [assignment: list of standards]

Group	Name	Description
CoreG_LC	Core with Logical Channels	The CoreG_LC contains the requirements concerning the runtime environment of the Java Card System implementing logical channels. This includes the firewall policy and the requirements related to the Java Card API. Logical channels are a Java Card specification version 2.2 feature.
ADELG	Applet deletion	The ADELG contains the security requirements for erasing installed applets from the card, a feature introduced in Java Card specification version 2.2.
ODELG	Object deletion	The ODELG contains the security requirements for the object deletion capability. This provides a safe memory recovering mechanism. This is a Java Card specification version 2.2 feature.

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

Subjects (prefixed with an "S") are described in the following table:

Subject	Description
S.ADEL	The applet deletion manager which also acts on behalf of the card issuer. It may be an applet ([JCRE310], §11), but its role asks anyway for a specific treatment from the security viewpoint.
S.APPLET	Any applet instance.
S.BCV	The bytecode verifier (BCV), which acts on behalf of the verification authority who is in charge of the bytecode verification of the CAP files.
S.CAD	The CAD represents off-card entity that communicates with the S.INSTALLER. If the TOE provides JCRMI functionality, CAD can request RMI services by issuing commands to the card.
S.INSTALLER	The installer is the on-card entity which acts on behalf of the card issuer. This subject is involved in the loading of CAP files and installation of applets.
S.JCRE	The runtime environment under which Java programs in a smart card are executed.
S.JCVM	The bytecode interpreter that enforces the firewall at runtime.
S.LOCAL	Operand stack of a JCVM frame, or local variable of a JCVM frame containing an object or an array of references.
S.MEMBER	Any object's field, static field or array position.
S.CAP_FILE	A CAP file may contain multiple Java language packages. A package is a namespace within the Java programming language that may contain classes and interfaces. A CAP file may contain packages that define either user library, or one or several applets. A COMPACT CAP file as specified in Java Card Specifications version 3.1 or CAP files compliant to previous versions of Java Card Specification, MUST contain only a single package representing a library or one or more applets.

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

Object	Description
O.APPLET	Any installed applet, its code and data.
O.CODE_CAP_FILE	The code of a CAP file, including all linking information. On the Java Card platform, a CAP file is the installation unit.
O.JAVAOBJECT	Java class instance or array. It should be noticed that KEYS, PIN, arrays and applet instances are specific objects in the Java programming language.

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

Information	Description
I.APDU	Any APDU sent to or from the card through the communication channel.
I.DATA	JCVM Reference Data: objectref addresses of APDU buffer, JCRE-owned instances of APDU class and byte array for install method.

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

Security attribute	Description / Value
Active Applets	The set of the active applets' AIDs. An active applet is an applet that is selected on at least one of the logical channels.
Applet Selection Status	"Selected" or "Deselected".
Applet's version number	The version number of an applet indicated in the export file.
CAP File AID	The AID of a CAP file.
Context	CAP file AID or "Java Card RE".
Currently Active Context	CAP file AID or "Java Card RE".
Dependent package AID	Allows the retrieval of the package AID and Applet's version number ([JCVM310], §4.5.2).
LC Selection Status	Multiselectable, Non-multiselectable or "None".
LifeTime	CLEAR_ON_DESELECT or PERSISTENT (*).
Owner	The Owner of an object is either the applet instance that created the object or the CAP file (library) where it has been defined (these latter objects can only be arrays that initialize static fields of the CAP file). The owner of a remote object is the applet instance that created the object.
Package AID	The AID of each package indicated in the export file.
Registered Applets	The set of AID of the applet instances registered on the card.
Resident CAP files	The set of AIDs of the CAP files already loaded on the card.
Resident packages	The set of AIDs of the packages already loaded on the card.
Selected Applet Context	CAP file AID or "None".
Sharing	Standard, SIO, Array View, Java Card RE entry point or global array.
Static References	Static fields of a CAP file may contain references to objects. The Static References attribute records those references.

(*) Transient objects of type CLEAR_ON_RESET behave like persistent objects in that they can be accessed only when the Currently Active Context is the object's context.

Operations (prefixed with "OP") are described in the following table. Each operation has parameters given between brackets, among which there is the "accessed object", the first one, when applicable. Parameters may be seen as security attributes that are under the control of the subject performing the operation.

Operation	Description
OP.ARRAY_ACCESS (O.JAVAOBJECT, field)	Read/Write an array component.
OP.ARRAY_LENGTH (O.JAVAOBJECT, field)	Get length of an array component.
OP.ARRAY_T_ALOAD(O.JAVAOBJECT, field)	Read from an array component.
OP.ARRAY_T_ASTORE(O.JAVAOBJECT, field)	Write to an array component.
OP.ARRAY_AASTORE(O.JAVAOBJECT, field)	Store into reference array component.
OP.CREATE(Sharing, LifeTime) (*)	Creation of an object (new, makeTransient or createArrayView call).
OP.DELETE_APPLET(O.APPLET,...)	Delete an installed applet and its objects, either logically or physically.
OP.DELETE_CAP_FILE(O.CODE_CAP_FILE,...)	Delete a CAP file, either logically or physically.
OP.DELETE_CAP_FILE_APPLET(O.CODE_CAP_FILE,...)	Delete a CAP file and its installed applets, either logically or physically.
OP.INSTANCE_FIELD(O.JAVAOBJECT, field)	Read/Write a field of an instance of a class in the Java programming language.
OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1,...)	Invoke a virtual method (either on a class instance or an array object).
OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1,...)	Invoke an interface method.
OP.JAVA(...)	Any access in the sense of [JCRE310], §6.2.8. It stands for one of the operations OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW, OP.TYPE_ACCESS, OP.ARRAY_LENGTH
OP.PUT(S1,S2,I)	Transfer a piece of information I from S1 to S2.
OP.THROW(O.JAVAOBJECT)	Throwing of an object (athrow, see [JCRE310], §6.2.8.7).

OP.TYPE_ACCESS (O.JAVAOBJECT, class)	Invoke checkcast or instance of on an object in order to access to classes (standard or shareable interfaces objects).

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

CoreG_LC Security Functional Requirements

This group is focused on the main security policy of the Java Card System, known as the firewall.

Firewall Policy

FDP_ACC.2/FIREWALL Complete access control

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

Refinement: the operations involved in the policy are: OP.CREATE, OP.INVK_INTERFACE, OP.INVK_VIRTUAL, OP.JAVA, OP.THROW, OP.TYPE_ACCESS, OP.ARRAY_LENGTH, OP.ARRAY_T_ALOAD, OP.ARRAY_T_ASTORE, OP.ARRAY_AASTORE.

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

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

FDP_ACF.1/FIREWALL Security attribute based access control

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

Subject / Object	Security attributes
S.CAP_FILE	LC Selection Status
S.JCVM	Active Applets, Currently Active Context
S.JCRE	Selected Applet Context
O.JAVAOBJECT	Sharing, Context, LifeTime

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

- **R.JAVA.1 ([JCRE310], §6.2.8): S.CAP_FILE may freely perform OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW or OP.TYPE_ACCESS upon any O.JAVAOBJECT whose Sharing attribute has value "JCRE entry point" or "global array".**
- **R.JAVA.2 ([JCRE310], §6.2.8): S.CAP_FILE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE or OP.THROW upon any O.JAVAOBJECT whose Sharing attribute has value "Standard" and whose Lifetime attribute has value "PERSISTENT" only if O.JAVAOBJECT's Context attribute has the same value as the active context.**
- **R.JAVA.3 ([JCRE310], §6.2.8.10): S.CAP_FILE may perform OP.TYPE_ACCESS upon an O.JAVAOBJECT with Context attribute different from the currently active context, whose Sharing attribute has value "SIO" only if O.JAVAOBJECT is being cast into (checkcast) or is being verified as being an instance of (instanceof) an interface that extends the Shareable interface.**
- **R.JAVA.4 ([JCRE310], §6.2.8.6): S.CAP_FILE may perform OP.INVK_INTERFACE upon an O.JAVAOBJECT with Context attribute different from the currently active context, whose Sharing attribute has the value "SIO", and whose Context attribute has the value**

"CAP File AID ", only if the invoked interface method extends the Shareable interface and one of the following conditions applies:

a) The value of the attribute Selection Status of the CAP file whose AID is "CAP File AID" is "Multiselectable",

b) The value of the attribute Selection Status of the CAP file whose AID is "CAP File AID " is "Non-multiselectable", and either "CAP File AID" is the value of the currently selected applet or otherwise "CAP File AID" does not occur in the attribute Active Applets.

- **R.JAVA.5: S.CAP_FILE may perform OP.CREATE upon O.JAVAOBJECT only if the value of the Sharing parameter is "Standard" or "SIO".**
- **R.JAVA.6 ([JCRE310], §6.2.8): S.CAP_FILE may freely perform OP.ARRAY_ACCESS or OP.ARRAY_LENGTH upon any O.JAVAOBJECT whose Sharing attribute has value "global array".**

FDP_ACF.1.3/FIREWALL The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

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

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

1) **Any subject with OP.JAVA upon an O.JAVAOBJECT whose LifeTime attribute has value "CLEAR_ON_DESELECT" if O.JAVAOBJECT's Context attribute is not the same as the Selected Applet Context.**
2) **Any subject attempting to create an object by the means of OP.CREATE and a "CLEAR_ON_DESELECT" LifeTime parameter if the active context is not the same as the Selected Applet Context.**
3) **S.CAP_FILE performing OP.ARRAY_AASTORE of the reference of an O.JAVAOBJECT whose sharing attribute has value "global array" or "Temporary".**
4) **S.CAP_FILE performing OP.PUTFIELD or OP.PUTSTATIC of the reference of an O.JAVAOBJECT whose sharing attribute has value "global array" or "Temporary".**
5) **R.JAVA.7 ([JCRE310], §6.2.8.2): S.CAP_FILE performing OP.ARRAY_T_ASTORE into an array view without ATTR_WRITABLE_VIEW access attribute.**
6) **R.JAVA.8 ([JCRE310], §6.2.8.2):S.CAP_FILE performing OP.ARRAY_T_ALOAD into an array view without ATTR_READABLE_VIEW access attribute.**

Application note, FDP_ACF.1.4/FIREWALL:

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

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

The Sharing attribute defines five categories of objects:

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

When a new object is created, it is associated with the Currently Active Context. But the object is owned by the applet instance within the Currently Active Context when the object is instantiated ([JCRE310], §6.1.3). An object is owned by an applet instance, by the JCRE or by the library where it has been defined (these latter objects can only be arrays that initialize static fields of CAP files).

([JCRE310], Glossary) Selected Applet Context. The Java Card RE keeps track of the currently selected Java Card applet. Upon receiving a SELECT command with this applet's AID, the Java Card RE makes this applet the Selected Applet Context. The Java Card RE sends all APDU commands to the Selected Applet Context.

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

([JCRE310], §6.1.2.1) At any point in time, there is only one active context within the Java Card VM (this is called the Currently Active Context).

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

It should be noticed that the Java Card platform, version 2.2.x and version 3.x.x Classic Edition, introduces the possibility for an applet instance to be selected on multiple logical channels at the same time, or accepting other applets belonging to the same CAP file being selected simultaneously. These applets are referred to as multiselectable applets. Applets that belong to a same CAP file are either all multiselectable or not ([JCVM310], §2.2.5). Therefore, the selection mode can be regarded as an attribute of CAP files. No selection mode is defined for a library CAP file.

An applet instance will be considered an active applet instance if it is currently selected in at least one logical channel. An applet instance is the currently selected applet instance only if it is processing the current command. There can only be one currently selected applet instance at a given time. ([JCRE310], §4).

FDP_IFC.1/JCVM Subset information flow control

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

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

FDP_IFF.1/JCVM Simple security attributes

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

Subjects	Security attributes
S.JCVM	Currently Active Context

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

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

FDP_IFF.1.3/JCVM The TSF shall enforce the **No additional rules**[105].

FDP_IFF.1.4/JCVM The TSF shall explicitly authorize an information flow based on the following rules: **No additional rules**[106].

FDP_IFF.1.5/JCVM The TSF shall explicitly deny an information flow based on the following rules: **No additional rules**[107].

Application note:

The storage of temporary Java Card RE-owned objects references is runtime-enforced ([JCRE310], §6.2.8.1-3).

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

FDP_RIP.1/OBJECTS Subset residual information protection

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

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

FMT_MSA.1/JCRE Management of security attributes

FMT_MSA.1.1/JCRE The TSF shall enforce the **FIREWALL access control SFP** to restrict the ability to **modify** the security attributes **Selected Applet Context** to **the Java Card RE**.

Application note: the modification of the Selected Applet Context should be performed in accordance with the rules given in [JCRE310], §4 and [JCVM310], §3.4.

FMT_MSA.1/JCVM Management of security attributes

FMT_MSA.1.1/JCVM The TSF shall enforce the **FIREWALL access control SFP and the JCVM information flow control SFP** to restrict the ability to **modify** the security attributes **Currently Active Context and Active Applets** to **the Java Card VM (S.JCVM)**.

Application note: the modification of the Currently Active Context should be performed in accordance with the rules given in [JCRE310], §4 and [JCVM310], §3.4.

[105] [assignment: additional information flow control SFP rules]
[106] [assignment: rules, based on security attributes, that explicitly authorize information flows]
[107] [assignment: rules, based on security attributes, that explicitly deny information flows]

FMT_MSA.2/FIREWALL_JCVM Secure security attributes

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

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

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

FMT_MSA.3/FIREWALL Static attribute initialization

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

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

Application note, FMT_MSA.3.1/FIREWALL:

Objects' security attributes of the access control policy are created and initialized at the creation of the object or the subject. Afterwards, these attributes are no longer mutable (FMT_MSA.1/JCRE). At the creation of an object (OP.CREATE), the newly created object, assuming that the FIREWALL access control SFP permits the operation, gets its Lifetime and Sharing attributes from the parameters of the operation; on the contrary, its Context attribute has a default value, which is its creator's Context attribute and AID respectively ([JCRE310], §6.1.3). There is one default value for the Selected Applet Context that is the default applet identifier's Context, and one default value for the Currently Active Context that is "Java Card RE".
The knowledge of which reference corresponds to a temporary entry point object or a global array and which does not is solely available to the Java Card RE (and the Java Card virtual machine).

Application note, FMT_MSA.3.2/FIREWALL:

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

FMT_MSA.3/JCVM Static attribute initialization

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

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

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: **modify the Currently Active Context, the Selected Applet Context and the Active Applets**.

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles**:**
- **Java Card RE (JCRE),**
- **Java Card VM (JCVM)**.

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

Application Programming Interface

The following SFRs are related to the Java Card API.

The whole set of cryptographic algorithms is generally not implemented because of limited memory resources and/or limitations due to exportation. Therefore, the following requirements only apply to the implemented subset.

It should be noticed that the execution of the additional native code is not within the TSF. Nevertheless, access to API native methods from the Java Card System is controlled by TSF because there is no difference between native and interpreted methods in their interface or invocation mechanism.

FCS_CKM.1/TDES Cryptographic key generation

FCS_CKM.1.1/TDES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **TDES Key generation**[108] and specified cryptographic key sizes **112 bits for TDES 2 keys**[109]**, 168 bits for TDES 3 keys**[110] that meet the following: **none (random numbers generation)**[111].

Application note: the keys are generated and diversified in accordance with [JCAPI310] in class KeyBuilder (buildKey method).

FCS_CKM.1/AES Cryptographic key generation

FCS_CKM.1.1/AES The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **AES Key generation**[112] and specified cryptographic key sizes **128, 192 and 256 bits**[113] that meet the following: **none (random numbers generation)**[114].

Application note: the keys are generated and diversified in accordance with [JCAPI310] in class KeyBuilder (buildKey method).

FCS_CKM.1/RSA Cryptographic key generation

FCS_CKM.1.1/RSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **RSA Standard and RSA CRT Key Pair Generation**[115] and

---

[108] [assignment: cryptographic key generation algorithm]
[109] From 2026, symmetric keys must be configured with a minimum of 128 bits.
[110] [assignment: cryptographic key sizes]
[111] [assignment: list of standards]
[112] [assignment: cryptographic key generation algorithm]
[113] [assignment: cryptographic key sizes]
[114] [assignment: list of standards]
[115] [assignment: cryptographic key generation algorithm]

specified cryptographic key sizes **1024[116] to 2048 bits by steps of 32 bits**[117] that meet the following: **see application note**[118].

Application note: the keys are generated and diversified in accordance with [JCAPI310] in classes KeyBuilder (buildKey method) and KeyPair (genKeyPair method).

FCS_CKM.1/ECDSA Cryptographic key generation

FCS_CKM.1.1/ECDSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **ECDSA Key Pair Generation**[119] and specified cryptographic key sizes **P ranging from 256 to 521 bits**[120] that meet the following: **see application note**[121].

Application note:

- The keys are generated and diversified in accordance with [JCAPI310] in classes KeyBuilder (buildKey, buildXECKey methods) and KeyPair (genKeyPair method).
- The TOE implements elliptic curve cryptography over GF(p), supporting the following [JCAPI310] key types:

[JCAPI310] class	Supported parameters
javacard.security.KeyBuilder	TYPE_EC_FP_PRIVATE LENGTH_EC_FP_256 TYPE_EC_FP_PRIVATE LENGTH_EC_FP_384 TYPE_EC_FP_PRIVATE LENGTH_EC_FP_521 TYPE_EC_FP_PRIVATE_TRANSIENT_RESET TYPE_EC_FP_PRIVATE_TRANSIENT_DESELECT
javacard.security.KeyPair	ALG_EC_FP LENGTH_EC_FP_256 ALG_EC_FP LENGTH_EC_FP_384 ALG_EC_FP LENGTH_EC_FP_521
javacard.security.NamedParameterSpec	BRAINPOOLP256R1 BRAINPOOLP256T1 BRAINPOOLP320R1 BRAINPOOLP320T1 BRAINPOOLP384R1 BRAINPOOLP384T1 BRAINPOOLP512R1 BRAINPOOLP512T1 SECP256R1 SECP384R1 SECP521R1

FCS_CKM.1/HMAC Cryptographic key generation

FCS_CKM.1.1/HMAC The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **HMAC Key generation**[122] and specified cryptographic key sizes **see application note**[123] that meet the following: **[JCAPI310] standard**[124].

[116] The minimum length of the module must be configured to 2048 bits, until 2030. From 2031, the minimum size is 3072 bits.
[117] [assignment: cryptographic key sizes]
[118] [assignment: list of standards]
[119] [assignment: cryptographic key generation algorithm]
[120] [assignment: cryptographic key sizes]
[121] [assignment: list of standards]
[122] [assignment: cryptographic key generation algorithm]
[123] [assignment: cryptographic key sizes]
[124] [assignment: list of standards]

Application note

In accordance with [JCAPI310], the keys are generated and diversified in class KeyBuilder (buildKey method). The following [JCAPI310] parameters are supported:

[JCAPI310] class	Supported parameters
javacard.security.KeyBuilder	TYPE_HMAC_TRANSIENT_RESET TYPE_HMAC_TRANSIENT_DESELECT TYPE_HMAC LENGTH_HMAC_SHA_1_BLOCK_64 TYPE_HMAC LENGTH_HMAC_SHA_256_BLOCK_64 TYPE_HMAC LENGTH_HMAC_SHA_384_BLOCK_64 TYPE_HMAC LENGTH_HMAC_SHA_384_BLOCK_128 TYPE_HMAC LENGTH_HMAC_SHA_512_BLOCK_64 TYPE_HMAC LENGTH_HMAC_SHA_512_BLOCK_128

As mentioned in [JCAPI310] the key can be of any length, but it is strongly recommended that the key is not shorter than the byte length of the hash output used in the HMAC implementation. Keys with length greater than the hash block length are first hashed with the hash algorithm used for the HMAC implementation. As required, the implementation also supports an HMAC key length equal to the length of the supported hash algorithm block size.

