SkySIM CX Hercules v2.0 Security Target Lite

Version 1.0 / 2014-12-18

Author	Giesecke & Devrient
File:	ASE_ST-Lite_SkySIM_CX_Hercules_Basic Configuration.doc

Giesecke & Devrient GmbH
Prinzregentenstraße 159
Postfach 80 07 29
D-81607 München

Contents

Document History

1.0	2014-12-18	wallhaek	Initial version based on SkySIM CX Hercules v2.0 Security Target Version 1.7 / 2014-12-17

1 ST Introduction

1.1 ST Reference

Title: Security Target Lite SkySIM CX Hercules v2.0

Reference: Security Target Lite SkySIM CX Hercules v2.0, Version 1.0 / 2014-12-18

Author: Giesecke & Devrient

TOE name: SkySIM CX Hercules v2.0

TOE guidance: [AGD_PRE], [AGD_OPE]

TOE hardware:

ST33G1M2 Rev. F (Certificate: ANSSI-CC-2014/46)

This TOE was evaluated against Common Criteria Version 3.1, Revision 4.

Assurance Level: EAL4-augmented with the following assurance components:

ALC_DVS.2 and AVA_VAN.5.

1.2 ST Overview

This Security Target describes the Target of Evaluation SkySIM CX Hercules v2.0.

Chapter 1 is an introduction to this document.

Chapter 2 includes the conformance claims.

Chapter 4 provides a discussion of the security problems related to the TOE. This section also defines the set of threats to be addressed either by the countermeasures implemented in the TOE hardware, the TOE software, or through environmental controls.

Chapter 5 defines the security objectives for both the TOE and the operational environment and the security objective rational to explicitly demonstrate that the information technology security objectives satisfy the policies and threats. Arguments are provided for the coverage of each policy and threat.

Chapter 6 contains the security functional requirements and assurance requirements derived from the Common Criteria [CC P1], Part 2 [CC P2] and Part 3 [CC P3], which must be satisfied and the security functional requirements rational. The section then explains how the set of requirements are complete relative to the objectives, and that each security objective is addressed by one or more component requirements. Arguments are provided for the coverage of each objective.

Chapter 8 contains the TOE Summary Specification.

Chapter 10 provides information on used acronyms and glossary and the used references.

1.3 Typographic Conventions

Underlined text is used to highlight changes in the Security Objectives and SFRs, including the rationale chapters, performed by the ST author.

Application Notes that introduced by the ST author have the addition '(ST author)'.

Bold text used in the Chapter 6 highlights assignments and selections for SFRs defined in the PP.

1.4 TOE Overview

The Target of Evaluation (TOE) is a (U)SIM Java Card platform embedded in a (U)SIM card intended to be plugged in a mobile phone or other mobile device. The TOE consists of the related embedded software and firmware in combination with the underlying hardware.

The TOE depends on the secure operational environment that includes the off-card bytecode verifier.

1.4.1 TOE Type

SkySIM CX Hercules v2.0 is the Target of Evaluation (TOE) of this Security Target. The TOE is composed of the following:

- A Java Card System defined in the [PPJCSv3.0], including all the native code, which manages and executes applications called applets. It provides APIs for developing applets in accordance with the Java Card™ specification, [JCAPI].
- GlobalPlatform (GP) packages providing a common interface to communicate with a smart card and manage applications in a secure way according to the [GP] specifications.
- (U)SIM APIs for interacting with (U)SIM applications, according to [TS131 130] specifications.
- The SCP (Smart Card Platform) comprehends the IC (Integrated Circuit) and the OS (Operating System).

The TOE boundaries are indicated in Figure 1 by the green dashed line. The TOE architecture is based on the generic architecture of the Basic TOE illustrated in [PP(U)SIM], Figure 1. The number bullets in Figure 1 and Figure 2 below demonstrate the correspondence of the TOE and non-TOE elements in the two PPs, to which this ST claims conformance.

	TOE of the PP	SkySIM CX Hercules TOE
①	The Smart Card Platform (SCP) is a combination of the Securtiy Integrated Circuit (IC) and the native Operating System (OS).	ST33G1M2 Rev. F and SkySIM CX Hercules OS
②	Java Card System (JCRE, JCVM, JCAPI)	Java Card Platform 3.0.4 classic implementation
③	Additional native code, proprietary applications	Native Telecommunication Application
④	Applets	The TOE does not include applets.