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **see application note**[125] that meets the following: **[JCAPI310] standard**[126].

Application note: the keys are reset as specified in [JCAPI310] Key class, with the method clearKey(). Any access to a cleared key for ciphering or signing shall throw an exception.

FCS_COP.1/TDES_CIPHER Cryptographic operation

FCS_COP.1.1/TDES_CIPHER The TSF shall perform **encryption and decryption of applet instance's data**[127] in accordance with a specified cryptographic algorithm **Triple DES 2 Keys or Triple DES 3 Keys with cipher modes mentioned in the application note below**[128] and cryptographic key sizes **112 bits**[129] **for TDES 2 Keys, 168 bits for TDES 3 Keys**[130] that meet the following: **FIPS PUB 46-3, FIPS PUB 81, ISO/IEC 9797-1, PKCS#5**[131].

Application note: the following TDES ciphers from [JCAPI310] are implemented:

Mode	Field name in [JCAPI310] Cipher class
CBC	ALG_DES_CBC_NOPAD
CBC	ALG_DES_CBC_ISO9797_M1
CBC	ALG_DES_CBC_ISO9797_M2
CBC	ALG_DES_CBC_PKCS5
ECB	ALG_DES_ECB_NOPAD
ECB	ALG_DES_ECB_ISO9797_M1
ECB	ALG_DES_ECB_ISO9797_M2
ECB	ALG_DES_ECB_PKCS5

[125] [assignment: cryptographic key destruction method]
[126] [assignment: list of standards]
[127] [assignment: list of cryptographic operations]
[128] [assignment: cryptographic algorithm]
[129] From 2026, symmetric keys must be configured with a minimum of 128 bits.
[130] [assignment: cryptographic key sizes]
[131] [assignment: list of standards]

FCS_COP.1/TDES_MAC Cryptographic operation

FCS_COP.1.1/TDES_MAC The TSF shall perform **MAC computation of applet instance's data**[132] in accordance with a specified cryptographic algorithm **MAC algorithms mentioned in the application note below**[133] and cryptographic key sizes **112 bits**[129] **for TDES 2 Keys, 168 bits for TDES 3 Keys**[134] that meet the following: **FIPS PUB 46-3, FIPS PUB 81, ISO/IEC 9797-1, PKCS#5**[135].

Application note: the following TDES MACs from [JCAPI310] are implemented:

MAC length	MAC algorithm	Field name in [JCAPI310] Signature class
4 bytes	ISO9797-1 MAC algorithm 3	ALG_DES_MAC4_ISO9797_1_M1_ALG3
4 bytes	ISO9797-1 MAC algorithm 3	ALG_DES_MAC4_ISO9797_1_M2_ALG3
4 bytes	3DES in outer CBC mode	ALG_DES_MAC4_ISO9797_M1
4 bytes	3DES in outer CBC mode	ALG_DES_MAC4_ISO9797_M2
4 bytes	3DES in outer CBC mode	SIG_CIPHER_DES_MAC4
4 bytes	3DES in outer CBC mode	ALG_DES_MAC4_PKCS5
4 bytes	3DES in outer CBC mode	ALG_DES_MAC4_NOPAD
8 bytes	ISO9797-1 MAC algorithm 3	ALG_DES_MAC8_ISO9797_1_M1_ALG3
8 bytes	ISO9797-1 MAC algorithm 3	ALG_DES_MAC8_ISO9797_1_M2_ALG3
8 bytes	3DES in outer CBC mode	ALG_DES_MAC8_ISO9797_M1
8 bytes	3DES in outer CBC mode	ALG_DES_MAC8_ISO9797_M2
8 bytes	3DES in outer CBC mode	SIG_CIPHER_DES_MAC8
8 bytes	3DES in outer CBC mode	ALG_DES_MAC8_PKCS5
8 bytes	3DES in outer CBC mode	ALG_DES_MAC8_NOPAD

FCS_COP.1/AES_CIPHER Cryptographic operation

FCS_COP.1.1/AES_CIPHER The TSF shall perform **encryption and decryption of applet instance's data**[136] in accordance with a specified cryptographic algorithm **AES with cipher modes mentioned in the application note below**[137] and cryptographic key sizes **128, 192 and 256 bits**[138] that meet the following: **FIPS PUB 197, NIST SP800-38A, NIST SP800-38D, ISO/IEC 9797-1, PKCS#5, RFC3610**[139].

Application note: the following AES ciphers from [JCAPI310] are implemented:

Mode	Field name in [JCAPI310] Cipher class
CBC	ALG_AES_BLOCK_128_CBC_NOPAD
CBC	ALG_AES_CBC_ISO9797_M1
CBC	ALG_AES_CBC_ISO9797_M2
CBC	ALG_AES_CBC_PKCS5
ECB	ALG_AES_BLOCK_128_ECB_NOPAD
ECB	ALG_AES_ECB_ISO9797_M1
ECB	ALG_AES_ECB_ISO9797_M2
ECB	ALG_AES_ECB_PKCS5
CTR	ALG_AES_CTR
Mode	**Field name in [JCAPI310] AEADCipher class**
Counter with CBC-MAC	ALG_AES_CCM
Counter with CBC-MAC	CIPHER_AES_CCM

[132] [assignment: list of cryptographic operations]
[133] [assignment: cryptographic algorithm]
[134] [assignment: cryptographic key sizes]
[135] [assignment: list of standards]
[136] [assignment: list of cryptographic operations]
[137] [assignment: cryptographic algorithm]
[138] [assignment: cryptographic key sizes]
[139] [assignment: list of standards]

Galois/Counter Mode (GCM)	ALG_AES_GCM
Galois/Counter Mode (GCM)	CIPHER_AES_GCM

FCS_COP.1/AES_MAC Cryptographic operation

FCS_COP.1.1/AES_MAC The TSF shall perform **MAC computation of applet instance's data**[140] in accordance with a specified cryptographic algorithm **MAC algorithms mentioned in the application note below**[141] and cryptographic key sizes **128, 192 and 256 bits**[142] that meet the following: **FIPS PUB 197, NIST SP800-38A**[143].

Application note: the following AES MACs from [JCAPI310] are implemented:

MAC length	MAC algorithm	Field name in [JCAPI310] Signature class
16 bytes	AES in CBC mode, block size 128 bits	ALG_AES_MAC_128_NOPAD
16 bytes	AES in CBC mode, block size 128 bits	SIG_CIPHER_AES_MAC128
16 bytes	AES in CBC mode, block size 128 bits	SIG_CIPHER_AES_CMAC128
16 bytes	AES in CBC mode, block size 128 bits	ALG_AES_CMAC_128

FCS_COP.1/RSA_SIGN Cryptographic operation

FCS_COP.1.1/RSA_SIGN The TSF shall perform **signature generation and signature verification of applet instance's data**[144] in accordance with a specified cryptographic algorithm **RSA Standard and RSA CRT with hash algorithms and padding schemes mentioned in the application note below**[145] and cryptographic key sizes **1024 to 2048**[146] **bits by steps of 32 bits**[147] that meet the following: **PKCS#1, PKCS#1-PSS (IEEE 1363-2000), ISO/IEC 9796-2 and RFC2409**[148].

Application note: the following RSA signatures from [JCAPI310] are implemented:

Hash algorithm	Padding scheme	Field name in [JCAPI310] Signature class
SHA224	PKCS#1	ALG_RSA_SHA_224_PKCS1
SHA224	PKCS#1-PSS scheme (IEEE 1363-2000)	ALG_RSA_SHA_224_PKCS1_PSS
SHA256	PKCS#1	ALG_RSA_SHA_256_PKCS1
SHA256	PKCS#1-PSS scheme (IEEE 1363-2000)	ALG_RSA_SHA_256_PKCS1_PSS
SHA384	PKCS#1	ALG_RSA_SHA_384_PKCS1
SHA384	PKCS#1-PSS scheme (IEEE 1363-2000)	ALG_RSA_SHA_384_PKCS1_PSS
SHA512	PKCS#1	ALG_RSA_SHA_512_PKCS1
SHA512	PKCS#1-PSS scheme (IEEE 1363-2000)	ALG_RSA_SHA_512_PKCS1_PSS
SHA1	ISO 9796-2	ALG_RSA_SHA_ISO9796
SHA1	ISO 9796-2	ALG_RSA_SHA_ISO9796_MR
SHA1	PKCS#1	ALG_RSA_SHA_PKCS1
SHA1	PKCS#1-PSS scheme (IEEE 1363-2000)	ALG_RSA_SHA_PKCS1_PSS
SHA1	RFC2409	ALG_RSA_SHA_RFC2409
-	-	SIG_CIPHER_RSA

[140] [assignment: list of cryptographic operations]
[141] [assignment: cryptographic algorithm]
[142] [assignment: cryptographic key sizes]
[143] [assignment: list of standards]
[144] [assignment: list of cryptographic operations]
[145] [assignment: cryptographic algorithm]
[146] The minimum length of the module must be configured to 2048 bits, until 2030. From 2031, the minimum size is 3072 bits.
[147] [assignment: cryptographic key sizes]
[148] [assignment: list of standards]

FCS_COP.1/RSA_CIPHER Cryptographic operation

FCS_COP.1.1/RSA_CIPHER The TSF shall perform **encryption and decryption of applet instance's data**[149] in accordance with a specified cryptographic algorithm **RSA Standard and RSA CRT as mentioned in the application note below**[150] and cryptographic key sizes **1024 to 2048**[151] **bits by steps of 32 bits**[152] that meet the following: **PKCS#1, PKCS#1-OAEP scheme (IEEE 1363-2000)**[153].

Application note: the following RSA ciphers from [JCAPI310] are implemented:

[JCAPI310] class	Implemented algorithms
Cipher	ALG_RSA_NOPAD
	ALG_RSA_PKCS1
	ALG_RSA_PKCS1_OAEP
	PAD_PKCS1_OAEP
	PAD_PKCS1_OAEP_SHA224
	PAD_PKCS1_OAEP_SHA256
Cipher	PAD_PKCS1_OAEP_SHA384
	PAD_PKCS1_OAEP_SHA512

FCS_COP.1/ECDSA_SIGN Cryptographic operation

FCS_COP.1.1/ECDSA_SIGN The TSF shall perform **signature generation and signature verification of applet instance's data**[154] in accordance with a specified cryptographic algorithm **ECDSA as mentioned in the application note below**[155] and cryptographic key sizes **P ranging from 256 to 521 bits**[156] that meet the following: **FIPS PUB 186-4**[157].

Application note: the following ECDSA signatures from [JCAPI310] are implemented:

Hash algorithm	Field name in [JCAPI310] Signature class
SHA1	ALG_ECDSA_SHA
SHA224	ALG_ECDSA_SHA_224
SHA256	ALG_ECDSA_SHA_256
SHA384	ALG_ECDSA_SHA_384
SHA512	ALG_ECDSA_SHA_512
-	SIG_CIPHER_ECDSA
-	SIG_CIPHER_ECDSA_PLAIN

FCS_COP.1/ECDH Cryptographic operation

FCS_COP.1.1/ECDH The TSF shall perform **Secret Key Agreement**[158] in accordance with a specified cryptographic algorithm **Elliptic Curve Diffie-Hellman (ECDH)**[159] and cryptographic key sizes **P ranging from 256 to 521 bits**[160] that meet the following: **IEEE P1363**[161].

---

[149] [assignment: list of cryptographic operations]
[150] [assignment: cryptographic algorithm]
[151] The minimum length of the module must be configured to 2048 bits, until 2030. From 2031, the minimum size is 3072 bits.
[152] [assignment: cryptographic key sizes]
[153] [assignment: list of standards]
[154] [assignment: list of cryptographic operations]
[155] [assignment: cryptographic algorithm]
[156] [assignment: cryptographic key sizes]
[157] [assignment: list of standards]
[158] [assignment: list of cryptographic operations]

Application note: the secret keys are derived using the KeyAgreement class (generateSecret method) of javacard.security. The following [JCAPI310] fields are supported:

Field name in [JCAPI310] KeyAgreement class
ALG_EC_SVDP_DH_KDF
ALG_EC_SVDP_DH_PLAIN
ALG_EC_SVDP_DH_PLAIN_XY
ALG_EC_SVDP_DHC_KDF
ALG_EC_SVDP_DHC_PLAIN

Application note : The parameters for ECDH key agreement operations are: BRAINPOOLP256R1, BRAINPOOLP256T1, BRAINPOOLP320R1, BRAINPOOLP320T1, BRAINPOOLP384R1, BRAINPOOLP384T1, BRAINPOOLP512R1, BRAINPOOLP512T1, SECP256R1, SECP384R1, SECP521R1.

FCS_COP.1/DH Cryptographic operation

FCS_COP.1.1/DH The TSF shall perform **Key Exchange**[162] in accordance with a specified cryptographic algorithm **Diffie-Hellman (DH)**[163] and cryptographic key sizes **RSA key sizes from 1024 to 2048**[164] **bits by steps of 32 bits**[165] that meet the following: **NIST SP 800-56Ar2**[166].

FCS_COP.1/Hash Cryptographic operation

FCS_COP.1.1/Hash The TSF shall perform **computation of a hash value for applet instance's data**[167] in accordance with a specified cryptographic algorithm **see application note**[168] and cryptographic key sizes **None**[169] that meet the following: **see application note**[170].

Application note: the following hash algorithms from [JCAPI310] are implemented:

Hash algorithm	Field name in [JCAPI310] MessageDigest class	Related Standard
SHA1[171]	ALG_SHA	FIPS 180-4
SHA-224	ALG_SHA_224	FIPS 180-4
SHA-256	ALG_SHA_256	FIPS 180-4
SHA-384	ALG_SHA_384	FIPS 180-4
SHA-512	ALG_SHA_512	FIPS 180-4

FCS_COP.1/HMAC Cryptographic operation

FCS_COP.1.1/HMAC The TSF shall perform **computation of a HMAC value for applet instance's data**[172] in accordance with a specified cryptographic algorithm **HMAC with hash algorithms**

---

159 [assignment: cryptographic algorithm]
160 [assignment: cryptographic key sizes]
161 [assignment: list of standards]
162 [assignment: list of cryptographic operations]
163 [assignment: cryptographic algorithm]
164 The minimum length of the module must be configured to 2048 bits, until 2030. From 2031, the minimum size is 3072 bits.
165 [assignment: cryptographic key sizes]
166 [assignment: list of standards]
167 [assignment: list of cryptographic operations]
168 [assignment: cryptographic algorithm]
169 [assignment: cryptographic key sizes]
170 [assignment: list of standards]
171 SHA-1 is vulnerable to collision attacks. Its usage must be limited to contexts with no risk of collision attacks.

mentioned in the application note below[173] and cryptographic key sizes **see application note**[174] that meet the following: **rfc2104**[175].

Application note: the following HMAC algorithms from [JCAPI310] are implemented:

Hash algorithm used in HMAC computation	Field name in [JCAPI310] Signature class
SHA1	ALG_HMAC_SHA1
SHA256	ALG_HMAC_SHA_256
SHA384	ALG_HMAC_SHA_384
SHA512	ALG_HMAC_SHA_512
-	SIG_CIPHER_HMAC

As mentioned in [JCAPI310] the key can be of any length, but it is strongly recommended that the key is not shorter than the byte length of the hash output used in the HMAC implementation. Keys with length greater than the hash block length are first hashed with the hash algorithm used for the HMAC implementation. As required, the implementation also supports an HMAC key length equal to the length of the supported hash algorithm block size.

FCS_COP.1/CRC Cryptographic operation

FCS_COP.1.1/CRC The TSF shall perform **Computation of checksum of applet instance's data**[176] in accordance with a specified cryptographic algorithm **CRC16 or CRC32**[177] and cryptographic key sizes **none**[178] that meet the following: **ISO/IEC 3309**[179].

Application note: the related algorithms in [JCAPI310] are ALG_ISO3309_CRC16 and ALG_ISO3309_CRC32 (class Checksum of javacard.security).

FCS_RNG.1 Random Number Generation

FCS_RNG.1.1 The TSF shall provide a **hybrid deterministic**[180] random number generator **DRG.4**[181] [AIS20] [AIS31] that implements: **enhanced backward secrecy & enhanced forward secrecy**[182].

FCS_RNG.1.2 The TSF shall provide random numbers that meet **[AIS31] test procedure A**[183].

FDP_RIP.1/ABORT Subset residual information protection

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

Application note: the events that provoke the de-allocation of a transient object are described in [JCRE310], §5.1.

---

172 [assignment: list of cryptographic operations]
173 [assignment: cryptographic algorithm]
174 [assignment: cryptographic key sizes]
175 [assignment: list of standards]
176 [assignment: list of cryptographic operations]
177 [assignment: cryptographic algorithm]
178 [assignment: cryptographic key sizes]
179 [assignment: list of standards]
180 [selection: physical, non-physical true, deterministic, hybrid, hybrid deterministic]
181 [selection: DRG.2, DRG.3, DRG.4, PTG.2, PTG.3, NTG.1]
182 [assignment: list of security capabilities]
183 [assignment: a defined quality metric]

FDP_RIP.1/APDU	Subset residual information protection

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

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

FDP_RIP.1/GlobalArray	Subset residual information protection

FDP_RIP.1.1/GlobalArray (refined) The TSF shall ensure that any previous information content of a resource is made unavailable upon **deallocation of the resource from** *the applet as a result of returning from the process method to* the following objects: **a user Global Array**.

Application note: An array resource is allocated when a call to the API method JCSystem.makeGlobalArray is performed. The Global Array is created as a transient JCRE Entry Point Object ensuring that reference to it cannot be retained by any application. On return from the method which called JCSystem.makeGlobalArray, the array is no longer available to any applet and is deleted and the memory in use by the array is cleared and reclaimed in the next object deletion cycle.

FDP_RIP.1/bArray	Subset residual information protection

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

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

FDP_RIP.1/KEYS	Subset residual information protection

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

Application note: the javacard.security & javacardx.crypto packages do provide secure interfaces to the cryptographic buffer in a transparent way. See javacard.security.KeyBuilder and Key interface of [JCAPI310].

FDP_RIP.1/TRANSIENT	Subset residual information protection

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

Application note:

- The events that provoke the de-allocation of any transient object are described in [JCRE310], §5.1.
- The clearing of CLEAR_ON_DESELECT objects is not necessarily performed when the owner of the objects is deselected. In the presence of multiselectable applet instances, CLEAR_ON_DESELECT memory segments may be attached to applets that are active in different logical channels. Multiselectable applet instances within a same CAP file must share the transient memory segment if they are concurrently active ([JCRE310], §4.3).

FDP_ROL.1/FIREWALL Basic rollback

FDP_ROL.1.1/FIREWALL The TSF shall enforce **the FIREWALL access control SFP and the JCVM information flow control SFP** to permit the rollback of the **operations OP.JAVA and OP.CREATE** on the **object O.JAVAOBJECT**.

FDP_ROL.1.2/FIREWALL The TSF shall permit operations to be rolled back within the **scope of a select(), deselect(), process(), install() or uninstall() call, notwithstanding the restrictions given in [JCRE310], §7.7, within the bounds of the Commit Capacity ([JCRE310], §7.8), and those described in [JCAPI310]**.

Application note: transactions are a service offered by the APIs to applets. It is also used by some APIs to guarantee the atomicity of some operation. This mechanism is either implemented in Java Card platform or relies on the transaction mechanism offered by the underlying platform. Some operations of the API are not conditionally updated, as documented in [JCAPI310] (see for instance, PIN-blocking, PIN-checking, update of Transient objects).

Card Security Management

FAU_ARP.1 Security alarms

FAU_ARP.1.1 The TSF shall take **one of the following actions: throw an exception, lock the card session, reinitialize the Java Card System and its data,** upon detection of a potential security violation.

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

- CAP file inconsistency,
- typing error in the operands of a bytecode,
- applet life cycle inconsistency,
- card tearing (unexpected removal of the Card out of the CAD) and power failure,
- abort of a transaction in an unexpected context, (see abortTransaction(), [JCAPI310] and ([JCRE310], §7.6.2)
- violation of the Firewall or JCVM SFPs,
- unavailability of resources,
- array overflow,
- **GlobalPlatform card state inconsistency**[184]

Application note: in FAU_ARP.1.1, the [assignment: list of other actions] is set to 'none', meaning that no other actions are defined in this SFR component.

FDP_SDI.2/DATA Stored data integrity monitoring and action

FDP_SDI.2.1/DATA The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors**[185] on all objects, based on the following attributes: **integrity check data**[186].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall **mute the card and decrease the global fault detection counter. Once the global fault detection counter reaches 0, the card is put in degraded mode.**[187]

Application note: the following data persistently stored by TOE have an integrity check data security attribute:

- Key (i.e. objects instance of classes implemented the interface Key)

[184] [assignment: list of other runtime errors]
[185] [assignment: integrity errors]
[186] [assignment: user data attributes]
[187] [assignment: action to be taken]

- PIN (objects instance of class OwnerPin)
- Package
- GlobalPlatform card state (OP_READY, SECURED, CARD_LOCKED, TERMINATE)

FPR_UNO.1 Unobservability

FPR_UNO.1.1 The TSF shall ensure that **any user**[188] are unable to observe the operation **read, write, cryptographic operations**[189] on **PIN, Key**[190] by **any other user or subject**[191].

FPT_FLS.1/JCS Failure with preservation of secure state

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

Application note: the Java Card RE Context is the Current context when the Java Card VM begins running after a card reset ([JCRE310], §6.2.3) or after a proximity card (PICC) activation sequence ([JCRE310]). Behavior of the TOE on power loss and reset is described in [JCRE310], §3.6 and §7.1. Behavior of the TOE on RF signal loss is described in [JCRE310], §3.6.1.

FPT_TDC.1 Inter-TSF basic TSF data consistency

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

FPT_TDC.1.2 The TSF shall use
- **the rules defined in [JCVM310] specification,**
- **the API tokens defined in the export files of reference implementation,**
- **none**[192]

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.

AID management

FIA_ATD.1/AID User attribute definition

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

Refinement: "Individual users" stand for applets.

[188] [assignment: list of users and/or subjects]
[189] [assignment: list of operations]
[190] [assignment: list of objects]
[191] [assignment: list of protected users and/or subjects]
[192] [assignment: list of interpretation rules to be applied by the TSF]

FIA_UID.2/AID User identification before any action

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

Application note:

- By users here it must be understood the ones associated to the CAP files (or applets) that act as subjects of policies. In the Java Card System, every action is always performed by an identified user interpreted here as the currently selected applet or the CAP file that is the subject's owner. Means of identification are provided during the loading procedure of the CAP file and the registration of applet instances.
- The role Java Card RE defined in FMT_SMR.1 is attached to an IT security function rather than to a "user" of the CC terminology. The Java Card RE does not "identify" itself to the TOE, but it is part of it.

FIA_USB.1/AID User-subject binding

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: **CAP file AID are defined with associated value during loading and with context identifier**[193].

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: **None**[194].

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

FMT_MTD.1/JCRE Management of TSF data

FMT_MTD.1.1/JCRE The TSF shall restrict the ability to **modify** the **list of registered applets' AIDs** to **the JCRE**.

Application note:

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

FMT_MTD.3/JCRE Secure TSF data

FMT_MTD.3.1/JCRE The TSF shall ensure that only secure values are accepted for **the registered applets' AIDs**.

ADELG Security Functional Requirements

This group consists of the SFRs related to the deletion of applets and/or CAP files, enforcing the applet deletion manager (ADEL) policy on security aspects outside the runtime. Deletion is a critical operation and therefore requires specific treatment.

[193] [assignment: rules for the initial association of attributes]
[194] [assignment: rules for the changing of attributes]

Application note: patch deletion is an extension of applet/package deletion defined in GlobalPlatform as a patch is managed as a JavaCard Package and registered with specific attributes handled with GemActivate.

FDP_ACC.2/ADEL Complete access control

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

Refinement: the operations involved in the policy are: OP.DELETE_APPLET, OP.DELETE_CAP_FILE, and OP.DELETE_CAP_FILE_APPLET.

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

FDP_ACF.1/ADEL Security attribute based access control

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

Subject / Object	Attributes
S.JCVM	Active Applets
S.JCRE	Selected Applet Context, Registered Applets, Resident CAP files
O.CODE_CAP_FILE	CAP file AID, AIDs of packages within a CAP file, Dependent package AID, Static References
O.APPLET	Applet Selection Status
O.JAVAOBJECT	Owner, Remote

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

- **In the context of this policy, an object O is reachable if and only one of the following conditions hold:**
  1) **the owner of O is a registered applet instance A (O is reachable from A),**
  2) **a static field of a resident package P contains a reference to O (O is reachable from P),**
  3) **there exists a valid remote reference to O (O is remote reachable),**
  4) **there exists an object O' that is reachable according to either (1) or (2) or (3) above and O' contains a reference to O (the reachability status of O is that of O').**
- **The following access control rules determine when an operation among controlled subjects and objects is allowed by the policy:**
  - **R.JAVA.14 ([JCRE310], §11.3.4.2, Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon an O.APPLET only if,**
    1) **S.ADEL is currently selected,**
    2) **there is no instance in the context of O.APPLET that is active in any logical channel and**
    3) **there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance distinct from O.APPLET, or O.JAVAOBJECT is reachable from a package P, or ([JCRE3], §8.5) O.JAVAOBJECT is remote reachable.**
  - **R.JAVA.15 ([JCRE310], §11.3.4.2.1, Multiple Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon several O.APPLET only if,**
    1) **S.ADEL is currently selected,**
    2) **there is no instance of any of the O.APPLET being deleted that is active in any logical channel and**
    3) **there is no O.JAVAOBJECT owned by any of the O.APPLET being deleted such that either O.JAVAOBJECT is reachable from an applet instance distinct from any of those O.APPLET, or O.JAVAOBJECT is reachable from a CAP file P, or ([JCRE310], §8.5) O.JAVAOBJECT is remote reachable.**
  - **R.JAVA.16 ([JCRE310], §11.3.4.3, Applet/Library CAP file Deletion): S.ADEL may perform OP.DELETE_CAP_FILE upon an O.CODE_CAP_FILE only if,**
    1) **S.ADEL is currently selected,**

2) **no reachable O.JAVAOBJECT, from a CAP file distinct from O.CODE_CAP_FILE that is an instance of a class that belongs to O.CODE_CAP_FILE, exists on the card and**
3) **there is no resident package on the card that depends on O.CODE_CAP_FILE.**

- **R.JAVA.17 ([JCRE310], §11.3.4.4, Applet CAP file and Contained Instances Deletion): S.ADEL may perform OP.DELETE_CAP_FILE_APPLET upon an O.CODE_CAP_FILE only if,**
  1) **S.ADEL is currently selected,**
  2) **no reachable O.JAVAOBJECT, from a CAP file distinct from O.CODE_CAP_FILE, which is an instance of a class that belongs to O.CODE_CAP_FILE exists on the card,**
  3) **there is no CAP file loaded on the card that depends on O.CODE_CAP_FILE, and**
  4) **for every O.APPLET of those being deleted it holds that: (i) there is no instance in the context of O.APPLET that is active in any logical channel and (ii) there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance not being deleted, or O.JAVAOBJECT is reachable from a CAP file not being deleted, or ([JCRE310], §8.5) O.JAVAOBJECT is remote reachable.**

FDP_ACF.1.3/ADEL The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **none**.

FDP_ACF.1.4/ADEL The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **any subject but S.ADEL to O.CODE_PKG or O.APPLET for the purpose of deleting them from the card**.

Application note, FDP_ACF.1.2/ADEL:

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

FDP_RIP.1/ADEL Subset residual information protection

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

Application note: deleted freed resources (both code and data) may be reused, depending on the way they were deleted (logically or physically). Requirements on de-allocation during applet/CAP file deletion are described in [JCRE310], §11.3.4.1, §11.3.4.2 and §11.3.4.3.

FMT_MSA.1/ADEL Management of security attributes

FMT_MSA.1.1/ADEL The TSF shall enforce the **ADEL access control SFP** to restrict the ability to **modify** the security attributes **Registered Applets and Resident CAP files** to **the Java Card RE**.

Application note: patch deletion is an extension of applet/package deletion defined in GlobalPlatform as a patch is managed as a JavaCard Package and registered with specific attributes.

FMT_MSA.3/ADEL Static attribute initialization

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

FMT_MSA.3.2/ADEL The TSF shall allow the **following role(s): none,** to specify alternative initial values to override the default values when an object or information is created.

Application note: patch deletion is an extension of applet/package deletion defined in GlobalPlatform as a patch is managed as a JavaCard Package and registered with specific attributes.

FMT_SMF.1/ADEL Specification of Management Functions

FMT_SMF.1.1/ADEL The TSF shall be capable of performing the following management functions: **modify the list of registered applets' AIDs and the Resident CAP files**.

FMT_SMR.1/ADEL Security roles

FMT_SMR.1.1/ADEL The TSF shall maintain the roles**: applet deletion manager**.

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

FPT_FLS.1/ADEL Failure with preservation of secure state

FPT_FLS.1.1/ADEL The TSF shall preserve a secure state when the following types of failures occur: **the applet deletion manager fails to delete a CAP file/applet as described in [JCRE310], §11.3.4**.
Application note:

- The TOE may provide additional feedback information to the card manager in case of a potential security violation (see FAU_ARP.1).
- The CAP file/applet instance deletion must be atomic. The "secure state" referred to in the requirement must comply with Java Card specification ([JCRE310], §11.3.4.)

Application note: patch deletion is an extension of applet/package deletion defined in GP as a patch is managed as a JavaCard Package and registered with specific attributes.

ODELG Security Functional Requirements

The following requirements concern the object deletion mechanism. This mechanism is triggered by the applet that owns the deleted objects by invoking a specific API method.

FDP_RIP.1/ODEL Subset residual information protection

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

Application note:

- Freed data resources resulting from the invocation of the method javacard.framework.JCSystem.requestObjectDeletion() may be reused. Requirements on de-allocation after the invocation of the method are described in [JCAPI310].
- There is no conflict with FDP_ROL.1 here because of the bounds on the rollback mechanism: the execution of requestObjectDeletion() is not in the scope of the rollback because it must be performed in between APDU command processing, and therefore no transaction can be in progress.

FPT_FLS.1/ODEL Failure with preservation of secure state

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

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

SCP Security Functional Requirements

This section states the security functional requirements for the Smart Card Platform.

Operating System

This section presents those requirements of the Smart Card Platform group that concern the Operating System. Due to the enlargement of the evaluation scope, the requirements related to OS are now assigned to the TOE and no more to the environment. Other internal security mechanisms are not addressed by SFR but ADV_ARC activities.

FPT_RCV.3/OS Automated recovery without undue loss

FPT_RCV.3.1/OS When automated recovery from **none, see application note below**[195] 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/OS For **execution access to a memory zone reserved for TSF data, writing access to a memory zone reserved for TSF's code, and any segmentation fault performed by a Java Card applet**[196] the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3/OS The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding

- **the contents of Java Card static fields, instance fields, and array positions that fall under the scope of an open transaction;**
- **the Java Card objects that were allocated into the scope of an open transaction;**
- **the contents of Java Card transient objects;**
- **any possible Executable Load File being loaded when the failure occurred**[197]

for loss of TSF data or objects under the control of the TSF.

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

Application note: there is no maintenance mode implemented within the TOE. Recovery is always enforced automatically as stated in FPT_RCV.3.2/OS.

FPT_RCV.4/OS Function recovery

FPT_RCV.4.1/OS The TSF shall ensure that **reading from and writing to static and objects' fields interrupted by power loss**[198] have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

Security Functional Requirements from 'Sensitive Array' package

Package SensitiveArrays defines mechanism for creating and handling integrity-sensitive array objects.

[195] [assignment: list of failures/service discontinuities during card content management operations]
[196] [assignment: list of failures/service discontinuities during card content management operations]
[197] [assignment: quantification]
[198] [assignment: list of functions and failure scenarios]

FDP_SDI.2/ARRAY Integrity_Sensitive_Array

FDP_SDI.2.1/ARRAY The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors** on all objects, based on the following attributes: **user data stored in arrays created by the makeIntegritySensitiveArray() method of the javacard.framework.SensitiveArrays class**.

FDP_SDI.2.2/ARRAY Upon detection of a data integrity error, the TSF shall **throw an exception**.

Security Functional Requirements from 'Sensitive Result' package

Package SensitiveResult defines mechanism for asserting results of sensitive functions.

FDP_SDI.2/RESULT Integrity_Sensitive_Result

FDP_SDI.2.1/RESULT The TSF shall monitor user data stored in containers controlled by the TSF for **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**.

Refinement: in addition of throwing an exception, the TSF will mute the card further if redundancy checking of data integrity detects an error.

Security Functional Requirements from 'Monotonic Counters' package

Package MonotonicCounter defines mechanism for creating a counter that can only be increased.

FDP_SDI.2/MONOTONIC_COUNTER

FDP_SDI.2.1/MONOTONIC_COUNTER The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors** on all objects, based on the following attributes: **stored user data i.e. the counter value in the MonotonicCounter object.**

FDP_SDI.2.2/MONOTONIC_COUNTER Upon detection of a data integrity error, the TSF shall **throw an exception**.

Application note: This requirement applies to MonotonicCounter objects created by the getInstance() method of the javacardx.security.util.MonotonicCounter class.

Security Functional Requirements from 'Cryptographic Certificate Management' package

Package Cryptographic Certificate Management defines mechanism for secure management of public key certificates.

FDP_SDI.2/CRT_MNGT

FDP_SDI.2.1/CRT_MNGT The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors** on all objects, based on the following attributes: **cryptographic certificate.**

FDP_SDI.2.2/CRT_MNGT Upon detection of a data integrity error, the TSF shall **throw an exception**.

FCS_COP.1/CRT_MNGT Certificate Management

FCS_COP.1.1/CRT_MNGT The TSF shall perform **verification of X.509 Certificate**[199] in accordance with a specified cryptographic algorithm **as mentioned in the application note below**[200] and cryptographic key sizes **as mentioned in the application note below**[201] that meet the following: **standards mentioned in the application note below**[202].

Application note: X.509 certificates are verified according to the following table listing the algorithms, key sizes and related standards:

Cryptographic algorithm	Cryptographic key sizes	Standard
ALG_TYPE_EC_FP_PUBLIC	LENGTH_EC_FP_256 LENGTH_EC_FP_384 LENGTH_EC_FP_521	IEEE P1363
ALG_TYPE_RSA_PUBLIC	LENGTH_RSA_1024 LENGTH_RSA_1280 LENGTH_RSA_1536 LENGTH_RSA_1984 LENGTH_RSA_2048	NIST SP800-56Ar2

Security Functional Requirements from 'Key Derivation Functions (KDF)' package

Package Key Derivation defines classes implementing cryptographic derivation functions.

FCS_CKM.5/KDF Key Derivation Function

FCS_CKM.5.1/KDF The TSF shall derive cryptographic keys **Keys generated according to the Key Derivation Functions mentioned in the application note below**[203] from **Key Derivation Buffer**[204] in accordance with a specified cryptographic key derivation algorithm **as mentioned in the application note below**[205] and specified cryptographic key sizes **as mentioned in the application note below**[206] that meet the following: **standards listed in the application note below**[207].

Application note: the following table lists the available key derivation options for FCS_CKM.5/KDF:

Key Derivation Function	Cryptographic algorithms as defined in [JCAPI310]	Key sizes as defined in [JCAPI310]	Standards
ALG_KDF_COUNTER_MODE	Signature.ALG_HMAC_SHA1 Signature.ALG_HMAC_SHA_256 Signature.ALG_AES_CMAC_128	LENGTH_SHA LENGTH_SHA_256 LENGTH_AES_128	NIST SP800-108
ALG_PRF_TLS12	Signature.ALG_HMAC_SHA1 Signature.ALG_HMAC_SHA_256	LENGTH_SHA LENGTH_SHA_256	IETF RFC 5246
ALG_KDF_ICAO_MRTD	MessageDigest.ALG_SHA MessageDigest.ALG_SHA_256	LENGTH_SHA LENGTH_SHA_256	ICAO MRTD Doc 9303
ALG_KDF_ANSI_X9_63	MessageDigest.ALG_SHA_224 MessageDigest.ALG_SHA_256 MessageDigest.ALG_SHA_384 MessageDigest.ALG_SHA_512	LENGTH_SHA_224 LENGTH_SHA_256 LENGTH_SHA_384 LENGTH_SHA_512	ANSI X9.63
ALG_KDF_HKDF	Signature.ALG_HMAC_SHA1 Signature.ALG_HMAC_SHA_256	LENGTH_SHA LENGTH_SHA_256	IETF RFC 5869

199 [assignment: verification of X.509 Certificate]
200 [assignment: cryptographic algorithm]
201 [assignment: cryptographic key sizes]
202 [assignment: list of standards]
203 [assignment: key type]
204 [assignment: input parameters]
205 [assignment: cryptographic key derivation algorithm]
206 [assignment: cryptographic key sizes]
207 [assignment: list of standards]

Security Functional Requirements from 'System Time' package

Package System Time defines mechanism for handling system time, suitable for timestamps or for estimating intervals between events.

FPT_STM.1/SYS_TIME

FPT_STM.1.1/SYS_TIME The TSF shall be able to provide reliable time stamps.

9.2 SECURITY ASSURANCE REQUIREMENTS

This ST is based on the EAL4 assurance package augmented with the components AVA_VAN.5 and ALC_DVS.2.