Table 1 Correspondance of TOE building blocks in PP and ST

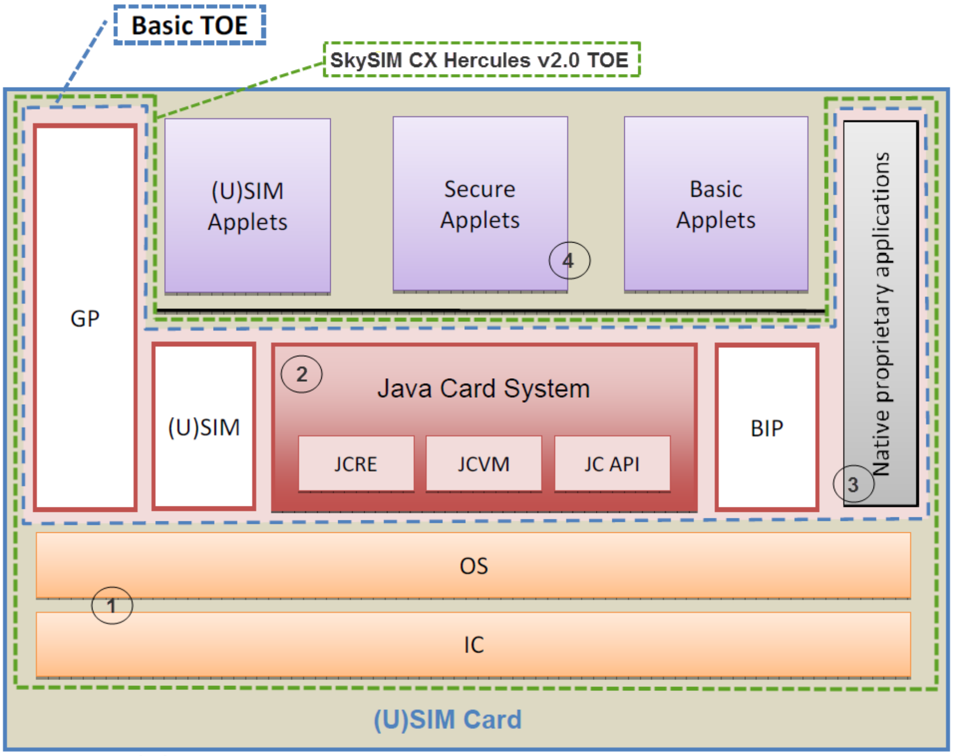


Figure 1 (U)SIM PP Basic TOE and the SkySIM CX Hercules v2.0 TOE

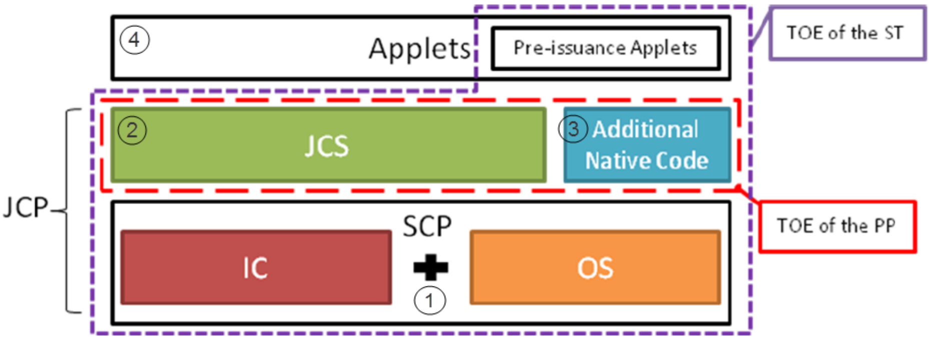


Figure 2 Java Card Platform TOE boundaries defined in JCS PP, Figure 1

1.4.2 TOE Description

SkySIM CX Hercules v2.0 is a Basic configuration TOE according to the definition in [PP(U)SIM], Section 1.3.2.

The TOE supports (U)SIM functionality and associated APIs as described in [PP(U)SIM], Section 1.3.2.1.

The TOE includes the Telecom Environment with Network Authentication Service, Over-The-Air and BIP communication, File System management, and Toolkit services. BIP support is as described in [PP(U)SIM], Section 1.3.2.2. BIP does not offer any security function.

The TOE includes the Java Card Virtual Machine (Java Card VM), the Java Card Runtime Environment (Java Card RE), and the Java Card Application Programming Interface (Java Card API), compliant with the version of the Java Card Platform given in the specifications [JCVM], [JCRE], and [JCAPI]. The [PP(U)SIM] details it in Section 1.3.2.3.

The TOE is compliant with the GlobalPlatform specification [GP], which defines a set of APIs and technologies to perform the operations involved in the management of applications hosted by the card in a secure way. The [PP(U)SIM] details it in Section 1.3.2.4.

The following GP functionality, at least, is present within the TOE:

- Card and Application management according to [GP]:
    - Content loading, installation, making selectable, removal, extradition
    - DAP verification and mandated DAP verification
    - Security Domain and application privileges
- Secure Channel protocols (SCP02, SCP80)
- Support for contactless cards (ATQ, different implicit selection on different interfaces and channels)
- Post-issuance personalization of Security Domain according to [GP UICC]
- Application personalization according to [GP UICC]

1.4.3 TOE Usage

The SIM, defined in the 3GPP standards as the Subscriber Identity Module, is a removable module within GSM mobile equipment that contains the International Mobile Subscriber Identity (IMSI) which unambiguously identifies a subscriber. When the SIM is placed in mobile equipment, users can register onto the GSM network. The primary function of the SIM is consequently used to authenticate the validity of a terminal when accessing the network. It also provides a means to authenticate the end user and may store other subscriber-related information or applications such as SIM Toolkit applications as specified in [TS102 223] and [TS131 111].

The SIM is the MNO’s property, and stores MNO’s specific information. The USIM defined in the 3GPP standards as the Universal Subscriber Identity Module is an evolution of the SIM developed to ensure compliance within UMTS networks (also called 3G). This new generation of SIM especially includes improvements of mutual authentication mechanisms.

In the following SIM and USIM are considered in the same way regarding security.

1.4.4 TOE Life Cycle

The TOE life cycle follows the description in the [PP(U)SIM], Section 1.3.5. The TOE life cycle phases are detailed in Figure 3. We refer to [PP0035] for a thorough description of Phases 1 to 7:

- Phases 1 and 2 compose the product development: Embedded Software (IC Dedicated Software, OS, Java Card System, (U)SIM applet, other platform components such as Card Manager, Applets) and IC development.
- Phase 3 and 4 correspond to IC manufacturing and packaging, respectively. Some IC pre-personalisation steps may occur in Phase 3.
- Phase 5 concerns the embedding of software components within the IC.
- Phase 6 is dedicated to the product personalisation prior final use.
- Phase 7 is the product operational phase.

The (U)SIM platform life cycle is composed of four stages:

- Development,
- Storage, pre-personalization and test,
- Personalization and test,
- Final usage.

These stages map to the TOE life cycle phases as shown inFigure 3.

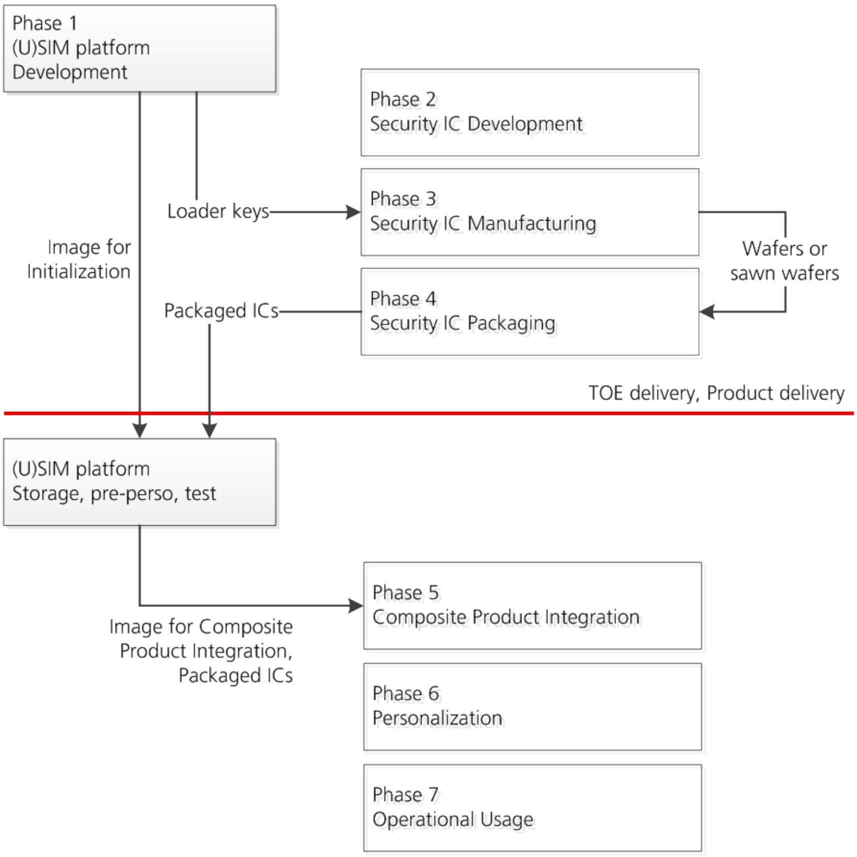


Figure 3 TOE Life Cycle

The (U)SIM platform Development is performed during Phase 1. This includes Java Card System (JCS) and (U)SIM conception, design, implementation, testing and documentation. The development fulfils requirements of the final product, including conformance to Java Card Specifications, and recommendations of the SCP user guidance. The development occurs in a controlled environment that avoids disclosure of source code, data and any critical documentation and that guarantees the integrity of these elements. The evaluation includes the (U)SIM platform development environment, in particular those locations where the TOE is accessible for installation or testing.

TOE delivery and product delivery fall together at the end of Phase 4. There is no delivery of the (U)SIM platform to the Security IC Manufacturing site in Phase 3. The image for initialization loaded on a chip represents the TOE. An encrypted image representing the (U)SIM Java Card System and the packaged ICs are delivered to the Composite Product Integration site. The evaluation includes the delivery process. Delivery and acceptance procedures guarantee the authenticity, the confidentiality and integrity of the exchanged pieces. (U)SIM platform delivery involves encrypted signed sending with previous exchange of public keys. The Composite Product Integration environment protects the confidentiality and integrity of the Security IC Embedded Software and any related material.

The (U)SIM platform is personalized in Phase 6.

The delivery processes at phases 2 to 4 and the sites involved are covered by the evaluation of the IC platform addressed in [ST33-ST].

1.4.5 Actors of the TOE

One of the characteristics of the (U)SIM Java Card platforms is that several entities are represented inside these platforms:

- The **Mobile Network Operator** (MNO or mobile operator), issuer of the (U)SIM Java Card platform and proprietary of the TOE. The TOE guarantees that the issuer, once authenticated, can manage the loading, instantiation and deletion of applications.
- The **Application Provider** (AP), entity or institution responsible for the applications and their associated services. It is a financial institution (a bank), a transport operator or a third party operator.
- The **Controlling Authority** (CA), entity independent from the MNO represented on the (U)SIM card and responsible for securing the keys creation and personalization of the Application Provider Security Domain (APSD) (Push and Pull personalization model of [GP UICC]). Note that in [GP2.2], the term "Controlling Authority" is also used for the authority managing Mandated DAP verification. This role, which is not necessarily assigned to the Controlling Authority in [GP UICC], is endorsed in this document by the Verification Authority, as follows.
- The **Verification Authority** (VA), trusted third party represented on the (U)SIM card, acting on behalf of the MNO and responsible for the verification of applications signatures (Mandated DAP) during the loading process. These applications shall be validated for the Basic ones or certified for the Secure ones.