9.3 SECURITY REQUIREMENTS RATIONALE

9.3.1 TOE security objectives coverage – Mapping table from [PP-GP]

Security Objective	SFRs
O.CARD-MANAGEMENT	FPT_FLS.1/GP, FDP_ROL.1/GP, FCO_NRO.2/GP, FMT_SMR.1/GP, FMT_SMF.1/GP, FDP_ITC.2/GP-ELF, FDP_ITC.2/GP-KL, FPT_RCV.3/GP, FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FIA_UID.1/GP, FIA_AFL.1/GP, FIA_UAU.1/GP, FIA_UAU.4/GP, FDP_UIT.1/GP, FDP_UCT.1/GP, FTP_ITC.1/GP, FPR_UNO.1/GP, FPT_TDC.1/GP, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL, FMT_MSA.1/GP, FMT_MSA.3/GP, FDP_ACC.1/GP-GS, FDP_ACF.1/GP-GS, FMT_MSA.1/GP-GS, FMT_MSA.3/GP-GS, FMT_SMF.1/GP-GS, FMT_SMR.1/GP-GS, FCS_COP.1/GP-DAP_SHA, FCS_COP.1/GP-DAP_VER, FCO_NRO.2/GP-DAP, FTP_TRP.1/GP-TF
O.DOMAIN-RIGHTS	FMT_SMR.1/GP, FMT_SMF.1/GP, FCO_NRO.2/GP, FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FIA_UID.1/GP, FIA_AFL.1/GP, FIA_UAU.1/GP, FIA_UAU.4/GP, FTP_ITC.1/GP, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL, FMT_MSA.1/GP, FMT_MSA.3/GP
O.APPLI-AUTH	FMT_SMR.1/GP, FDP_ITC.2/GP-ELF, FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FTP_ITC.1/GP, FMT_MSA.1/GP, FMT_MSA.3/GP, FCS_COP.1/GP-DAP_SHA, FCS_COP.1/GP-DAP_VER, FCO_NRO.2/GP-DAP
O.SECURITY-DOMAINS	FMT_SMR.1/GP, FMT_SMF.1/GP, FMT_MSA.1/GP, FMT_MSA.3/GP
O.COMM-AUTH	FMT_SMR.1/GP, FMT_SMF.1/GP, FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FIA_UID.1/GP, FIA_UAU.1/GP, FIA_UAU.4/GP, FTP_ITC.1/GP, FCS_COP.1/GP-SCP, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL, FMT_MSA.1/GP, FMT_MSA.3/GP
O.COMM-INTEGRITY	FMT_SMR.1/GP, FMT_SMF.1/GP, FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FTP_ITC.1/GP, FCS_COP.1/GP-SCP, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL, FMT_MSA.1/GP, FMT_MSA.3/GP
O.COMM-CONFIDENTIALITY	FMT_SMR.1/GP, FMT_SMF.1/GP, FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FTP_ITC.1/GP, FCS_COP.1/GP-SCP, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL, FMT_MSA.1/GP,

	FMT_MSA.3/GP
O.NO-KEY-REUSE	FIA_AFL.1/GP, FIA_UAU.4/GP
O.PRIVILEGES-MANAGEMENT	FMT_SMR.1/GP, FMT_SMF.1/GP, FMT_MTD.1/GP-PR, FMT_MTD.3/GP
O.LC-MANAGEMENT	FMT_MTD.1/GP-LC, FMT_MTD.3/GP, FMT_SMF.1/GP, FMT_SMR.1/GP, FMT_MSA.1/GP, FMT_MSA.3/GP
O.CLFDB-DECIPHER	FCS_COP.1/GP-CLFDB
O.GLOBAL-CVM	FPR_UNO.1/GP-CVM
O.CVM-BLOCK	FIA_AFL.1.1/GP-CVM
O.CVM-MGMT	FIA_AFL.1.1/GP-CVM, FPR_UNO.1/GP-CVM
O.RECEIPT	FCO_NRR.1/GP-RECEIPT, FCS_COP.1/GP-RECEIPT
O.TOKEN	FCO_NRO.2/GP-TOKEN, FCS_COP.1/GP-TOKEN
O.ELF_AUTHORISED	FMT_MSA.1/GP-ELFU, FMT_MSA.3/GP-ELFU, FMT_SMF.1/GP-ELFU, FDP_ACC.1/GP-ELFU, FDP_ACF.1/GP-ELFU
O.ELF_INTEGRITY	FIA_UID.1/GP, FDP_ACC.1/GP-ELFU, FDP_ACF.1/GP-ELFU
O.ELF_APP_DATA	FPT_FLS.1/GP-ELFU
O.ELF_SESSION	FMT_SMF.1/GP-ELFU, FIA_UID.1/GP
O.ELF_DELE_IRR	FDP_ROL.1/GP-ELFU
O.ELF_DATA_PRO	FDP_RIP.1/ADEL
O.SECURE_LOAD_ACODE	FDP_ACC.1/OS-UPDATE, FDP_ACF.1/OS-UPDATE, FMT_MSA.3/OS-UPDATE, FMT_SMR.1/OS-UPDATE, FMT_SMF.1/OS-UPDATE, FCS_COP.1/OS-UPDATE-VER
O.SECURE_AC_ACTIVATION	FDP_ACC.1/OS-UPDATE, FDP_ACF.1/OS-UPDATE, FMT_MSA.3/OS-UPDATE, FMT_SMR.1/OS-UPDATE, FMT_SMF.1/OS-UPDATE, FPT_FLS.1/OS-UPDATE
O.TOE_IDENTIFICATION	FDP_ACC.1/OS-UPDATE, FDP_ACF.1/OS-UPDATE, FIA_ATD.1/OS-UPDATE, FMT_MSA.3/OS-UPDATE, FMT_SMR.1/OS-UPDATE, FMT_SMF.1/OS-UPDATE
O.CONFID-OS-UPDATE.LOAD	FDP_ACC.1/OS-UPDATE, FDP_ACF.1/OS-UPDATE, FMT_MSA.3/OS-UPDATE, FMT_SMR.1/OS-UPDATE, FMT_SMF.1/OS-UPDATE, FTP_TRP.1/OS-UPDATE, FCS_COP.1/OS-UPDATE-DEC
O.SID	FDP_ITC.2/GP-ELF, FDP_ITC.2/GP-KL, FMT_SMR.1/GP, FMT_SMF.1/GP, FMT_MSA.1/GP, FMT_MSA.3/GP, FIA_ATD.1/AID, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_SMF.1/ADEL, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FIA_UID.2/AID, FIA_USB.1/AID
O.FIREWALL	FMT_SMR.1/GP, FMT_SMF.1/GP, FDP_ITC.2/GP-ELF, FDP_ITC.2/GP-KL, FMT_MSA.1/GP, FMT_MSA.3/GP, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FDP_IFF.1/JCVM, FDP_IFC.1/JCVM, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1, FMT_SMF.1, FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_MSA.2/FIREWALL_JCVM, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM
O.GLOBAL_ARRAYS_CONFID	FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_IFF.1/JCVM, FDP_IFC.1/JCVM
O.GLOBAL_ARRAYS_INTEG	FDP_IFF.1/JCVM, FDP_IFC.1/JCVM
O.ARRAY_VIEWS_CONFID	FDP_IFF.1/JCVM, FDP_IFC.1/JCVM
O.ARRAY_VIEWS_INTEG	FDP_IFF.1/JCVM, FDP_IFC.1/JCVM
O.NATIVE	FDP_ACF.1/FIREWALL
O.OPERATE	FPT_FLS.1/GP, FPT_RCV.3/GP, FPT_TDC.1, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FAU_ARP.1, FDP_ROL.1/FIREWALL, FIA_ATD.1/AID, FIA_USB.1/AID, FPT_STM.1.1/SYS_TIME

O.REALLOCATION	FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ADEL
O.RESOURCES	FPT_RCV.3/GP, FMT_SMR.1/GP, FMT_SMF.1/GP, FPT_FLS.1/GP, FAU_ARP.1, FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FDP_ROL.1/FIREWALL, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1, FMT_SMF.1, FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FPT_STM.1.1/SYS_TIME
O.ALARM	FPT_FLS.1/GP, FPT_FLS.1/JCS, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL, FAU_ARP.1
O.CIPHER	FCS_CKM.1/GP-SCP, FCS_COP.1/GP-SCP, FCS_COP.1/GP-DAP_SHA, FCS_COP.1/GP-DAP_VER, FCO_NRO.2/GP-DAP, FCS_CKM.1/TDES, FCS_CKM.1/AES, FCS_CKM.1/RSA, FCS_CKM.1/ECDSA, FCS_CKM.1/HMAC, FCS_CKM.4, FCS_COP.1/TDES_CIPHER, FCS_COP.1/TDES_MAC, FCS_COP.1/AES_CIPHER, FCS_COP.1/AES_MAC, FCS_COP.1/RSA_SIGN, FCS_COP.1/RSA_CIPHER, FCS_COP.1/ECDSA_SIGN, FCS_COP.1/ECDH, FCS_COP.1/Hash, FCS_COP.1/HMAC, FCS_COP.1/DH, FCS_COP.1/CRC, FPR_UNO.1, FCS_CKM.5/KDF
O.RNG	FCS_RNG.1
O.KEY-MNGT	FPT_TDC.1/GP, FCS_CKM.1/GP-SCP, FCS_COP.1/GP-SCP, FCS_CKM.1/TDES, FCS_CKM.1/AES, FCS_CKM.1/RSA, FCS_CKM.1/ECDSA, FCS_CKM.1/HMAC, FCS_CKM.4, FCS_COP.1/TDES_CIPHER, FCS_COP.1/TDES_MAC, FCS_COP.1/AES_CIPHER, FCS_COP.1/AES_MAC, FCS_COP.1/RSA_SIGN, FCS_COP.1/RSA_CIPHER, FCS_COP.1/ECDSA_SIGN, FCS_COP.1/ECDH, FCS_COP.1/Hash, FCS_COP.1/HMAC, FCS_COP.1/DH, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FCS_CKM.5/KDF
O.PIN-MNGT	FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FPR_UNO.1, FDP_ROL.1/FIREWALL, FDP_SDI.2/DATA, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL
O.TRANSACTION	FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT, FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_RIP.1/OBJECTS
O.OBJ-DELETION	FDP_RIP.1/ODEL, FPT_FLS.1/ODEL
O.DELETION	FPT_RCV.3/GP, FDP_ACC.2/ADEL, FDP_ACF.1/ADEL, FDP_RIP.1/ADEL, FPT_FLS.1/ADEL, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL
O.LOAD	FCO_NRO.2/GP, FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FDP_UIT.1/GP, FIA_UID.1/GP, FTP_ITC.1/GP, FIA_UAU.1/GP, FIA_UAU.4/GP, FCS_COP.1/GP-DAP_SHA, FCS_COP.1/GP-DAP_VER, FCO_NRO.2/GP-DAP
O.INSTALL	FDP_ITC.2/GP-ELF, FPT_FLS.1/GP, FPT_RCV.3/GP, FCS_COP.1/GP-DAP_SHA, FCS_COP.1/GP-DAP_VER, FCO_NRO.2/GP-DAP
O.SCP.IC	FPT_FLS.1/JCS
O.SCP.RECOVERY	FPT_RCV.3/OS
O.SCP.SUPPORT	FPT_RCV.4/OS
O.SENSITIVE_ARRAYS_INTEG	FDP_SDI.2/ARRAY
O.SENSITIVE_RESULTS_INTEG	FDP_SDI.2/RESULT
O.MTC-CTR-MNGT	FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU,

	FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_ROL.1/FIREWALL, FDP_SDI.2/MONOTONIC_COUNTER
O.CRT-MNGT	FDP_SDI.2/CRT_MNGT, FCS_COP.1/CRT_MNGT

Table 17: TOE Security Objectives coverage by SFRs – Mapping table [PP-GP]

9.3.2 TOE security objectives coverage – Rationale from [PP-GP]

O.CARD-MANAGEMENT is fulfilled by the following SFRs:

- FDP_UIT.1/GP ensures the integrity of card management operations.
- FDP_UCT.1/GP ensures the confidentiality of card management operations.
- FDP_ROL.1/GP ensures the rollback of the installation or removal operation on the executable files and application instances.
- FDP_ITC.2/GP-ELF enforces the ELF loading information flow policy when importing ELFs.
- FDP_ITC.2/GP-KL enforces the Data & Key information flow policy when importing keys and data.
- FPT_FLS.1/GP requires the card to preserve a secure state when failures occur during loading/installing/deleting an Executable File / application instance.
- FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL enforce the information flow control policy for managing, authenticating, and protecting the Card management commands and responses between off-card and on-card entities.
- FIA_UID.1/GP, FIA_UAU.1/GP and FIA_UAU.4/GP ensure appropriate identification and authentication mechanisms. In addition, these SFRs specify the actions being performed before the authentication of the origin of the received APDU commands takes place.
- FCO_NRO.2/GP enforces the evidence of the origin during the loading of Executable Load Files, SD/Application data and keys.
- FPR_UNO.1/GP enforces the invisibility of the imported keys and the encryption, decryption, signature generation and verification cryptographic mechanisms on SD/Application keys and data.
- FPT_TDC.1/GP specifies requirements preventing any possible misinterpretation of the Security Domain keys used to implement a Secure Channel when those are loaded from the off-card entity.
- FTP_ITC.1/GP requires a trusted channel for authenticating the card management commands and for securely protecting (authenticity, integrity, and/or confidentiality) the loading of ELF/data.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
  - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
  - Enforce the TOE Life cycle management and transitions.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.
- FPT_RCV.3/GP ensures safe recovery from failure.
- FIA_AFL.1/GP supports the objective by bounding the number of signatures that the attacker may try to attach to a message to authenticate its origin.
- FDP_ACC.1/GP-GS, FDP_ACF.1/GP-GS enforce the GlobalPlatform Services access control policy for managing the registration, deregistration, and access of the Global Service.
- FMT_MSA.1/GP-GS and FMT_MSA.3/GP-GS specify security attributes that support management of the Global Service privilege, the service name and AID.
- FMT_SMR.1/GP-GS maintains the roles S.OPEN, Global Services Application and their associated Life Cycle states.

- FMT_SMF.1/GP-GS enforces the management of Global Services Applications (Registering, Deregistering, Accessing).
- FCS_COP.1/GP-DAP_SHA, FCS_COP.1/GP-DAP_VER, FCO_NRO.2/GP-DAP ensure that ELFs received by the TOE have been generated by an authorized actor (integrity and authenticity evidence).
- FTP_TRP.1/GP-TF ensures that a trusted path is enforced for application personalization through the GlobalPlatform Trusted Framework.

O.DOMAIN-RIGHTS is fulfilled by the following SFRs:

- FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL enforce the ELF, data and keys loading information flow control policy for managing, authenticating and protecting the Card management commands and responses between off-card and on-card entities.
- FIA_UID.1/GP, FIA_UAU.1/GP and FIA_UAU.4/GP ensure appropriate identification and authentication mechanisms. In addition, these SFRs specify the actions being performed before the authentication of the origin of the received APDU commands takes place.
- FTP_ITC.1/GP requires a trusted channel for authenticating the card management commands and for securely protecting (authenticity, integrity, and/or confidentiality) the loading of ELF/data.
- FCO_NRO.2/GP enforces the evidence of the origin during the loading of Executable Load Files, SD/Application data and keys.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
    - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
    - Enforce the TOE Life cycle management and transitions.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.

O.APPLI-AUTH is fulfilled by the following SFRs:

- FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF enforce the ELF loading information flow control policy for managing, authenticating, and protecting the Card management commands.
- FDP_ITC.2/GP-ELF enforces the ELF loading information flow policy when importing ELFs.
- FTP_ITC.1/GP requires a trusted channel for authenticating the card management commands and for securely protecting (authenticity, integrity, and/or confidentiality) the loading of ELF/data.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
    - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
    - Enforce the TOE Life cycle management and transitions.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.
- FCS_COP.1/GP-DAP_SHA, FCS_COP.1/GP-DAP_VER, FCO_NRO.2/GP-DAP ensure that ELFs received by the TOE have been generated by an authorized actor (integrity and authenticity evidence).

O.SECURITY-DOMAINS is fulfilled by the following SFRs:

- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles

and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
  - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
  - Enforce the TOE Life cycle management and transitions.

O.COMM-AUTH is fulfilled by the following SFRs:
- FTP_ITC.1/GP requires a trusted channel for authenticating the card management commands and for securely protecting (authenticity, integrity, and/or confidentiality) the loading of ELF/data.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity and confidentiality.
- FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL enforce the ELF, data and keys loading information flow control policy for managing, authenticating, and protecting the Card management commands and responses between off-card and on-card entities.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
  - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
  - Enforce the TOE Life cycle management and transitions.
- FIA_UID.1/GP, FIA_UAU.1/GP and FIA_UAU.4/GP ensure appropriate identification and authentication mechanisms. In addition, these SFRs specify the actions being performed before the authentication of the origin of the received APDU commands takes place.
- FCS_COP.1/GP-SCP specifies the cryptographic operations and algorithms that shall be applied for the authorization of the card management commands.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.

O.COMM-INTEGRITY is fulfilled by the following SFRs:
- FTP_ITC.1/GP requires a trusted channel for authenticating the card management commands and for securely protecting (authenticity, integrity, and/or confidentiality) the loading of ELF/data.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.
- FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL enforce the ELF, data and keys loading information flow control policy for managing, authenticating, and protecting the Card management commands and responses between off-card and on-card entities.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
  - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
  - Enforce the TOE Life cycle management and transitions.
- FCS_COP.1/GP-SCP specifies the cryptographic operations and algorithms that shall be used to ensure the integrity of the card management commands.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.

O.COMM-CONFIDENTIALITY is fulfilled by the following SFRs:

- FTP_ITC.1/GP requires a trusted channel for authenticating the card management commands and for securely protecting (authenticity, integrity, and/or confidentiality) the loading of ELF/data.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.
- FDP_IFC.2/GP-ELF, FDP_IFF.1/GP-ELF, FDP_IFC.2/GP-KL, FDP_IFF.1/GP-KL enforce the ELF, data and keys loading information flow control policy for managing, authenticating, and protecting the Card management commands and responses between off-card and on-card entities.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
    - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
    - Enforce the TOE Life cycle management and transitions.
- FCS_COP.1/GP-SCP specifies the cryptographic operations and algorithms that shall be used to ensure the confidentiality of the card management commands (decryption of the card management commands).

O.NO-KEY-REUSE is fulfilled by the following SFRs:

- FIA_UAU.4/GP enforces the objective by requesting the TSF to prevent the reuse of authentication data related to the implementation of Secure Channels.
- FIA_AFL.1/GP supports the objective by bounding the number of signatures that the attacker may try to attach to a message to authenticate its origin.

O.PRIVILEGES-MANAGEMENT is fulfilled by the following SFRs:

- FMT_MTD.1/GP-PR, FMT_MTD.3/GP cover Privileges Assignment and Management functions.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity and confidentiality.

O.LC-MANAGEMENT is fulfilled by the following SFRs:

- FMT_MTD.1/GP-LC, FMT_MTD.3/GP cover Life Cycle Management functions and transitions.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
    - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
    - Enforce the TOE Life cycle management and transitions.

O.CLFDB-DECIPHER is fulfilled by FCS_COP.1/GP-CLFDB which specifies the cryptographic operations and algorithms that shall be used to decrypt the Ciphered Load File Data Block when it is received by the SE.

O.GLOBAL-CVM is fulfilled by FPR_UNO.1/GP-CVM which ensures that unauthorized users are unable to observe the comparison on Global PIN.

O.CVM-BLOCK is fulfilled by FIA_AFL.1.1/GP-CVM which detects the authentication failure attempts related to user authentication using CVM.

O.CVM-MGMT is fulfilled by the following SFRs:
- FPR_UNO.1/GP-CVM ensures that unauthorized users are unable to observe the comparison on Global PIN.
- FIA_AFL.1.1/GP-CVM detects the authentication failure attempts related to user authentication using CVM.

O.RECEIPT is fulfilled by the following SFRs:
- FCO_NRR.1/GP-RECEIPT generates evidence of receipt for received card management operation requests.
- FCS_COP.1/GP-RECEIPT ensures that the card management command has been successfully processed by computing the Receipt signature.

O.TOKEN is fulfilled by the following SFRs:
- FCO_NRO.2/GP-TOKEN generates an evidence of origin for 'ELF with Token Verification' received from the off-card entity.
- FCS_COP.1/GP-TOKEN ensures that the card management command is authorized by verifying the Token signature.