1.4.6 TOE Security Features

Secure or Basic applets can be loaded and instantiated onto the TOE either before card issuance or over-the-air (OTA) in post-issuance through the mobile network, without physical manipulation of the TOE and in a connected environment. Besides these, other administrative operations can also be done OTA.

The main security feature of the TOE is the correct and secure execution of sensitive applications, in a connected environment and with the presence on the TOE of Basic (non-certified) applications.

1.4.6.1 Security Services to Applications

The TOE offers to applications a panel of security services in order to protect application data and assets:

- Confidentiality and integrity of cryptographic keys and associated operations. Cryptographic operations are protected, including protection against observation or perturbation attacks. Confidentiality and integrity of cryptographic keys and application data are guaranteed at all time during execution of cryptographic operations.
- Confidentiality and integrity of authentication data. Authentication data are protected, including protection against observation or perturbation attacks. Confidentiality and integrity of authentication data and application data are guaranteed at all time during execution of authentication operations.
- Confidentiality and integrity of application data among applications. Applications belonging to different contexts are isolated from each other. Application data are not accessible by a normal or abnormal execution of another Basic or Secure application.
- Application code execution integrity. The Java Card VM and the “applications isolation” property guarantee that the application code is operating as specified in absence of perturbations. In case of perturbation, this TOE security feature must also be valid.

1.4.6.2 Application Management

The TOE offers additional security services for applications management, relying on the GlobalPlatform framework:

- The MNO as Card issuer is initially the only entity authorized to manage applications (loading, instantiation, deletion) through a secure communication channel with the card, based on SMS, BIP. However, the MNO can grant these privileges to the AP through the delegated management functionality of GP.
- Before loading, all applications are verified by a validation laboratory for the Basic applications, or by an ITSEF for the secure applications. All loaded applications are associated at load time to a Verification Authority (VA) signature (Mandated DAP) that is verified on card by the on-card representative of the VA prior to the completion of the application loading operation and prior to the instantiation of any applet defined in the loaded application.
- Before loading, application code can be encrypted (Encrypted Load File) using a key owned by the Controlling Authority Security Domain so as to ensure its confidentiality [GP CCM]. The application code will later be decrypted once extradited to its target APSD.
- Application Providers personalize their applications and Security Domains (APSD) in a confidential manner. Application Providers have Security Domain keysets enabling them to be authenticated to the corresponding Security Domain and to establish a trusted channel between the TOE and an external trusted device. The

Certification Authority is responsible for securing the Security Domain keysets creation and personalization of the Application Provider Security Domain (APSD) [GP CCM]). These keysets are not known by the Card issuer.

- A Security Domain with a Receipt Generation privilege is able to generate a receipt acting as a proof of the completion of the requested card content management operations initiated by the SD. This covers the following operations: loading, extradition, installing, removing and updating the GlobalPlatform registry operations (see Section 9.1.3.6 in [GP]).

Basic and Secure applets (defined in [PP(U)SIM], Section 1.3.8.4 and 1.3.8.5) are loaded in different Java Card packages.

1.4.7 Non-TOE HW/SW/FW Available to the TOE

The following non-TOE HW/SW/FW defined in [PP(U)SIM] and [PPJCSv3.0] are non-TOE HW/SW/FW in the scope of this Security Target:

- Off-card Bytecode Verifier, [PPJCSv3.0], Section 2.3.1
- Mobile Terminals, [PP(U)SIM], Section 1.3.8.3
- Basic Applets, [PP(U)SIM], Section 1.3.8.4
- Secure Applets loaded post-evaluation, [PP(U)SIM], Section 1.3.8.5
- Terminal, remote servers and trusted IT products, [PP(U)SIM], Section 1.3.8.6
- Contactless Applications, the CRS Application and Contactless Services according to [GP CS]
- Customer-specific plugins and APIs
- Authentication algorithm plugins and APIs

2 Conformance Claim

2.1 CC Conformance Claim

This Security Target claims conformance to [CC P1], [CC P2] (extended) and [CC P3] (conformant).

2.2 PP Claim

This Security Target claims **demonstrable** conformance to the (U)SIM Java Card™ Platform Protection Profile- Basic configuration, [PP(U)SIM].

The TOE is conformant to the Common Criteria Protection Profile Java Card™ System Open Configuration, Version 3.0, May 2012, [PPJCSv3.0].

2.3 Package Claim

This Security Target is conformant to the assurance package EAL4 augmented with ALC_DVS.2 and AVA_VAN.5 as defined in [CC P3].

2.4 Conformance Claims Rationale

The differences between this Security Target and the Protection Profiles [PP(U)SIM] and [PPJCSv3.0] are described here to justify the claimed conformance to the PP.

The TOE is conformant to [PPJCSv3.0], which references Java Card Platform (JCAPI, JCRE, JCVM) specification version 3.0.1. The [PP(U)SIM] references Java Card Platform specification version 2.2.2. According to the Release Notes for the Java Card 3 Platform Version 3.0.4, September 2011, the TOE is still conformant to the PPs and to the specifications [JCAPI], [JCRE], and [JCVM], because:

- The Java Card Runtime Environment Specification Version 3.0.4 mirrors those REs found in previous releases of the Java Card platform, including v3.0.1 and v2.2.2.
- Java Card API Specification Version 3.0.4 mirrors those APIs found in previous releases of the Java Card platform, including v3.0.1 and v2.2.2.
- Java Card Virtual Machine Specification Version 3.0.4 mirrors those VMs found in previous releases of the Java Card platform, including v3.0.1 and v2.2.2.

The assumption A.DELETION is replaced by the threat T.SECURE_DELETION.

The SCP is part of the TOE of this ST. Therefore objectives for the operational environment OE.SCP.RECOVERY, OE.SCP.SUPPORT, and OE.SCP.IC have been changed to objectives for the TOE O.SCP.RECOVERY, O.SCP.SUPPORT, and O.SCP.IC.

The following SFRs have been introduced: FPT_PHP.3, FCS_RNG.1.

The following SFRs from JCS PP are not included, since they are covered by the SFRs that were included from the (U)SIM PP (see Table 15): FDP_IFC.2/CM, FDP_IFF.1/CM, FDP_UIT.1/CM, FIA_UID.1/CM, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMF.1/CM, FMT_SMR.1/CM, FTP_ITC.1/CM.

The following InstG SFRs from JCS PP of are not included, because the are covered by the card content management SFRs from the (U)SIM PP: FDP_ITC.2/Installer is covered by FDP_ITC.2/CCM, FPT_FLS.1.1/Installer is covered by FPT_FLS.1.1/CCM.

The exclusion of the above listed SFRs does not compromise the demonstrable conformance to the PPs, since the SFRs introduced through the (U)SIM PP either extend the SFRs originating JCS PP or address exactly the same functionality. Exclusion of the JCS PP SFRs is therefore considered as removal of duplicates.

3 Security Aspects

This Security Target includes all Security Aspects as defined in the JCS PP. The list below includes all Security Aspects without repeating their definition. For a detailed definition of the security aspects, see [PPJCSv3.0], Chapter 4.

#.CONFID-APPLI-DATA

#.CONFID-JCS-CODE

#.CONFID-JCS-DATA

#.INTEG-APPLI-CODE

#.INTEG-APPLI-DATA

#.INTEG-JCS-CODE

#.INTEG-JCS-DATA

#.EXE-APPLI-CODE

#.EXE-JCS-CODE

#.FIREWALL

#.NATIVE

#.VERIFICATION

#.CARD-MANAGEMENT

#.INSTALL

#.SID

#.OBJ-DELETION

#.DELETION

#.ALARM

#.OPERATE

#.RESOURCES

#.CIPHER

#.KEY-MNGT

#.PIN-MNGT

#.SCP

#.TRANSACTION

4 Security Problem Definition

All SFRs of [PP(U)SIM] and all SFRs of [PPJCSv3.0] are relevant to the TOE of this ST. All assets, threats, OSPs, assumptions, security objectives of the JCS PP are included in the (U)SIM PP. This Security Target does not repeat the definition of assets, assumptions, threats, and organisational security policies from the Protection Profiles.

The tables included in subsections below summarize the refinements, additions, and equivalences between the JCS PP, the (U)SIM PP and this ST. In order to precisely state the relationship between assets, assumptions, threats, organisational security policies, objectives and SFRs from the Protection Profiles and the SPD of this ST the following notation is used:

D: Definition of the asset, threat, OSP, assumption, objective, or SFR in the JCS PP.

A: Additional assets, subjects, threats, OSPs, assumptions, objectives or SFRs specified in the (U)SIM PP that are independent from the JCS PP SPD and do not affect it.

R: Refinement made in (U)SIM PP or in this ST. (U)SIM PP assets, threats, OSPs, assumptions or objectives are more restrictive than the ones required in JCS PP SPD.

E: Equivalent threats, OSPs, objectives or assumptions in (U)SIM PP are the same as in JCS PP but apply to additional assets defined in (U)SIM PP.

x: The asset, threat, OSP, objective, assumption, or SFR is included and corresponds to its original definition in the JCS PP or in the (U)SIM PP.

- : The asset, threat, OSP, objective, OE, assumption, or SFR is not included in the PP or in this ST.

4.1 Assets

Table 2 indicates the assets inclded in this Security Target. All assets defined in the Protection Profiles ([PPJCSv3.0], Section 5.1, [PP(U)SIM], Section 3.1) are relevant for the TOE of this Security Target. The definition of assets is not repeated here.