O.ELF_AUTHORISED is fulfilled by the following SFRs:
- Only the entity authenticated at the SD to which an ELF belongs can upgrade the ELF. That entity must have access rights to the security domain according to the ELF upgrade access control policy (FDP_ACC.1/GP-ELFU, FDP_ACF.1/GP-ELFU).
- FMT_MSA.3/GP-ELFU enforces the access control policy by providing restrictive default values for security attributes defined in FDP_ACF.1.1/GP-ELFU.
- FMT_MSA.1/GP-ELFU enforces the access control policy by restricting the ability to set and maintain the security attributes defined in FDP_ACF.1.1/GP-ELFU to the S.OPEN.
- FMT_SMF.1/GP-ELFU contributes to this objective by specifying the management functions available to load an authorized ELF

O.ELF_INTEGRITY is related to the integrity of the upgraded ELF being loaded onto the platform, which is protected by the Secure Channel protocol (FIA_UID.1/GP) and the ELF upgrade access control policy (FDP_ACC.1/GP-ELFU, FDP_ACF.1/GP-ELFU).

O.ELF_APP_DATA is fulfilled by FPT_FLS.1/GP-ELFU which contributes to this objective by preventing the use of corrupted application data.

O.ELF_SESSION is fulfilled by the following SFRs:
- FMT_SMF.1/GP-ELFU contributes to this Objective by defining the start & end of the ELF_UPGRADE session.
- FIA_UID.1/GP specifies the actions that can be performed before the origin of the APDU commands that the card receives has been authorized.

O.ELF_DELE_IRR is fulfilled by FDP_ROL.1/GP-ELFU which contributes to this objective by preserving the completion of the deletion operation.

O.ELF_DATA_PRO is fulfilled by FDP_RIP.1/ADEL which ensures that contents of resources are only available to subjects having explicitly granted access to these resources.

O.SECURE_LOAD_ACODE is fulfilled by the following SFRs:
- FDP_ACC.1/OS-UPDATE and FDP_ACF.1/OS-UPDATE enforce the OS Update Access Control Policy on the loading, installation, and activation of additional code.
- FMT_MSA.3/OS-UPDATE specifies security attributes that support management of the loading, installation, and activation of additional code.

- FMT_SMR.1/OS-UPDATE maintains the role of OS Developer, which is responsible for signature verification and decryption of additional code before Loading, Installation, and Activation.
- FMT_SMF.1/OS-UPDATE manages the activation of additional code.
- FCS_COP.1/OS-UPDATE-VER specifies the cryptographic algorithms used to perform digital signature verification of the additional code to be loaded.

O.SECURE_AC_ACTIVATION is fulfilled by the following SFRs:
- FDP_ACC.1/OS-UPDATE and FDP_ACF.1/OS-UPDATE enforce the OS Update Access Control Policy on the loading, installation, and activation of additional code.
- FMT_MSA.3/OS-UPDATE specifies security attributes that support management of the loading, installation, and activation of additional code.
- FMT_SMR.1/OS-UPDATE maintains the role of OS Developer, which is responsible for signature verification and decryption of additional code before Loading, Installation, and Activation.
- FMT_SMF.1/OS-UPDATE manages the activation of additional code.
- FPT_FLS.1/OS-UPDATE ensures that the TOE remains in a secure state in case of interruption or incident which prevents the forming of the Updated TOE.

O.TOE_IDENTIFICATION is fulfilled by the following SFRs:
- FDP_ACC.1/OS-UPDATE and FDP_ACF.1/OS-UPDATE enforce the OS Update Access Control Policy on the loading, installation, and activation of additional code.
- FIA_ATD.1/OS-UPDATE maintains the additional code ID for each activated additional code.
- FMT_MSA.3/OS-UPDATE specifies security attributes that support management of the loading, installation, and activation of additional code.
- FMT_SMR.1/OS-UPDATE maintains the role of OS Developer, which is responsible for signature verification and decryption of additional code before Loading, Installation, and Activation.
- FMT_SMF.1/OS-UPDATE manages the activation of additional code.

O.CONFID-OS-UPDATE.LOAD is fulfilled by the following SFRs:
- FDP_ACC.1/OS-UPDATE and FDP_ACF.1/OS-UPDATE enforce the OS Update Access Control Policy on the loading, installation, and activation of additional code.
- FMT_MSA.3/OS-UPDATE specifies security attributes that support management of the loading, installation, and activation of additional code.
- FMT_SMR.1/OS-UPDATE maintains the role of OS Developer, which is responsible for signature verification and decryption of additional code before Loading, Installation, and Activation.
- FMT_SMF.1/OS-UPDATE manages the activation of additional code.
- FTP_TRP.1/OS-UPDATE provides a trusted path during the transmission of the additional code to the TOE for loading.
- FCS_COP.1/OS-UPDATE-DEC specifies the cryptographic algorithms used to decrypt the additional code prior to installation.

O.SID is fulfilled by the following SFRs:
- FDP_ITC.2/GP-ELF enforces the ELF loading information flow policy when importing ELFs.
- FDP_ITC.2/GP-KL enforces the Data & Key information flow policy when importing keys and data.
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
    - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
    - Enforce the TOE Life cycle management and transitions.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.

- As stated in [PP-JCS], subjects' identity is AID-based (applets, packages and CAP files), and is met by FIA_ATD.1/AID, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_SMF.1/ADEL, FMT_MTD.1/JCRE and FMT_MTD.3/JCRE.
- As stated in [PP-JCS], installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSF) or re-use of identities (FIA_UID.2/AID, FIA_USB.1/AID).

O.FIREWALL is fulfilled by the following SFRs:
- FMT_MSA.1/GP and FMT_MSA.3/GP specify security attributes enabling to:
  - Ensure the authenticity, integrity, and/or confidentiality of card management commands;
  - Enforce the TOE Life cycle management and transitions.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the belonging commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider, Controlling Authority roles and specifies the authorized roles enabled for sending and authenticating card management commands. These commands have to be protected with regard to integrity, authenticity and confidentiality.
- FDP_ITC.2/GP-ELF enforces the ELF loading information flow policy when importing ELFs.
- FDP_ITC.2/GP-KL enforces the Data & Key information flow policy when importing keys and data.
- As stated in [PP-JCS], this objective is also met by the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL, the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM). The functional requirements of the class FMT (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1, FMT_SMF.1, FMT_SMR.1/ADEL, FMT_SMF.1/ADEL, FMT_MSA.2/FIREWALL_JCVM, FMT_MSA.3/FIREWALL, FMT_MSA.3/JCVM, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.1/JCRE, FMT_MSA.1/JCVM) also indirectly contribute to meet this objective.

O.GLOBAL_ARRAYS_CONFID coverage: only arrays can be designated as global, and the only global arrays required in the Java Card API are the APDU buffer, the global byte array input parameter (bArray) to an applet's install method and the global arrays created by the JCSystem.makeGlobalArray(…) method. The clearing requirement of these arrays is met by FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray and FDP_RIP.1/bArray respectively. The JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.

O.GLOBAL_ARRAYS_INTEG is met by the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM), which prevents an application from keeping a pointer to the APDU buffer of the card, to the global byte array of the applet's install method or to the global arrays created by the JCSystem.makeGlobalArray(…) method. Such a pointer could be used to access and modify it when the buffer is being used by another application.

O.ARRAY_VIEWS_CONFID coverage: array views have security attributes of temporary objects where the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from storing a reference to the array view. Furthermore, array views may not have ATTR_READABLE_VIEW security attribute which ensures that no application can read the contents of the array view.

O.ARRAY_VIEWS_INTEG coverage: array views have security attributes of temporary objects where the JCVM information flow control policy (FDP_IFF.1/JCVM, FDP_IFC.1/JCVM) prevents an application from storing a reference to the array view. Furthermore, array views may not have ATTR_WRITABLE_VIEW security attribute which ensures that no application can alter the contents of the array view.

O.NATIVE is covered by FDP_ACF.1/FIREWALL: the only means to execute native code is the invocation of a Java Card API method. This objective mainly relies on the environmental objective OE.CAP_FILE, which uphold the assumption A.CAP_FILE.

O.OPERATE is fulfilled by the following SFRs:

- FPT_FLS.1/GP requires the card to preserve a secure state when failures occur during loading/installing/deleting an Executable File / application instance
- FPT_RCV.3/GP ensures safe recovery from failure
- As stated in [PP-JCS], the TOE is protected in various ways against applets' actions (FPT_TDC.1, the FIREWALL access control policy FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL), and is able to detect and block various failures or security violations during usual working (FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL, FAU_ARP.1). Its security-critical parts and procedures are also protected: applets' installation may be cleanly aborted (FDP_ROL.1/FIREWALL), communication with external users and their internal subjects is well-controlled (FIA_ATD.1/AID, FIA_USB.1/AID) to prevent alteration of TSF data (also protected by components of the FPT class).
- FPT_STM.1.1/SYS_TIME requires the TSF to provide reliable time stamps as optional System Time package is implemented.

O.REALLOCATION is satisfied by the following SFRs: FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ADEL, which imposes that the contents of the re-allocated block shall always be cleared before delivering the block.

O.RESOURCES is fulfilled by the following SFRs:

- FPT_RCV.3/GP ensures safe recovery from failure.
- FMT_SMF.1/GP enforces the card management operations (Loading, Installation, etc.), the privileges, the life cycle states and transition by defining the protective actions for the corresponding commands.
- FMT_SMR.1/GP maintains the roles S.OPEN, ISD, SSD, Application, and their associated Life Cycle states. In addition, it maintains the Application Provider and the Controlling Authority roles and specifies the authorized roles that are allowed to send and authenticate the card management commands. These commands have to be protected with regard to integrity, authenticity, and confidentiality.
- FPT_FLS.1/GP requires the card to preserve a secure state when failures occur during loading/installing/deleting of an Executable File / application instance.
- As stated in [PP-JCS], the TSF detects stack/memory overflows during execution of applications (FAU_ARP.1, FPT_FLS.1/ADEL, FPT_FLS.1/JCS, FPT_FLS.1/ODEL). Failed installations are not to create memory leaks (FDP_ROL.1/FIREWALL) as well. Memory management is controlled by the TSF (FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_SMR.1, FMT_SMF.1, FMT_SMR.1/ADEL and FMT_SMF.1/ADEL).
- FPT_STM.1.1/SYS_TIME requires the TSF to provide reliable time stamps as optional System Time package is implemented.

O.ALARM is fulfilled by the following SFRs:

- FPT_FLS.1/GP requires the card to preserve a secure state when failures occur during loading/installing/deleting an Executable File / application instance.
- As stated in [PP-JCS], O.ALARM is also met by FPT_FLS.1/JCS, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL which guarantee that a secure state is preserved by the TSF when failures occur, and FAU_ARP.1 which defines TSF reaction upon detection of a potential security violation.

O.CIPHER is fulfilled by the following SFRs:

- FCS_CKM.1/GP-SCP specifies the algorithm, key sizes, and standards used for the generation of session keys.
- FCS_COP.1/GP-SCP specifies the cryptographic operations and algorithms that shall be used to establish a Secure Channel to protect the card management commands.
- FCS_COP.1/GP-DAP_SHA and FCS_COP.1/GP-DAP_VER ensure that the loaded Executable Application is legitimate by specifying the algorithm to be used in order to verify the DAP signature of the Verification Authority.

- FCO_NRO.2/GP-DAP generates an evidence of origin for 'ELF with DAP' received from the off-card entity.
- As stated in [PP-JCS], O.CIPHER is also covered by FCS_CKM.1/TDES, FCS_CKM.1/AES, FCS_CKM.1/RSA, FCS_CKM.1/ECDSA, FCS_CKM.1/HMAC, FCS_CKM.4, FCS_COP.1/TDES_CIPHER, FCS_COP.1/TDES_MAC, FCS_COP.1/AES_CIPHER, FCS_COP.1/AES_MAC, FCS_COP.1/RSA_SIGN, FCS_COP.1/RSA_CIPHER, FCS_COP.1/ECDSA_SIGN, FCS_COP.1/ECDH, FCS_COP.1/Hash, FCS_COP.1/HMAC, FCS_COP.1/CRC and FCS_COP.1/DH. The SFR FPR_UNO.1 contributes in covering this security objective and controls the observation of the cryptographic operations which may be used to disclose the keys.
- FCS_CKM.5/KDF for Key Derivation Function. The TSF behind these are implemented by API classes.

O.RNG is directly covered by FCS_RNG.1 which ensures the cryptographic quality of random number generation.

O.KEY-MNGT is fulfilled by the following SFRs:
- FPT_TDC.1/GP specifies requirements preventing any possible misinterpretation of the Security Domain keys used to implement a Secure Channel when those are loaded form the off-card entity.
- FCS_CKM.1/GP-SCP specifies the algorithm, key sizes, and standards used for the generation of session keys.
- FCS_COP.1/GP-SCP specifies the cryptographic operations and algorithms that shall be used to establish a Secure Channel to protect the card management commands.
- As stated in [PP-JCS], this objective is also covered by FCS_CKM.1/TDES, FCS_CKM.1/AES, FCS_CKM.1/RSA, FCS_CKM.1/ECDSA, FCS_CKM.1/HMAC, FCS_CKM.4, FCS_COP.1/TDES_CIPHER, FCS_COP.1/TDES_MAC, FCS_COP.1/AES_CIPHER, FCS_COP.1/AES_MAC, FCS_COP.1/RSA_SIGN, FCS_COP.1/RSA_CIPHER, FCS_COP.1/ECDSA_SIGN, FCS_COP.1/ECDH, FCS_COP.1/Hash, FCS_COP.1/HMAC, FCS_COP.1/DH, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL and FDP_RIP.1/TRANSIENT.
- FCS_CKM.5/KDF for Key Derivation Function. The TSF behind these are implemented by API classes.

O.PIN-MNGT is ensured by FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FPR_UNO.1, FDP_ROL.1/FIREWALL and FDP_SDI.2/DATA security functional requirements. The TSFs behind these are implemented by API classes. The firewall security functions FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL shall protect the access to private and internal data of the objects.

O.TRANSACTION is directly met by FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT, FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT and FDP_RIP.1/OBJECTS.

O.OBJ-DELETION specifies that deletion of objects is secure. The security objective is met by the security functional requirements FDP_RIP.1/ODEL and FPT_FLS.1/ODEL.

O.DELETION is fulfilled by the following SFRs:
- FPT_RCV.3/GP ensures safe recovery from failure
- As stated in [PP-JCS], this security objective specifies that applet and CAP file deletion must be secure. The non-introduction of security holes is ensured by the ADEL access control policy (FDP_ACC.2/ADEL, FDP_ACF.1/ADEL). The integrity and confidentiality of data that does not belong to the deleted applet or CAP file is a by-product of this policy as well. Non-accessibility of deleted data is met by FDP_RIP.1/ADEL and the TSFs are protected against possible failures of the deletion procedures (FPT_FLS.1/ADEL). The security functional requirements of the class FMT (FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL) included in the group ADELG also contribute to meet this objective.

O.LOAD is fulfilled by the following SFRs:

- FCO_NRO.2/GP enforces the evidence of the origin during the loading of Executable Load Files, SD/Application data and keys.
- FDP_IFC.2/GP-ELF and FDP_IFF.1/GP-ELF enforce the ELF loading information flow control policy for managing, authenticating, and protecting the card management commands.
- FDP_UIT.1/GP ensures the integrity of the card management operations.
- FIA_UID.1/GP, FIA_UAU.1/GP and FIA_UAU.4/GP ensure appropriate identification and authentication mechanisms. In addition, these SFRs specify the actions being performed before the authentication of the origin of the received APDU commands takes place.
- FTP_ITC.1/GP requires a trusted channel for authenticating the card management commands and for securely protecting (authenticity, integrity, and/or confidentiality) the loading of ELF/data.
- FCS_COP.1/GP-DAP_SHA and FCS_COP.1/GP-DAP_VER ensure that the loaded Executable Application is legitimate by specifying the algorithm to be used in order to verify the DAP signature of the Verification Authority.
- FCO_NRO.2/GP-DAP generates an evidence of origin for ‘ELF with DAP’ received from the off-card entity.

O.INSTALL is fulfilled by the following SFRs:

- FDP_ITC.2/GP-ELF enforces the ELF loading information flow policy when importing ELFs.
- FPT_FLS.1/GP requires the card to preserve a secure state when failures occur during loading/installing/deleting an Executable File / application instance.
- FPT_RCV.3/GP ensures safe recovery from failure.
- FCS_COP.1/GP-DAP_SHA and FCS_COP.1/GP-DAP_VER ensure that the loaded Executable Application is legitimate by specifying the algorithm to be used in order to verify the DAP signature of the Verification Authority.
- FCO_NRO.2/GP-DAP generates an evidence of origin for ‘ELF with DAP’ received from the off-card entity.

O.SCP.IC coverage: the IC is a part of the TOE supporting TSFs of the upper layer of the TOE and more specially FPT_FLS.1/JCS.

O.SCP.RECOVERY coverage: the SCP is a part of the TOE supporting TSFs of the upper layer of the TOE, especially for recovery operations as dealt with in FPT_RCV.3/OS.

O.SCP.SUPPORT coverage: the SCP is a part of the TOE supporting TSFs of the upper layer of the TOE, especially for recovery operations as dealt with in FPT_RCV.4/OS.

O.SENSITIVE_ARRAYS_INTEG is covered directly by FDP_SDI.2/ARRAY which ensures that integrity errors related to the user data stored in sensitive arrays are detected by the TOE

O.SENSITIVE_RESULTS_INTEG is covered directly by FDP_SDI.2/RESULT which ensures that integrity errors related to the sensitive API result are detected by the TOE.

O.MTC-CTR-MNGT is ensured by FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/GlobalArray FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_ROL.1/FIREWALL and FDP_SDI.2/MONOTONIC_COUNTER security functional requirements. The TSFs behind these are implemented by API classes. The firewall security functions FDP_ACC.2/FIREWALL and FDP_ACF.1/FIREWALL shall protect the access to private and internal data of the objects.

O.CRT-MNGT is ensured by FDP_SDI.2/CRT_MNGT and FCS_COP.1/CRT_MNGT security functional requirements. The applets that manage cryptographic certificates rely on the security functions that implement these SFRs.