Asset	JCS PP	(U)SIM PP Basic TOE	ST TOE
TSF Data			
D.API_DATA	D	x	x
D.CRYPTO	D	x	x
D.JCS_CODE	D	x	x
D.JCS_DATA	D	x	x
D.SEC_DATA	D	x	x
D.GP_CODE	-	A	x
D.CARD_MNGT_DATA	-	A	x
User Data			
D.APP_CODE	D	x	x
D.APP_C_DATA	D	x	x
D.APP_I_DATA	D	x	x
D.PIN	D	x	x
D.APP_KEYS	D	R	x
D.APSD_KEYS	-	R	x
D.CASD_KEYS	-	R	x
D.ISD_KEYS	-	R	x
D.VASD_KEYS	-	R	x
D.(U)SIM_CODE	-	R	x
D.(U)SIM_DATA	-	R	x

Table 2 Assets

R: The assets D.APSD_KEYS, D.CASD_KEYS, D.ISD_KEYS, D.VASD_KEYS in the (U)SIM PP are refinements of the asset D.APP_KEYS in JCS PP. The assets D.(U)SIM_CODE, D.(U)SIM_DATA in the (U)SIM PP are refinements of the asset D.JCS_CODE and D.JCS_DATA in JCS PP.

4.2 Subjects

Table 3 indicates the subjects included in this Security Target. All subjects defined in the Protection Profiles ([PPJCSv3.0], Section 7.1, [PP(U)SIM], Section 3.2) are relevant for the TOE of this Security Target. The definition of the subjects is not repeated here.

Subject	JCS PP	(U)SIM PP Basic TOE	ST TOE
S.ADEL	D	x	x
S.APPLET	D	x	x
S.BCV	D	x	x
S.CAD	D	x	x
S.INSTALLER	D	x	x
S.JCRE	D	x	x
S.JCVM	D	x	x
S.LOCAL	D	x	x
S.MEMBER	D	x	x
S.PACKAGE	D	x	x
S.SD	-	A	x

Table 3 Subjects

The subject S.INSTALLER is represented by the Issuer Security Domain on the card. S.ADEL is represented by a Security Domain with the ability to delete Applications and Executable Load Files. The subject S.SD includes the subjects S.INSTALLER and S.ADEL.

4.3 Threats

Table 4 indicates the threats to be averted by the TOE independently or in collaboration with its IT environment. These threats result from the TOE method of use in the operational environment and the assets stored in or protected by the TOE.

All threats included in this Security Target are as defined in the Protection Profiles ([PPJCSv3.0], Section 6.3.1, [PP(U)SIM], Section 3.3).

Threat	JCS PP	(U)SIM PP Basic TOE	ST TOE
T.CONFID-APPLI-DATA	D	x	x
T.UNAUTHORIZED_ACCESS	-	R1	x
T.CONFID-JCS-CODE	D	x	x
T.CONFID-JCS-DATA	D	x	x
T.INTEG-APPLI-CODE	D	x	x
T.INTEG-APPLI-CODE.LOAD	D	x	x
T.INTEG-APPLI-DATA	D	x	x
T.INTEG-USER-DATA	-	R2	x
T.INTEG-APPLI-DATA.LOAD	D	x	x
T.INTEG-JCS-CODE	D	x	x
T.INTEG-JCS-DATA	D	x	x
T.SID.1	D	x	x
T.SID.2	D	x	x
T.EXE-CODE.1	D	x	x
T.EXE-CODE.2	D	x	x
T.NATIVE	D	x	x
T.RESOURCES	D	x	x
T.DELETION	D	x	x
T.INSTALL	D	x	x
T.UNAUTHORIZED_CARD_MNGT	-	R3	x
T.LIFE_CYCLE	-	A	x
T.OBJ-DELETION	D	x	x
T.PHYSICAL	D	E2	x
T.COM_EXPLOIT	-	A	x
T.SECURE-DELETION	-	-	A

Table 4 Threats

R1: T.UNAUTHORIZED_ACCESS is a refinement of JCS PP threat T.CONFID-APPLI-DATA to Shareable Objects.

R2: T.INTEG-USER-DATA is a refinement of JCS PP threat T.INTEG-APPLI-DATA. Applies to (U)SIM PP assets.

R3: T.UNAUTHORIZED_CARD_MNGT is a refinement of JCS PP threat T.INSTALL.

A: T.SECURE_DELETION replaces A.DELETION.

E2: T.PHYSICAL in (U)SIM PP is equivalent to T.PHYSICAL of JCS PP. It applies to the new assets introduced in the (U)SIM PP. The definition of T.PHYSICAL in this ST is identical to that in (U)SIM PP.

In the rational of the [PP(U)SIM], Sections 4.3.1.1, T.PHYSICAL was covered by the objectives OE.SCP.SUPPORT and O.INTEG-APPLI-CODE. The inclusion of the SCP into the TOE of this composite evaluation requires that T.PHYSICAL is covered by TOE objectives. This is justified in Section 5.3.

T.SECURE_DELETION

The attacker exploits security holes that are introduced through the deletion of an installed applet in the form of broken references to garbage collected code or data or alter integrity or confidentiality of remaining applets. That could be used to maliciously bypass the TSF and jeopardize the TOE (or its assets) in case of failure (such as power shortage).

Directly threated assest(s): D.APP_I_DATA, D.SEC_DATA, D_APP_KEY, D.PIN and D.CRYPTO.

4.4 Organisational Security Policies

The TOE complies with all Organizational Security Policies (OSP) defined in the Protection Profiles ([PPJCSv3.0], Section 6.3.2, [PP(U)SIM], Sections 3.4) as indicated in Table 5.