9.3.3 *SFR dependency rationale*

Security Functional Requirement	CC dependencies	Satisfied dependencies
FDP_IFC.2/GP-ELF	(FDP_IFF.1)	FDP_IFF.1/GP-ELF
FDP_IFF.1/GP-ELF	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.2/GP-ELF FMT_MSA.3/GP
FDP_ITC.2/GP-ELF	(FDP_ACC.1 or FDP_IFC.1) and (FPT_TDC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.2/GP-ELF FPT_TDC.1/GP FTP_ITC.1/GP
FDP_IFC.2/GP-KL	(FDP_IFF.1)	FDP_IFF.1/GP-KL
FDP_IFF.1/GP-KL	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.2/GP-KL FMT_MSA.3/GP
FDP_ITC.2/GP-KL	(FDP_ACC.1 or FDP_IFC.1) and (FPT_TDC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.2/GP-KL FPT_TDC.1/GP FTP_ITC.1/GP
FMT_MTD.1/GP-LC	(FMT_SMF.1) and (FMT_SMR.1)	FMT_SMR.1/GP FMT_SMF.1/GP
FMT_MTD.1/GP-PR	(FMT_SMF.1) and (FMT_SMR.1)	FMT_SMR.1/GP FMT_SMF.1/GP
FCS_CKM.1/GP-SCP	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1/GP-SCP FCS_CKM.4
FCS_COP.1/GP-SCP	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/GP-SCP FCS_CKM.4
FTP_TRP.1/GP-TF	No dependencies	
FMT_MSA.1/GP	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_IFC.2/GP-ELF FDP_IFC.2/GP-KL FMT_SMR.1/GP FMT_SMF.1/GP
FMT_MSA.3/GP	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/GP FMT_SMR.1/GP
FMT_SMR.1/GP	(FIA_UID.1)	FIA_UID.1/GP
FMT_SMF.1/GP	No dependencies	
FPT_RCV.3/GP	(AGD_OPE.1)	AGD_OPE.1
FPT_FLS.1/GP	No dependencies	
FPT_TDC.1/GP	No dependencies	
FTP_ITC.1/GP	No dependencies	
FCO_NRO.2/GP	(FIA_UID.1)	FIA_UID.1/GP
FIA_UID.1/GP	No dependencies	
FDP_UIT.1/GP	(FDP_ACC.1 or FDP_IFC.1) and (FTP_ITC.1 or FTP_TRP.1)	FDP_IFC.2/GP-ELF FDP_IFC.2/GP-KL FTP_ITC.1/GP
FDP_ROL.1/GP	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.2/GP-ELF FDP_IFC.2/GP-KL
FDP_UCT.1/GP	(FTP_ITC.1 or FTP_TRP.1) and (FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.2/GP-ELF FDP_IFC.2/GP-KL FTP_ITC.1/GP
FPR_UNO.1/GP	No dependencies	
FIA_UAU.1/GP	(FIA_UID.1)	FIA_UID.1/GP
FIA_UAU.4/GP	No dependencies	
FIA_AFL.1/GP	(FIA_UAU.1)	FIA_UAU.1/GP
FMT_MTD.3/GP	(FMT_MTD.1)	FMT_MTD.1/GP-PR FMT_MTD.1/GP-LC
FCS_COP.1/GP-CLFDB	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FDP_ITC.2/GP-ELF FCS_CKM.4
FDP_ACC.1/GP-GS	(FDP_ACF.1)	FDP_ACF.1/GP-GS
FDP_ACF.1/GP-GS	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.1/GP-GS FMT_MSA.3/GP-GS
FMT_MSA.1/GP-GS	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.1/GP-GS FMT_SMF.1/GP-GS FMT_SMR.1/GP-GS
FMT_MSA.3/GP-GS	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/GP-GS FMT_SMR.1/GP-GS
FMT_SMR.1/GP-GS	(FIA_UID.1)	FIA_UID.1/GP

Security Functional Requirement	CC dependencies	Satisfied dependencies
FMT_SMF.1/GP-GS	No dependencies	
FIA_AFL.1/GP-CVM	(FIA_UAU.1)	FIA_UAU.1/GP
FPR_UNO.1/GP-CVM	No dependencies	
FCO_NRR.1/GP-RECEIPT	(FIA_UID.1)	FIA_UID.1/GP
FCO_NRO.2/GP-TOKEN	(FIA_UID.1)	FIA_UID.1/GP
FCS_COP.1/GP-TOKEN	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FDP_ITC.2/GP-ELF FDP_ITC.2/GP-KL FCS_CKM.4
FCS_COP.1/GP-RECEIPT	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FDP_ITC.2/GP-ELF FDP_ITC.2/GP-KL FCS_CKM.4
FCS_COP.1/GP-DAP_SHA	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FDP_ITC.2/GP-ELF FCS_CKM.4
FCS_COP.1/GP-DAP_VER	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FDP_ITC.2/GP-ELF FCS_CKM.4
FCO_NRO.2/GP-DAP	(FIA_UID.1)	FIA_UID.1/GP
FDP_ACC.1/GP-ELFU	(FDP_ACF.1)	FDP_ACF.1/GP-ELFU
FDP_ACF.1/GP-ELFU	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.1/GP-ELFU FMT_MSA.3/GP-ELFU
FDP_ROL.1/GP-ELFU	(FDP_ACC.1 or FDP_IFC.1)	FDP_ACC.1/GP-ELFU
FMT_MSA.1/GP-ELFU	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.1/GP-ELFU FMT_SMR.1/GP FMT_SMF.1/GP-ELFU
FMT_MSA.3/GP-ELFU	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/GP-ELFU FMT_SMR.1/GP
FMT_SMF.1/GP-ELFU	No dependencies	
FPT_FLS.1/GP-ELFU	No dependencies	
FDP_ACC.1/OS-UPDATE	(FDP_ACF.1)	FDP_ACF.1/OS-UPDATE
FDP_ACF.1/OS-UPDATE	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.1/OS-UPDATE FMT_MSA.3/OS-UPDATE
FMT_MSA.3/OS-UPDATE	(FMT_MSA.1) and (FMT_SMR.1)	FMT_SMR.1/OS-UPDATE **See rationale**
FMT_SMR.1/OS-UPDATE	(FIA_UID.1)	FIA_UID.1/GP
FMT_SMF.1/OS-UPDATE	No dependencies	
FIA_ATD.1/OS-UPDATE	No dependencies	
FTP_TRP.1/OS-UPDATE	No dependencies	
FCS_COP.1/OS-UPDATE-DEC	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FDP_ITC.2/GP-ELF FCS_CKM.4
FCS_COP.1/OS-UPDATE-VER	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FDP_ITC.2/GP-ELF FCS_CKM.4
FPT_FLS.1/OS-UPDATE	No dependencies	
FDP_ACC.2/FIREWALL	(FDP_ACF.1)	FDP_ACF.1/FIREWALL
FDP_ACF.1/FIREWALL	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2/FIREWALL FMT_MSA.3/FIREWALL
FDP_IFC.1/JCVM	(FDP_IFF.1)	FDP_IFF.1/JCVM
FDP_IFF.1/JCVM	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1/JCVM FMT_MSA.3/JCVM
FDP_RIP.1/OBJECTS	No dependencies	
FMT_MSA.1/JCRE	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL FMT_SMR.1 **See rationale**
FMT_MSA.1/JCVM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL FDP_IFC.1/JCVM FMT_SMF.1 FMT_SMR.1
FMT_MSA.2/FIREWALL_JCVM	(FDP_ACC.1 or FDP_IFC.1) and (FMT_MSA.1) and (FMT_SMR.1)	FDP_ACC.2/FIREWALL FDP_IFC.1/JCVM FMT_MSA.1/JCRE FMT_MSA.1/JCVM FMT_SMR.1
FMT_MSA.3/FIREWALL	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCRE FMT_MSA.1/JCVM FMT_SMR.1

Security Functional Requirement	CC dependencies	Satisfied dependencies
FMT_MSA.3/JCVM	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/JCVM FMT_SMR.1
FMT_SMF.1	No dependencies	
FMT_SMR.1	(FIA_UID.1)	FIA_UID.2/AID
FCS_CKM.1/TDES	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1/TDES_CIPHER FCS_COP.1/TDES_MAC FCS_CKM.4
FCS_CKM.1/AES	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1/AES_CIPHER FCS_COP.1/AES_MAC FCS_CKM.4
FCS_CKM.1/RSA	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1/RSA_SIGN FCS_COP.1/RSA_CIPHER FCS_CKM.4
FCS_CKM.1/ECDSA	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1/ECDSA_SIGN FCS_COP.1/ECDH FCS_CKM.4
FCS_CKM.1/HMAC	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1/HMAC FCS_CKM.4
FCS_CKM.4	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2)	FCS_CKM.1/TDES FCS_CKM.1/AES FCS_CKM.1/RSA FCS_CKM.1/ECDSA FCS_CKM.1/HMAC
FCS_COP.1/TDES_CIPHER	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/TDES FCS_CKM.4
FCS_COP.1/TDES_MAC	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/TDES FCS_CKM.4
FCS_COP.1/AES_CIPHER	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/AES FCS_CKM.4
FCS_COP.1/AES_MAC	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/AES FCS_CKM.4
FCS_COP.1/RSA_SIGN	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/RSA FCS_CKM.4
FCS_COP.1/RSA_CIPHER	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/RSA FCS_CKM.4
FCS_COP.1/ECDSA_SIGN	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/ECDSA FCS_CKM.4
FCS_COP.1/ECDH	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/ECDSA FCS_CKM.4
FCS_COP.1/DH	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/RSA FCS_CKM.4
FCS_COP.1/Hash	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	**See rationale**
FCS_COP.1/HMAC	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/HMAC FCS_CKM.4
FCS_COP.1/CRC	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	**See rationale**
FCS_RNG.1	No dependencies	
FDP_RIP.1/ABORT	No dependencies	
FDP_RIP.1/APDU	No dependencies	
FDP_RIP.1/GlobalArray	No dependencies	
FDP_RIP.1/bArray	No dependencies	
FDP_RIP.1/KEYS	No dependencies	
FDP_RIP.1/TRANSIENT	No dependencies	
FDP_ROL.1/FIREWALL	(FDP_ACC.1 or FDP_IFC.1)	FDP_ACC.2/FIREWALL FDP_IFC.1/JCVM
FAU_ARP.1	(FAU_SAA.1)	**See rationale**
FDP_SDI.2/DATA	No dependencies	
FPR_UNO.1	No dependencies	
FPT_FLS.1/JCS	No dependencies	
FPT_TDC.1	No dependencies	
FIA_ATD.1/AID	No dependencies	
FIA_UID.2/AID	No dependencies	

Security Functional Requirement	CC dependencies	Satisfied dependencies
FIA_USB.1/AID	(FIA_ATD.1)	FIA_ATD.1/AID
FMT_MTD.1/JCRE	(FMT_SMF.1) and (FMT_SMR.1)	FMT_SMF.1 FMT_SMR.1
FMT_MTD.3/JCRE	(FMT_MTD.1)	FMT_MTD.1/JCRE
FDP_ACC.2/ADEL	(FDP_ACF.1)	FDP_ACF.1/ADEL
FDP_ACF.1/ADEL	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2/ADEL FMT_MSA.3/ADEL
FDP_RIP.1/ADEL	No dependencies	
FMT_MSA.1/ADEL	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2/ADEL FMT_SMF.1/ADEL FMT_SMR.1/ADEL
FMT_MSA.3/ADEL	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/ADEL FMT_SMR.1/ADEL
FMT_SMF.1/ADEL	No dependencies	
FMT_SMR.1/ADEL	(FIA_UID.1)	**See rationale**
FPT_FLS.1/ADEL	No dependencies	
FDP_RIP.1/ODEL	No dependencies	
FPT_FLS.1/ODEL	No dependencies	
FPT_RCV.3/OS	(AGD_OPE.1)	AGD_OPE.1
FPT_RCV.4/OS	No dependencies	
FDP_SDI.2/ARRAY	No dependencies	
FDP_SDI.2/RESULT	No dependencies	
FDP_SDI.2/MONOTONIC_COUNTER	No dependencies	
FDP_SDI.2/CRT_MNGT	No dependencies	
FCS_COP.1/CRT_MNGT	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	FCS_CKM.1/RSA FCS_CKM.1/ECDSA FCS_CKM.4
FCS_CKM.5/KDF	(FCS_CKM.2 or FCS_COP.1) and (FCS_CKM.4)	FCS_COP.1/AES_MAC FCS_COP.1/Hash FCS_COP.1/HMAC FCS_CKM.4
FPT_STM.1/SYS_TIME	No dependencies	

Rationale for the exclusion of dependencies:

- **The dependency FMT_MSA.1 of FMT_MSA.3/OS-UPDATE is unsupported.**
  No history information has to be kept by the TOE.
- **The dependency FMT_SMF.1 of FMT_MSA.1/JCRE is unsupported.**
  The dependency between FMT_MSA.1/JCRE and FMT_SMF.1 is not satisfied because no management functions are required for the Java Card RE.
- **The dependencies of FCS_COP.1/Hash are unsupported**
  Hash operation does not require any key.
- **The dependencies of FCS_COP.1/CRC are unsupported**
  CRC operations do not require any key.
- **The dependency FAU_SAA.1 of FAU_ARP.1 is unsupported**
  The dependency of FAU_ARP.1 on FAU_SAA.1 assumes that a “potential security violation” generates an audit event. On the contrary, the events listed in FAU_ARP.1 are self-contained (arithmetic exception, ill-formed bytecodes, access failure) and ask for a straightforward reaction of the TSFs on their occurrence at runtime. The JCVM or other components of the TOE detect these events during their usual working order. Thus, there is no mandatory audit recording in this ST.
- **The dependency FIA_UID.1 of FMT_SMR.1/ADEL is unsupported**
  This ST does not require the identification of the “deletion manager” since it can be considered as part of the TSF.

9.3.4 *SAR – Evaluation Assurance Level Rationale*

The EAL4 package and addition of ALC_DVS.2 and AVA_VAN.5 are required by [PP-GP].

9.3.5 *SAR – Dependency rationale*

Security Assurance Requirement	CC dependencies	Satisfied dependencies
ADV_ARC.1	(ADV_FSP.1) and (ADV_TDS.1)	ADV_FSP.4 ADV_TDS.3
ADV_FSP.4	(ADV_TDS.1)	ADV_TDS.3
ADV_TDS.3	(ADV_FSP.4)	ADV_FSP.4
ADV_IMP.1	(ADV_TDS.3) and (ALC_TAT.1)	ADV_TDS.3 ALC_TAT.1
AGD_OPE.1	(ADV_FSP.1)	ADV_FSP.4
AGD_PRE.1	No dependencies	
ALC_CMC.4	(ALC_CMS.1) and (ALC_DVS.1) and (ALC_LCD.1)	ALC_CMS.4 ALC_DVS.2 ALC_LCD.1
ALC_CMS.4	No dependencies	
ALC_DEL.1	No dependencies	
ALC_DVS.2	No dependencies	
ALC_LCD.1	No dependencies	
ALC_TAT.1	(ADV_IMP.1)	ADV_IMP.1
ASE_CCL.1	(ASE_ECD.1) and (ASE_INT.1) and (ASE_REQ.1)	ASE_ECD.1 ASE_INT.1 ASE_REQ.2
ASE_ECD.1	No dependencies	
ASE_INT.1	No dependencies	
ASE_OBJ.2	(ASE_SPD.1)	ASE_SPD.1
ASE_REQ.2	(ASE_ECD.1) and (ASE_OBJ.2)	ASE_ECD.1 ASE_OBJ.2
ASE_SPD.1	No dependencies	
ASE_TSS.1	(ADV_FSP.1) and (ASE_INT.1) and (ASE_REQ.1)	ADV_FSP.4 ASE_INT.1 ASE_REQ.2
ATE_COV.2	(ADV_FSP.2) and (ATE_FUN.1)	ADV_FSP.4 ATE_FUN.1
ATE_DPT.1	(ADV_ARC.1) and (ADV_TDS.2) and (ATE_FUN.1)	ADV_ARC.1 ADV_TDS.3 ATE_FUN.1
ATE_FUN.1	(ATE_COV.1)	ATE_COV.2
ATE_IND.2	(ADV_FSP.2) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_COV.1) and (ATE_FUN.1)	ADV_FSP.4 AGD_OPE.1 AGD_PRE.1 ATE_COV.2 ATE_FUN.1
AVA_VAN.5	(ADV_ARC.1) and (ADV_FSP.4) and (ADV_IMP.1) and (ADV_TDS.3) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_DPT.1)	ADV_ARC.1 ADV_FSP.4 ADV_IMP.1 ADV_TDS.3 AGD_OPE.1 AGD_PRE.1 ATE_DPT.1

The table here-above shows that all SAR dependencies are met.

9.4 COMPOSITION TASKS – SFR PART

The following table (see next page) lists the SFRs that are declared in the security target [ST_IC], and separates them in relevant platform[208]-SFRs (RP_SFR-SERV and RP_SFR-MECH[209]) and irrelevant platform-SFRs (IP_SFR), as requested in [CCDB]. The table also provides the link between the relevant platform-SFRs and the composite product SFRs.

[208] Using the composition tasks terminology, the platform is the S3SSE2A chip.
[209] RP_SFR-SERV designates relevant IC SFRs used by the composite TOE to implement security services with associated TSFI. RP_SFR-MECH designates relevant IC SFRs used by the composite TOE as mechanisms to provide global protection against attacks.

Platform-SFR	Platform-SFR title	Addressing (as stated in platform ST)	RP_SFR-SERV	RP_SFR-MECH	IP_SFR	Composite product SFRs
FRU_FLT.2	Limited fault tolerance	Malfunctions		X		No direct link to composite TOE SFRs but provides global protection against attacks.
FPT_FLS.1	Failure with preservation of secure state			X		No direct link to composite TOE SFRs but provides global protection against attacks.
FMT_LIM.1	Limited capabilities	Abuse of Functionality		X		No direct link to composite TOE SFRs but provides global protection against attacks.
FMT_LIM.2	Limited availability			X		No direct link to composite TOE SFRs but provides global protection against attacks.
FAU_SAS.1	Audit storage		X			No direct link to composite TOE SFRs but used for the composite-product identification.
FDP_SDC.1	Stored data confidentiality	Physical Manipulation and Probing		X		No direct link to composite TOE SFRs but provides global protection against attacks.
FDP_SDI.2	Stored data integrity monitoring and action			X		No direct link to composite TOE SFRs but provides global protection against attacks.
FPT_PHP.3	Resistance to physical attack			X		No direct link to composite TOE SFRs but provides global protection against attacks.
FDP_ITT.1	Basic internal transfer protection	Leakage		X		FPR_UNO.1
FPT_ITT.1	Basic internal TSF data transfer protection			X		FPR_UNO.1
FDP_IFC.1	Subset information flow control			X		FPR_UNO.1
FCS_RNG.1/PTG.2	Random number generation - PTG.2	Random Numbers	X			FCS_RNG.1

Platform-SFR	Platform-SFR title	Addressing (as stated in platform ST)	RP_SFR-SERV	RP_SFR-MECH	IP_SFR	Composite product SFRs
FCS_RNG.1/RGS-IC	Random number generation – RGS-IC				X	The RNG of the composite TOE is AIS31-compliant and leverages FCS_RNG.1/PTG.2 from [ST_IC].
FDP_ACC.1	Subset access control	Memory Access Control	X			FDP_ACC.2/FIREWALL
FDP_ACF.1	Security attribute based access control		X			FDP_ACF.1/FIREWALL
FMT_MSA.3	Static attribute initialization		X			FMT_MSA.3/FIREWALL FMT_MSA.3/JCVM
FMT_MSA.1	Management of security attributes		X			FMT_MSA.1/JCRE FMT_MSA.1/JCVM
FMT_SMF.1	Specification of management functions		X			FMT_SMF.1
FCS_COP.1/TDES	Cryptographic operation – TDES	Cryptographic Support	X			FCS_COP.1/TDES_CIPHER FCS_COP.1/TDES_MAC
FCS_CKM.4/TDES	Cryptographic Key destruction – TDES		X			FCS_CKM.4
FCS_COP.1/AES	Cryptographic operation – AES		X			FCS_COP.1/AES_CIPHER FCS_COP.1/AES_MAC
FCS_CKM.4/AES	Cryptographic key destruction – AES		X			FCS_CKM.4
FMT_LIM.1/Loader	Limited capabilities - Loader	Bootloader			X	Not applicable to the composite TOE, as the IC Loader is no more available after phase 5.
FMT_LIM.2/Loader	Limited availability - Loader		X			No direct link to composite TOE SFRs, however this IC SFR participates to the composite TOE protection during phases 6 and 7.
FTP_ITC.1	Inter-TSF trusted channel		X			No direct link to composite TOE SFRs, since the IC Loader is no more available after phase 5. However, these IC SFRs are essential to protect the
FDP_UCT.1	Basic data exchange confidentiality		X			
FDP_UIT.1	Data exchange integrity		X			
FDP_ACC.1/Loader	Subset access control - Loader		X			

Platform-SFR	Platform-SFR title	Addressing (as stated in platform ST)	RP_SFR-SERV	RP_SFR-MECH	IP_SFR	Composite product SFRs
FDP_ACF.1/Loader	Security attribute based access control - Loader		X			composite TOE during phases 4 and 5 (covered by the ALC assurance classes).
FIA_API.1	Authentication Proof of Identity	Authentication Proof of Identity	X			No direct link to composite TOE SFRs, since the IC Loader is no more available after phase 5. However, this IC SFR is essential to protect the composite TOE during phases 4 and 5 (covered by the ALC assurance classes).

10 TOE summary specification

This section provides a summary of the security functions implemented by the TOE in order to fulfil the security functional requirements. The security functionalities concerning the IC are described in [ST_IC] and are not redefined in this security target, although they must be considered for the TOE.

10.1 TEQS V1.0 PLATFORM

GP.CardContentManagement
This security function provides the capability and a dedicated flow control for the loading, installation, extradition, registry update, selection and removal of eSE content and especially executable files and application instances. Such features are offered to the OEM and its business partners, allowing the OEM to delegate eSE content management to an Application Provider according to privileges assigned to the various security domains on the eSE. It supports Delegated management (DM), Authorized management (AM) and it can use DAP or Mandated DAP verification and generation of Reception token. It also checks that only the eSE management commands specified and allowed at each state of the eSE's life cycle are accepted, and ill-formed ones are rejected with an appropriate error response.

GP.KeyLoading
This security function provides the capability and a dedicated flow control for the loading of keys and other sensitive data using the GlobalPlatform STORE DATA and PUT KEY APDUs, or by using GlobalPlatform APIs for loading and storing data and keys.

GP.SecurityDomain
This security function provides security domain management, as SD creation, SD selection, SD privileges setting and SD deletion in SD hierarchy. It provides means to associate or extradite an application to a security domain in order to provide services (as secure channel) to the dedicated application without sharing the related keys stored in SD. It also provides Keyset Management in SD, with Key Set creation, Key set deletion, key importation, replacement, or deletion in Key Set.
Security Domains are privileged Applications as defined in [GPCS] § 7, holding cryptographic keys to be used to support Secure Channel Protocol operations and/or to authorize card content management functions. There are different types of security domain with dedicated privileges and associated operations: ISD Security domain, Supplementary Security domains, and Controlling Authority Security domains.
ISD Security domain as defined in [GPCS] §7.1.1, is the mandatory Security Domain, implicitly selected if the Application implicitly selectable on the same logical channel of the same I/O interface is removed. It inherits of the Final Application privilege if the Application with that privilege is removed.
Supplementary Security Domains are privileged Applications with dedicated privileges:

- Token Verification Privilege as described in [GPCS] §9.1.3.1
- Authorized Management Privilege as described in [GPCS] §9.1.3.2
- Delegated Management Privilege as described in [GPCS] §9.1.3.3
- Global Delete Privilege as described in [GPCS] §9.1.3.4
- Global Lock Privilege as described in [GPCS] §9.1.3.5
- Receipt Generation Privilege as described in [GPCS] §9.1.3.6
- Ciphered Load File Data Block Privilege as described in [GPCS] §9.1.3.7

Controlling Authority Security Domain is a supplementary Security Domain dedicated to the Controlling Authority with dedicated privileges. It contains Security Domains cryptographic keys needed to confidentially personalize an initial set of Secure Channel Keys of an APSD.

GP.SecureChannel

This security function provides a secure communication channel between the eSE and an external entity during an Application Session according to [GPCS], [Amd D] and [Amd F]. It provides an APDU flow control using the Command security level check according to eSE Life cycle and type of APDU.

A Secure Channel Session is divided into three sequential phases:

- Secure Channel Initiation when the Application on the eSE and the external entity have exchanged sufficient information enabling them to perform the required cryptographic functions. The Secure Channel Session initiation always includes (at least) the authentication of the external entity by the Application on the eSE; performing also the setting of the Command security level used for the session.
- Secure Channel Operation when the Application on the eSE and the external entity exchange data within the cryptographic protection of the Secure Channel Session. The Secure Channel services offered may vary from one Secure Channel Protocol to the other;
- Secure Channel Termination when either the Application on the eSE or the external entity determines that no further communication is required or allowed via an established Secure Channel Session.

The following services are provided by the Secure Channel:

- Entity authentication in which the eSE or the external entity proves its authenticity to the other entity through a cryptographic exchange, based on session key generation and a dedicated flow control;
- Integrity and authentication in which the receiving entity (the eSE or the external entity) ensures that the data being received from the sending entity (respectively the external entity or the eSE) actually came from an authenticated entity in the correct sequence and has not been altered;
- Confidentiality in which data being transmitted from the sending entity (the external entity or the eSE) to the receiving entity (respectively the eSE or the external entity) is not viewable by an unauthenticated entity.

The following Secure Channel Protocols are supported by the TOE: SCP02, SCP03 and SCP11.

GP.GPRegistry

This security function provides management and access to the GlobalPlatform Registry used for:

- Store eSE management information;
- Store relevant application management information (e.g., AID, associated Security Domain and Privileges);
- Support eSE resource management data;
- Store Application Life Cycle information;
- Store eSE Life Cycle information;
- Track any counters associated with logs.

The content of the GlobalPlatform Registry may be accessed by administrative commands or by applet using a dedicated GlobalPlatform API.

Only secure values are accepted for the information stored in the GlobalPlatform registry (including Life Cycle states, Security Levels and Privileges).

GP.TrustedFramework

This security function provides a trusted path for inter-application communication, according to the Trusted Framework defined in [GPCS]. The trusted path provides assured identification of its end points and protection of the communicated data from modification and disclosure. Targeted use case is application personalization, where the GlobalPlatform Trusted Framework forwards the unwrapped command (STORE DATA) to the Target Application indicated by the Receiving SD through its GlobalPlatform Application interface.

GP.CLFDB
This security function handles the decryption of Ciphered Load File Data Blocks according to [GPCS] section C.6. Decryption is done using either TDES (112 bits key length) with CBC mode, or AES with CBC mode with a null ICV (128, 192, or 256 bits key length).

GP.GlobalServices
This security function implements the controls related to Global Services Applications, as described in [GPCS] section 8.1: access control, management and initialization of security attributes, roles.

GP.CVM
This security function implements the controls related to the CVM services, as described in [GPCS] section 8.2. The Global PIN is blocked after a configured number of unsuccessful (and consecutive) PIN verification attempts is reached; this number is comprised between 1 and 255 and is set during eSE personalization (phase 6). The comparison between the PIN value provided by the mobile phone holder and the reference PIN is done securely within the TOE (in particular, without any leakage that could allow an observer to gain information on the PIN value).

GP.DelegatedManagement
The TOE implements the verification of DM tokens and generation of DM receipts as specified in [GPCS] sections C.4 and C.5. The following algorithms are supported for both operations:
- TDES (112 bits key length) according to [GPCS] section B.1.2.2
- AES (128, 192, or 256 bits key length) according to [GPCS] section B.2.2
- RSA (1024 or 2048 bits key length) according to [GPCS] section B.3.1.1 / B3.2.1
- ECC (256, 384, or 512 bits key length) according to [GPCS] section B.4.3.

GP.DAP
The TOE implements the verification of DAP (and Mandated DAP) blocks as specified in [GPCS] sections C.2 and C.3. The following algorithms are supported:
- SHA-1, SHA-256, SHA-384, or SHA-512 for the hash computation
- TDES (112 bits key length), AES (128, 192, or 256 bits key length), RSA (1024 or 2048 bits key length) or ECC (256, 384, or 512 bits key length) for the DAP signature verification. This verification is done at the time an ELF with DAP is received.

GP.ELFU
The TOE implements the ELF Upgrade capability according to [Amd H]. Associated access control rules are enforced, as defined in [Amd H]. Management functions include the Saving, Loading, Restore phases of the Executable Load File Process, the management of the ELF upgrade session status and the eSE management during the ELF upgrade session. Rollback of deletion operations is supported under the following rules:
- If the deletion of the application instances and ELF(s) (atomic and irreversible operation) was started and then interrupted and/or disturbed by for example unexpected power-down, it shall automatically restart and complete at next power-up.
- If the interruption occurred during the Deletion Sequence and the latter did not complete automatically (i.e. the irreversible deletion operation did not start already), the Deletion Sequence shall restart.

A secure state is preserved when the following types of failures occur:
- The required minimum amount of memory is not available at the time the command MANAGE ELF UPGRADE is received
- A fatal error occurs using the new ELF version during the Restore Phase
- The ELF Upgrade Recovery Procedure fails
- The installation of an Application instance fails
- An interruption occurred during the Installation, Saving, Restore, or Consolidation Sequences.

GP.OS-UPDATE

The TOE implements an OS Update capability by means of the GemActivate proprietary mechanism, allowing the Connected eSE 5.3.4 Platform to be updated post-issuance (during phase 7 of the TOE life-cycle). OS updates are performed through the loading, installation and activation of related ELFs, fulfilling the same rules as for any other ELF. DAP verification (AES128 CMAC) is mandatory for ELFs containing OS updates, ensuring the authenticity and integrity protection of the code update, and the content of the ELF is directly encrypted (AES128 in CBC mode) with a dedicated encryption key, ensuring the confidentiality protection. Note that both the DAP signature verification key and the encryption key are GemActivate keys, meaning that OS updates can only be issued and decrypted by Thales. Verification of TOE identification data is also enforced before allowing any OS update. The whole OS update operation is done through an atomic process, ensuring the permanent consistency between the Connected eSE 5.3.4 Platform active code and its identification data.

A secure state is preserved in case of failure during the OS update process. More precisely:

- There are 3 steps in an OS Update operation:
  - step 1: loading
  - step 2: activation
  - step 3: update of TOE identification data

  Steps 2 and 3 are performed atomically, so that the TOE active code and identification data always remain consistent.
- If a failure (interruption or incident) occurs during step 1 (loading), then the TOE remains in its initial state (no update, neither of code nor of the TOE identification data).
- If a failure (interruption or incident) occurs during the atomic sequence step 2 / step 3 (activation / update of TOE identification data), then the enforced behavior depends on the nature of the update:
  - For java code updates, the TOE remains in its initial state and the OS Update operation is aborted.
  - For native code updates, the TOE does some retries to complete the atomic sequence step 2 / step 3 (activation / update of TOE identification data) until it is successful.
  - In any case, only two possible secure states are possible at any given time:
    - Either activation is not done and the TOE identification data is not updated (i.e. initial state)
    - Or the atomic sequence completes successfully, i.e. the OS update is activated and the TOE identification data is updated accordingly.

JCS.APDUBuffer

The security function maintains a byte array buffer accessible from any applet context. This buffer is used to transfer incoming APDU header and data bytes as well as outgoing data according to [JCAPI310]. The APDU class API is designed to be transport protocol independent (as defined in [ISO 7816] Book 3).
Application note: ADPU buffer is a JCRE temporary entry point object where no associated reference can be stored in a variable or an array component.

JCS.ByteCodeExecution

This security function handles applet bytecode execution according to the rules defined in [JCVM310]. The JCVM execution may be summarized in JCVM interpreter start-up, bytecode execution and JCVM interpreter loop. The applet bytecode execution consists in:

- fetching the next bytecode to execute according to the applet's code flow control,
- decoding the next bytecode,
- executing the fetched bytecode.

The JCVM manages several types of objects, such as persistent objects, transient objects, persistent arrays (boolean, byte, short, int or reference), transient arrays (boolean, byte, short, int or reference) and static field images. For each type of object, different types of control are performed.

JCS.Firewall
This security function enforces a Firewall access control policy and a JCVM information flow control policy at runtime. It defines how accessing the following items: Static Class Fields, Array Objects, Class Instance Object Fields, Class Instance Object Methods, Standard Interface Methods, Shareable Interface Methods, Classes, Standard Interfaces, Shareable Interfaces, Array Object Methods.
Based on security attributes (Sharing, Context, Lifetime), it performs access control to object fields between objects and throws security exception when access is denied. Thus, it enforces applet isolation located in different packages and controls the access to global data containers shared by all applet instances.
The JCRE shall allocate and manage a context for each Java API package containing applets. The JCRE maintains for its own context a special system privilege so that it can perform operations that are denied to contexts of applets.

JCS.Package
This security function manages packages. A package is a structural item defined for naming, loading, storing, execution context definition. There are rules for package identification, for structure check and access rules definition. If inconsistent items are found during checks, an error message is sent.

JCS.CryptoAPI
This security function offers the following cryptographic services to applets through the JavaCard API:

- Generation of random numbers as defined in [JCAPI310] to be used for key values or challenges during external exchanges. The Random Number Generator (RNG) is hybrid deterministic and conformant to [AIS31] DRG.4, providing enhanced backward secrecy & enhanced forward secrecy. It passes [AIS31] test procedure A.
- Computation of checksum CRC16 and CRC32 conformant with ISO3309, as defined in [JCAPI310] Checksum class. Both ALG_ISO3309_CRC16 and ALG_ISO3309_CRC32 are supported.
- Encryption and decryption using TDES algorithm as defined in [JCAPI310] Cipher class. The following algorithms are supported: ALG_DES_CBC_NOPAD, ALG_DES_CBC_ISO9797_M1, ALG_DES_CBC_ISO9797_M2, ALG_DES_CBC_PKCS5, ALG_DES_ECB_NOPAD, ALG_DES_ECB_ISO9797_M1, ALG_DES_ECB_ISO9797_M2 and ALG_DES_ECB_PKCS5. Both TDES 2-keys (112 bits key length) and TDES 3-keys (168 bits key length) are supported.
- Generation of 4-byte or 8-byte MAC using TDES algorithm as defined in [JCAPI310] Signature class. The following algorithms are supported: ALG_DES_MAC4_ISO9797_1_M1_ALG3, ALG_DES_MAC4_ISO9797_1_M2_ALG3, ALG_DES_MAC4_ISO9797_M1, ALG_DES_MAC4_ISO9797_M2, SIG_CIPHER_DES_MAC4, ALG_DES_MAC4_PKCS5, 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, SIG_CIPHER_DES_MAC8, ALG_DES_MAC8_PKCS5 and ALG_DES_MAC8_NOPAD. Both TDES 2-keys (112 bits key length) and TDES 3-keys (168 bits key length) are supported.
- Encryption and decryption using AES (128, 192 or 256 bits key) algorithm as defined in [JCAPI310] Cipher and AEADCipher classes. The following algorithms are supported: ALG_AES_BLOCK_128_CBC_NOPAD, ALG_AES_CBC_ISO9797_M1, ALG_AES_CBC_ISO9797_M2, ALG_AES_CBC_PKCS5, ALG_AES_BLOCK_128_ECB_NOPAD, ALG_AES_ECB_ISO9797_M1, ALG_AES_ECB_ISO9797_M2, ALG_AES_ECB_PKCS5, ALG_AES_CTR, ALG_AES_CCM, CIPHER_AES_CCM, ALG_AES_GCM and CIPHER_AES_GCM.

- Generation of 16-byte, 24-byte or 32-byte MAC using AES algorithm (128, 192 or 256 bits key) in CBC mode as defined in [JCAPI310] Signature class. The following algorithms are supported: ALG_AES_MAC_128_NOPAD, SIG_CIPHER_AES_MAC128, SIG_CIPHER_AES_CMAC128, ALG_AES_CMAC_128, ALG_AES_MAC_192_NOPAD and ALG_AES_MAC_256_NOPAD.
- Data hash computation as defined in [JCAPI310] MessageDigest class. The following algorithms are supported: ALG_SHA, ALG_SHA_224, ALG_SHA_256, ALG_SHA_384 and ALG_SHA_512.
- HMAC computation as defined in [JCAPI310] Signature class. The following algorithms are supported: ALG_HMAC_SHA1, ALG_HMAC_SHA_256, ALG_HMAC_SHA_384, ALG_HMAC_SHA_512 and SIG_CIPHER_HMAC.
- Encryption and decryption using RSA with Standard or CRT modes, as defined in [JCAPI310] Cipher class. The following algorithms are supported: ALG_RSA_NOPAD, ALG_RSA_PKCS1 and ALG_RSA_PKCS1_OAEP. All key lengths from 1024 to 2048 bits (by steps of 32 bits) are supported.
- Generation and verification of RSA signatures in Standard or CRT modes, as defined in [JCAPI310] Signature class. The following algorithms are supported: 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, ALG_RSA_SHA_ISO9796_MR, ALG_RSA_SHA_PKCS1, ALG_RSA_SHA_PKCS1_PSS, ALG_RSA_SHA_RFC2409 and SIG_CIPHER_RSA. All key lengths from 1024 to 2048 bits (by steps of 32 bits) are supported.
- Generation and verification of ECDSA signatures as defined in [JCAPI310] Signature class. The following algorithms are supported: ALG_ECDSA_SHA, ALG_ECDSA_SHA_224, ALG_ECDSA_SHA_256, ALG_ECDSA_SHA_384, ALG_ECDSA_SHA_512, SIG_CIPHER_ECDSA and SIG_CIPHER_ECDSA_PLAIN. Elliptic curve cryptography over GF(p) is considered here, with P ranging from 256 to 521 bits.
- Secret key agreement according to the ECDH algorithm, as defined in [JCAPI310] KeyAgreement class. The following algorithms are supported: ALG_EC_SVDP_DH_KDF, ALG_EC_SVDP_DH_PLAIN, ALG_EC_SVDP_DH_PLAIN_XY, ALG_EC_SVDP_DHC_KDF and ALG_EC_SVDP_DHC_PLAIN. Elliptic curve cryptography over GF(p) is considered here, with P ranging from 256 to 521 bits.
- Key Exchange according to the DH algorithm. RSA key sizes ranging from 1024 to 2048 bits (by steps of 32 bits) are supported.

These operations are performed in a way to avoid revealing the key values. If the applet specifies an algorithm that the platform does not support, the JCRE refuses to perform the cryptographic operation and generates an exception.

JCS.KeyManagement

This security function enforces key management for the different associated operations: key building and generation, key importation, key exportation, key masking and key destruction using the standard API defined in [JCAPI310].

- Key generation implemented through KeyBuilder and/or KeyPair classes : TDES key generation (112 or 168 bits), AES key generation (128, 192 or 256 bits), RSA Standard and RSA CRT Key Pair Generation (1024 to 2048 bits by steps of 32 bits), ECDSA Key Pair Generation (P ranging from 256 to 521 bits) and HMAC Key generation.
- Key importation and exportation is done using method protecting confidentiality and integrity of key.
- Key masking protects the confidentiality of cryptographic keys from being read out from the memory. It ensures the service of accessing and modifying them.

- Key destruction (implemented through the method clearKey() of the Key class) disables the use of a key both logically and physically.

JCS.OwnerPIN
This security function provides to applets a means to perform user identification and authentication with the OwnerPin class conformant to [JCAPI310].
It offers to create a PIN and store it securely in the persistent memory. It allows access to PIN value only to perform a secure comparison between a PIN stored in the persistent memory and a data received as parameter.
A method returns a positive result if a valid Pin has been presented during current session. If the PIN is not blocked and the comparison is successful, the validated flag is set and the try counter is set to its maximum, otherwise the authentication fails and the associated try counter is decremented. When the validated flag is set, it is assumed that the user is authenticated.
When the try counter reaches zero, the PIN is blocked and the authentication is no more possible until the PIN is unblocked.

JCS.EraseResidualData
This security function ensures that sensitive data are locked upon the following operations as defined in [JCRE310]:
- Deletion of package and/or applications,
- Deletion of objects.

They are erased when space needs to be reused for allocation of new objects.
This security function also ensures that the sensitive temporary buffers (transient object, bArray object, Global Array object, APDU buffer, Cryptographic buffer) are securely cleared after their usage with respect to their life-cycle and interface as defined in [JCRE310], transient object at reset or allocation and persistent object are erased at allocation for new object.