Organizational Security Policy	JCS PP	(U)SIM PP Basic TOE	ST TOE
OSP.VERIFICATION	D	R	x
OSP.SECURE-APPS-CERTIFICATION	-	R	x
OSP.BASIC-APPS-VALIDATION	-	R	x
OSP.SHARE-CONTROL	-	R	x
OSP.AID-MANAGEMENT	-	R	x
OSP.OTA-LOADING	-	A	x
OSP.OTA-SERVERS	-	A	x
OSP.APSD-KEYS	-	A	x
OSP.OPERATOR-KEYS	-	A	x
OSP.KEY-GENERATION	-	A	x
OSP.CASD-KEYS	-	A	x
OSP.VASD-KEYS	-	A	x
OSP.KEY-CHANGE	-	A	x
OSP.SECURITY-DOMAINS	-	A	x
OSP.QUOTAS	-	A	x

Table 5 Organizational Security Policies

R: OSP.SECURE-APPS-CERTIFICATION, OSP.BASIC-APPS-VALIDATION, OSP.SHARE-CONTROL, and OSP.AID-MANAGEMENT are refinements of OSP.VERIFICATION in the JCS PP.

4.5 Assumptions

The assumptions describe the security aspects of the environment in which the TOE will be used or is intended to be used. The assumptions defined in the Protection Profiles ([PPJCSv3.0], Section 6.3.3, [PP(U)SIM], Sections 3.5) are included in this Security Target as indicated in Table 6.

All assumptions are relevant for the TOE except A.DELETION that is replaced by T.SECURE_DELETION.

Assumption	JCS PP	(U)SIM PP Basic TOE	ST TOE
A.APPLET	D	x	x
A.DELETION	D	x	-
A.VERIFICATION	D	x	x
A.MOBILE-OPERATOR	-	A	x
A.OTA-ADMIN	-	A	x
A.APPS-PROVIDER	-	A	x
A.VERIFICATION-AUTHORITY	-	A	x
A.CONTROLLING-AUTHORITY	-	A	x
A.KEY-ESCROW	-	A	x
A.PERSONALIZER	-	A	x
A.PRODUCTION	-	A	x

Table 6 Assumptions

5 Security Objectives

5.1 Security Objectives for the TOE

Table 7 indicates the security objectives for the TOE addressing the aspects of identified threats to be countered by the TOE and organizational security policies to be met by the TOE. All security objectives that are relevant for the TOE of this Security Target are as defined in the Protection Profiles ([PPJCSv3.0], Section 6.1, [PP(U)SIM], Section 4.1).

Objective	JCS PP	(U)SIM PP Basic TOE	ST TOE
O.SID	D	x	x
O.FIREWALL	D	x	x
O.GLOBAL-ARRAYS-CONFID	D	x	x
O.GLOBAL-ARRAYS-INTEG	D	x	x
O.NATIVE	D	x	x
O.OPERATE	D	x	x
O.REALLOCATION	D	x	x
O.RESOURCES	D	x	x
O.ALARM	D	x	x
O.CIPHER	D	x	x
O.KEY-MNGT	D	x	x
O.PIN-MNGT	D	x	x
O.REMOTE	D	x	-
O.TRANSACTION	D	x	x
O.OBJ-DELETION	D	x	x
O.DELETION	D	x	x
O.LOAD	D	x	x
O.INSTALL	D	x	x
O.CARD-MANAGEMENT	-	R1	x
O.DOMAIN-RIGHTS	-	A	x
O.APPLI-AUTH	-	R2	x
O.COMM_AUTH	-	A	x
O.COMM_INTEGRITY	-	A	x
O.COMM_CONFIDENTIALITY	-	A	x
O.SCP.IC	-	-	R3
O.SCP.RECOVERY	-	-	R3
O.SCP.SUPPORT	-	-	R3

Table 7 Security Objectives for the TOE

R1: O.CARD-MANAGEMENT is a refinement of the JCS PP objectives O.INSTALL, O.LOAD, O.DELETION and OE.CARD-MANAGEMENT.

R2: O.APPLI-AUTH in the (U)SIM PP is a refinement of the objective O.LOAD in JCS PP.

R3: The TOE of this composite evaluation was extended by the SCP; therefore objectives for the environment OE.SCP.IC, OE.SCP.RECOVERY and OE.SCP.SUPPORT have been changed to objectives for the TOE O.SCP.IC, O.SCP.RECOVERY, and O.SCP.SUPPORT (see Section 5.2).

A: O.DOMAIN-RIGHTS, O.COMM_AUTH, O.COMM_INTEGRITY, O.COMM_CONFIDENTIALITY in the (U)SIM PP are additions to the JCS PP for card content management environment.

The following sections define the security objectives to be achieved by the TOE.

5.1.1 IDENTIFICATION

O.SID

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

5.1.2 EXECUTION

O.FIREWALL

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

O.GLOBAL_ARRAYS_CONFID

The TOE shall ensure that the APDU buffer that is shared by all applications is always cleaned upon applet selection.

The TOE shall ensure that the global byte array used for the invocation of the install method of the selected applet is always cleaned after the return from the install method.

O.GLOBAL_ARRAYS_INTEG

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

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.

5.1.3 SERVICES

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.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. 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 counter's value is as sensitive as that of the PIN.

O.TRANSACTION

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

O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION and O.CIPHER are actually provided to applets in the form of Java Card APIs.