JCS.OutOfLifeDataUndisclosure
This security function ensures that sensitive data are locked until postponed erasure on the following operations: Deletion of persistent and transient objects according to [JCRE310].

JCS.RunTimeExecution
This security function provides a secure run time environment conformant to [JCRE310] and deals with:
- Instance registration or deletion,
- Application selection,
- Applet opcode execution,
- JCAPI methods execution,
- Logical channel management,
- APDU flow control, dispatch and buffer management,
- JCRE memory and context management,
- JCRE reference deletion,
- JCRE access rights,
- JCRE throw exception,
- JCRE security reaction.

JCS.Exception
This security function manages throwing of an instance of Exception class in the following cases:
- a SecurityException when an illegal access to an object is detected,
- a SystemException with an error code describing the error condition,
- a CryptoException in case of algorithm error or illegal use,

- any exception decided by the applet or the JCRE handled as temporary JCRE entry point object with associated JCAPI. It also offers a means to applet to handle exception and to JCRE to handle uncaught exception by applets.

JCS.SensitiveArray
The TOE implements the 'SensitiveArrays' optional Javacard package. This security function ensures the integrity protection of the user data stored in arrays created by the makeIntegritySensitiveArray() method of the javacard.framework.SensitiveArrays class. An exception is thrown upon detection of an integrity error.

JCS.SensitiveResult
The TOE implements the 'SensitiveResult' optional Javacard package. This security function ensures the integrity protection of the sensitive API result stored in the javacardx.security.SensitiveResult class. An exception is thrown upon detection of an integrity error, additionally the TSF will mute the card further if redundancy checking of data integrity detects an error.

JCS.MonotonicCounters
The TOE implements the 'MonotonicCounter' optional Javacard package. This security function ensures the integrity protection of the counter value in the MonotonicCounter object. An exception is thrown upon detection of an integrity error.

JCS.CryptoCertManagement
The TOE implements the 'Cryptographic Certificate Management' optional Javacard package. This security function ensures the integrity protection of cryptographic certificates; an exception is thrown upon detection of an integrity error. The TOE supports X.509 certificate verification according to the following algorithms:

- ALG_TYPE_EC_FP_PUBLIC with key length ranging from 256 bits to 521 bits, related standard IEEE P1363
- ALG_TYPE_RSA_PUBLIC with key length ranging from 1024 bits to 2048 bits, related standard NIST SP800-56Ar2.

JCS.KDF
The TOE partially implements the 'Key Derivation Functions (KDF)' optional Javacard package. The following table lists the supported Key Derivation Functions, the underlying cryptographic algorithms as well as supported key sizes, and the related standards:

Key Derivation Function	Cryptographic algorithms as defined in [JCAPI310]	Key sizes as defined in [JCAPI310]	Standards
ALG_KDF_COUNTER_MODE	Signature.ALG_HMAC_SHA1 Signature.ALG_HMAC_SHA_256 Signature.ALG_AES_CMAC_128	LENGTH_SHA LENGTH_SHA_256 LENGTH_AES_128	NIST SP800-108
ALG_PRF_TLS12	Signature.ALG_HMAC_SHA1 Signature.ALG_HMAC_SHA_256	LENGTH_SHA LENGTH_SHA_256	IETF RFC 5246
ALG_KDF_ICAO_MRTD	MessageDigest.ALG_SHA MessageDigest.ALG_SHA_256	LENGTH_SHA LENGTH_SHA_256	ICAO MRTD Doc 9303
ALG_KDF_ANSI_X9_63	MessageDigest.ALG_SHA_224 MessageDigest.ALG_SHA_256 MessageDigest.ALG_SHA_384 MessageDigest.ALG_SHA_512	LENGTH_SHA_224 LENGTH_SHA_256 LENGTH_SHA_384 LENGTH_SHA_512	ANSI X9.63
ALG_KDF_HKDF	Signature.ALG_HMAC_SHA1 Signature.ALG_HMAC_SHA_256	LENGTH_SHA LENGTH_SHA_256	IETF RFC 5869

JCS.SystemTime
The TOE implements the 'System Time' optional Javacard package (javacardx.framework.time). This security function ensures that the TOE is able to provide reliable time stamps.

OS.MemoryManagement
This security function allocates memory areas and performs access control on them to avoid unauthorized access. It manages circular writing to avoid instable memory state. It enforces memory recovery in case of error detection. It offers (when required) confidentiality services for data storage: Ciphering / Deciphering of Data in RAM or in FLASH, Scrambling / Unscrambling of Data in RAM or in FLASH.

OS.Atomicity
This security function performs write operations atomically on complex type or object in order to avoid incomplete update. Prior to be written, data is stored in an atomic back-up area. In case on writing interrupt, the only two possible values are: initial value if writing is not started or final value if writing is started. At next start-up, the atomic back-up area is checked to finalize interrupted writing.

10.2 TSS RATIONALE

Security Functional Requirement	Coverage by TSS Security Function(s)
FDP_IFC.2/GP-ELF	This SFR is covered by GP.CardContentManagement managing flow control for loading and installing application instances.
FDP_IFF.1/GP-ELF	This SFR is covered by GP.CardContentManagement managing flow control for loading and installing application instances.
FDP_ITC.2/GP-ELF	This SFR is covered by JCS.Package checking the binary compatibility of dependent packages using their version numbers and AIDs prior to installation operations.
FDP_IFC.2/GP-KL	This SFR is covered by GP.KeyLoading, GP.SecurityDomain and GP.SecureChannel.
FDP_IFF.1/GP-KL	This SFR is covered by GP.KeyLoading, GP.SecurityDomain and GP.SecureChannel.
FDP_ITC.2/GP-KL	This SFR is covered by GP.KeyLoading.
FMT_MTD.1/GP-LC	This SFR is covered by GP.CardContentManagement, GP.SecurityDomain and GP.GPRegistry.
FMT_MTD.1/GP-PR	This SFR is covered by GP.CardContentManagement, GP.SecurityDomain and GP.GPRegistry.
FCS_CKM.1/GP-SCP	This SFR is covered by GP.SecureChannel.
FCS_COP.1/GP-SCP	This SFR is covered by GP.SecureChannel.
FTP_TRP.1/GP-TF	This SFR is enforced by GP.TrustedFramework.
FMT_MSA.1/GP	This SFR is covered by GP.SecureChannel providing an APDU flow control using the Command security level check according to Card Life cycle and type of APDU.
FMT_MSA.3/GP	This SFR is covered by GP.SecureChannel providing setting of the default value.
FMT_SMR.1/GP	This SFR is covered by JCS.RunTimeExecution and GP.SecurityDomain managing the roles: S.OPEN, issuer, application provider, verification authority and controlling authority.
FMT_SMF.1/GP	This SFR is covered by GP.SecurityDomain and GP.SecureChannel.
FPT_RCV.3/GP	This SFR is addressed by JCS.RunTimeExecution, OS.MemoryManagement, GP.GPRegistry and GP.CardContentManagement covering the applet instance erasure when applet instance registration operation fails.
FPT_FLS.1/GP	This SFR is addressed by JCS.Package, JCS.RunTimeExecution and GP.CardContentManagement covering the applet instance registration operations and associated error handling.
FPT_TDC.1/GP	This SFR is addressed by GP.CardContentManagement, GP.SecureChannel and GP.KeyLoading.
FTP_ITC.1/GP	This SFR is addressed by GP.SecureChannel.
FCO_NRO.2/GP	This SFR is covered by GP.SecureChannel managing the secure channel protocol where several checks are performed prior ELF or Key loading: * mutual authentication between the external entity (Issuer or Application provider) and the selected security Domain, including creation of a session key, * by the verification of a (chained) MAC that the Issuer or Application provider attaches to each file or data block

Security Functional Requirement	Coverage by TSS Security Function(s)
	sent, * by the erase of the session key at the end of the session.
FIA_UID.1/GP	This SFR is covered by JCS.RunTimeExecution and GP.SecurityDomain controlling accessible action prior identification and action when SD or application associated to SD are selected.
FDP_UIT.1/GP	This SFR is covered by GP.SecureChannel providing a session key generation. It ensures that the whole package or data has been correctly received.
FDP_ROL.1/GP	This SFR is addressed by GP.CardContentManagement, GP.KeyLoading and OS.Atomicity.
FDP_UCT.1/GP	This SFR is covered by GP.SecureChannel which provides confidentiality protection for sensitive data (such as secret keys).
FPR_UNO.1/GP	This SFR is covered by JCS.RunTimeExecution and JCS.CryptoAPI.
FIA_UAU.1/GP	This SFR is covered by JCS.RunTimeExecution and GP.SecurityDomain (as for FIA_UID.1/GP).
FIA_UAU.4/GP	This SFR is covered by GP.SecureChannel.
FIA_AFL.1/GP	This SFR is covered by GP.SecureChannel.
FMT_MTD.3/GP	This SFR is covered by GP.GPRegistry.
FCS_COP.1/GP-CLFDB	This SFR is covered by GP.CLFDB.
FDP_ACC.1/GP-GS	This SFR is addressed by GP.GlobalServices.
FDP_ACF.1/GP-GS	This SFR is addressed by GP.GlobalServices.
FMT_MSA.1/GP-GS	This SFR is addressed by GP.GlobalServices.
FMT_MSA.3/GP-GS	This SFR is addressed by GP.GlobalServices.
FMT_SMR.1/GP-GS	This SFR is addressed by GP.GlobalServices.
FMT_SMF.1/GP-GS	This SFR is addressed by GP.GlobalServices.
FIA_AFL.1/GP-CVM	This SFR is addressed by GP.CVM.
FPR_UNO.1/GP-CVM	This SFR is addressed by GP.CVM.
FCO_NRR.1/GP-RECEIPT	This SFR is addressed by GP.DelegatedManagement.
FCO_NRO.2/GP-TOKEN	This SFR is addressed by GP.DelegatedManagement.
FCS_COP.1/GP-TOKEN	This SFR is addressed by GP.DelegatedManagement.
FCS_COP.1/GP-RECEIPT	This SFR is addressed by GP.DelegatedManagement.
FCS_COP.1/GP-DAP_SHA	This SFR is addressed by GP.DAP.
FCS_COP.1/GP-DAP_VER	This SFR is addressed by GP.DAP.
FCO_NRO.2/GP-DAP	This SFR is addressed by GP.DAP.
FDP_ACC.1/GP-ELFU	This SFR is addressed by GP.ELFU.
FDP_ACF.1/GP-ELFU	This SFR is addressed by GP.ELFU.
FDP_ROL.1/GP-ELFU	This SFR is addressed by GP.ELFU.
FMT_MSA.1/GP-ELFU	This SFR is addressed by GP.ELFU.
FMT_MSA.3/GP-ELFU	This SFR is addressed by GP.ELFU.
FMT_SMF.1/GP-ELFU	This SFR is addressed by GP.ELFU.
FPT_FLS.1/GP-ELFU	This SFR is addressed by GP.ELFU.
FDP_ACC.1/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FDP_ACF.1/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FMT_MSA.3/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FMT_SMR.1/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FMT_SMF.1/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FIA_ATD.1/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FTP_TRP.1/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FCS_COP.1/OS-UPDATE-DEC	This SFR is addressed by GP.OS-UPDATE.
FCS_COP.1/OS-UPDATE-VER	This SFR is addressed by GP.OS-UPDATE.
FPT_FLS.1/OS-UPDATE	This SFR is addressed by GP.OS-UPDATE.
FDP_ACC.2/FIREWALL	This SFR is covered by JCS.Firewall.
FDP_ACF.1/FIREWALL	This SFR is covered by JCS.Firewall.
FDP_IFC.1/JCVM	This SFR is covered by JCS.Firewall and JCS.APDUBuffer controlling unauthorized access or invalid storage of reference.
FDP_IFF.1/JCVM	This SFR is covered by JCS.Firewall.
FDP_RIP.1/OBJECTS	This SFR is covered by JCS.OutOfLifeDataUndisclosure (to avoid access to data prior erase) and JCS.EraseResidualData (to erase data).
FMT_MSA.1/JCRE	This SFR is covered by JCS.RunTimeExecution covering context switch and application selection.
FMT_MSA.1/JCVM	This SFR is covered by JCS.ByteCodeExecution requiring context switch for specific code execution and JCS.RunTimeExecution

Security Functional Requirement	Coverage by TSS Security Function(s)
	covering context switch and modification of the Currently Active Context according to given rules.
FMT_MSA.2/FIREWALL_JCVM	This SFR is addressed by JCS.RunTimeExecution covering object sharing.
FMT_MSA.3/FIREWALL	This SFR is addressed by JCS.RunTimeExecution covering object sharing.
FMT_MSA.3/JCVM	This SFR is addressed by JCS.RunTimeExecution covering object sharing.
FMT_SMF.1	This SFR is addressed by JCS.RunTimeExecution covering context management and instance registration.
FMT_SMR.1	This SFR is addressed by JCS.RunTimeExecution covering JCVM and JCRE roles.
FCS_CKM.1/TDES	This SFR is addressed by JCS.KeyManagement covering key generation.
FCS_CKM.1/AES	This SFR is addressed by JCS.KeyManagement covering key generation.
FCS_CKM.1/RSA	This SFR is addressed by JCS.KeyManagement covering key generation.
FCS_CKM.1/ECDSA	This SFR is addressed by JCS.KeyManagement covering key generation.
FCS_CKM.1/HMAC	This SFR is addressed by JCS.KeyManagement covering key generation.
FCS_CKM.4	This SFR is addressed by JCS.KeyManagement covering key deletion.
FCS_COP.1/TDES_CIPHER	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/TDES_MAC	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/AES_CIPHER	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/AES_MAC	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/RSA_SIGN	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/RSA_CIPHER	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/ECDSA_SIGN	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/ECDH	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/DH	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/Hash	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/HMAC	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_COP.1/CRC	This SFR is covered by JCS.CryptoAPI dealing with the cryptographic services provided to applets through the Javacard API.
FCS_RNG.1	This SFR is covered by JCS.CryptoAPI providing AIS31 DRG.4 random number generation to applets.
FDP_RIP.1/ABORT	This SFR is addressed by JCS.EraseResidualData covering data erasure.
FDP_RIP.1/APDU	This SFR is addressed by JCS.EraseResidualData covering data erasure.
FDP_RIP.1/GlobalArray	This SFR is addressed by JCS.EraseResidualData covering data erasure.
FDP_RIP.1/bArray	This SFR is addressed by JCS.OutOfLifeDataUndisclosure and JCS.EraseResidualData covering data erasure.
FDP_RIP.1/KEYS	This SFR is addressed by JCS.EraseResidualData covering data erasure.
FDP_RIP.1/TRANSIENT	This SFR is covered by JCS.OutOfLifeDataUndisclosure managing the access control to transient object to be erased prior the erasure of the content in memory.
FDP_ROL.1/FIREWALL	This SFR is addressed by JCS.RunTimeExecution covering

Security Functional Requirement	Coverage by TSS Security Function(s)
	transaction rollback during specific operations.
FAU_ARP.1	This SFR is addressed by JCS.RunTimeExecution, JCS.Exception, JCS.Firewall, and OS.MemoryManagement covering exception handling with different specific operations.
FDP_SDI.2/DATA	This SFR is addressed by JCS.OwnerPIN, JCS.KeyManagement, OS.Atomicity and OS.MemoryManagement covering integrity handling with specific operations.
FPR_UNO.1	This SFR is addressed by JCS.OwnerPIN, JCS.KeyManagement, JCS.CryptoAPI and OS.MemoryManagement covering data handling with specific operations avoiding observation.
FPT_FLS.1/JCS	This SFR is covered by JCS.Exception, JCS.ByteCodeExecution, JCS.RunTimeExecution, and OS.Atomicity preserving a secure state when unexpected events occur during specific operations.
FPT_TDC.1	This SFR is covered by JCS.Package enforcing export check, CAP file translation and link specific operations.
FIA_ATD.1/AID	This SFR is covered by JCS.RunTimeExecution and GP.GPRegistry controlling applet registration and uninstallation.
FIA_UID.2/AID	This SFR is covered by GP.GPRegistry and JCS.RunTimeExecution managing user identity (package AID) during applet selection and identify associated context provided.
FIA_USB.1/AID	This SFR is covered by GP.GPRegistry and JCS.RunTimeExecution managing registration of each applet and associated package during its installation with its AID.
FMT_MTD.1/JCRE	This SFR is covered by JCS.RunTimeExecution offering services for applet registration and uninstallation managing associated access rights.
FMT_MTD.3/JCRE	This SFR is fully covered by JCS.RunTimeExecution managing presence and legacy of AID with ISO rules.
FDP_ACC.2/ADEL	This SFR is covered by GP.CardContentManagement, GP.GPRegistry and JCS.RunTimeExecution checking rules for applet instance uninstallation and deletion dependency rules.
FDP_ACF.1/ADEL	This SFR is covered by GP.CardContentManagement, GP.GPRegistry and JCS.RunTimeExecution checking rules for applet instance uninstallation and deletion dependency rules.
FDP_RIP.1/ADEL	This SFR is covered by GP.CardContentManagement and JCS.OutOfLifeDataUndisclosure by checking operations to avoid access to freed resources prior to its reuse.
FMT_MSA.1/ADEL	This SFR is covered by GP.GPRegistry, GP.CardContentManagement and JCS.RunTimeExecution responsible of checking rules concerning applet attributes, implicit and explicit selection rules prior to authorize deletion operation.
FMT_MSA.3/ADEL	This SFR is covered by JCS.RunTimeExecution and GP.CardContentManagement dealing with Security Attributes initialization, providing secure, restrictive default values for the security attributes of subject and objects involved in applet deletion.
FMT_SMF.1/ADEL	This SFR is covered by GP.CardContentManagement, GP.SecurityDomain and JCS.RunTimeExecution.
FMT_SMR.1/ADEL	This SFR is covered by GP.SecurityDomain maintaining the ISD and SDD roles responsible of applet deletion. This SFR is also covered by JCS.RunTimeExecution maintaining the JCRE role for applet uninstallation
FPT_FLS.1/ADEL	This SFR is covered by GP.GPRegistry, JCS.RunTimeExecution and OS.Atomicity preserving a secure state when unexpected events occur during package or instance deletion, managing the transaction part of the deletion operation by either rolling back, or completing it.
FDP_RIP.1/ODEL	This SFR is covered by JCS.EraseResidualData and OS.MemoryManagement ensuring that the content of deleted objects is erased upon the deletion and by JCS.OutOfLifeDataUndisclosure making unavailable for disclosure upon further reallocation of the freed space.
FPT_FLS.1/ODEL	This SFR is covered by JCS.RunTimeExecution and OS.MemoryManagement performing memory management to release no more used memory on unreferenced objects and preserves a secure state when unexpected events occur during object deletion.

Security Functional Requirement	Coverage by TSS Security Function(s)
FPT_RCV.3/OS	This SFR is covered by OS.Atomicity.
FPT_RCV.4/OS	This SFR is covered by OS.MemoryManagement.
FDP_SDI.2/ARRAY	This SFR is covered by JCS.SensitiveArray.
FDP_SDI.2/RESULT	This SFR is covered by JCS.SensitiveResult.
FDP_SDI.2/MONOTONIC_COUNTER	This SFR is covered by JCS.MonotonicCounters.
FDP_SDI.2/CRT_MNGT	This SFR is covered by JCS.CryptoCertManagement.
FCS_COP.1/CRT_MNGT	This SFR is covered by JCS.CryptoCertManagement.
FCS_CKM.5/KDF	This SFR is covered by JCS.KDF.
FPT_STM.1/SYS_TIME	This SFR is covered by JCS.SystemTime.

END OF DOCUMENT