5.1.4 OBJECT DELETION

O.OBJ-DELETION

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

5.1.5 APPLET MANAGEMENT

O.DELETION

The TOE shall ensure that both applet and package deletion perform as expected. See #.DELETION for details.

O.LOAD

The TOE shall ensure that the loading of a package 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 package 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 packages sent to the card. He may also try to send one of its own applications as if it came from the card issuer. Thus, this objective is intended to ensure the integrity and authenticity of loaded CAP files.

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 package by the verification authority. If not performed during the loading process, this verification by the TOE shall occur during the install process.

5.1.6 Smart Card Platform

O.SCP.IC

The SCP shall possess IC security features against physical attacks.

This security objective refers to the point (7) of the security aspect #.SCP:

(7) It is required that the IC is designed in accordance with a well-defined set of policies and Standards, 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.

Application Note (ST author):

> O.SCP.IC covers also leakage attacks like DPA. The IC provides support against leakage attacks, which will be used by the TOE.

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 TOE shall support the following functionalities:

(1) It does not allow the TSFs to be bypassed or altered and does not allow access to other low-level functions than those made available by the packages of the API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System.

(2) It provides secure low-level cryptographic processing to the Java Card System, GlobalPlatform and SCWS frameworks (for SCWS TOE).

(3) 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.

(4) 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).

5.1.7 Card Management

O.CARD-MANAGEMENT

The TOE shall provide card management functionalities (loading, installation, extradition, deletion of applications and GP registry updates) in charge of the life cycle of the whole (U)SIM card and installed applications (applets).

The card manager, the application with specific rights responsible for the administration of the smart card, shall control the access to card management functions. It shall also implement the card issuer's policy on card management.

Application Note:

> The card manager will be tightly connected in practice with the rest of the TOE, which in return shall very likely rely on the card manager for the effective enforcement of some of its security functions.
>
> The mechanism used to ensure authentication of the TOE issuer, that manages the TOE, or of the Service Providers owning a Security Domain with card management privileges is a secure channel. This channel will be used afterwards to protect commands exchanged with the TOE in confidentiality and integrity.
>
> The platform guarantees that only the ISD or the Service Providers owning a Security Domain with the appropriate privilege (Delegated Management) can manage the applications on the card associated with its Security Domain. This is done accordingly with the card issuer's policy on card management.
>
> The actor performing the operation must beforehand authenticate with the Security Domain. In the case of Delegated Management, the card management command will be associated with an electronic signature (GlobalPlatform token) verified by the ISD before execution.

O.DOMAIN-RIGHTS

The Card issuer shall not get access or change personalized AP security domain keys which belong to the AP. Modification of a security domain keyset is restricted to the AP who owns the security domain.

Application Note:

> APs have a set of keys that allows them to establish a secure channel between them and the platform. These keys sets are not known by the TOE issuer. The security domain initial keys are changed before any operation on the SD (OE.KEY-CHANGE) through standard PUT KEY procedures (if the initial keys were kept by key escrow) or through one of the SD personalization mechanisms described in Section 4.3.3 of [GP UICC].

O.APPLI-AUTH

The card manager shall enforce the application security policies established by the card issuer by requiring application authentication during application loading on the card.

Application Note:

> Each application loaded onto the TOE has been signed by the VA. The VA will guarantee that the security policies established by the card issuer on applications are enforced. This authority is present on the TOE as a Security Domain whose role is to verify each signature at application loading.
>
> The platform provides important extra features about application management and especially loading:
>
> - Loaded applications are previously validated by an accredited laboratory for basic applications and certified by an accredited ITSEF for secure applications.
> - All loaded applications are associated to a DAP signature generated by a VA which is verified at loading by the third party representative present on the platform (Mandated DAP verification).

5.1.8 Communication

O.COMM_AUTH

The TOE shall authenticate the origin of the card management requests that the card receives, 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.

5.2 Security Objectives for the Operational Environment

Table 8 indicates the security objectives relevant for the operational environment of the TOE. The definition of all security objectives for the operational environment from the Protection Profiles ([PPJCSv3.0], Section 6.2, [PP(U)SIM], Section 4.2) is included. An exception is the definition of security objectives related to the SCP in [PPJCSv3.0]. The SCP is part of the TOE. Therefore, the security objectives for the operational environment have been declared as security objectives for the TOE and detailed in Section 5.1.

Objective for the Operational Environment	JCS PP	(U)SIM PP Basic TOE	ST TOE
OE.APPLET	D	x	x
OE.CARD-MANAGEMENT	D	R1	-
OE.SCP.IC	D	x	-
OE.SCP.RECOVERY	D	x	-
OE.SCP.SUPPORT	D	E	-
OE.VERIFICATION	D	R2	x
OE.CODE-EVIDENCE	D	x	x
OE.MOBILE-OPERATOR	-	A	x
OE.OTA-ADMIN	-	A	x
OE.APPS-PROVIDER	-	A	x
OE.VERIFICATION-AUTHORITY	-	A	x
OE.KEY-ESCROW	-	A	x
OE.PERSONALIZER	-	A	x
OE.CONTROLLING-AUTHORITY	-	A	x
OE.PRODUCTION	-	A	x
OE.SECURE-APPS-CERTIFICATION	-	R2	x
OE.BASIC-APPS-VALIDATION	-	R2	x
OE.AID-MANAGEMENT	-	R2	x
OE.OTA-LOADING	-	A	x
OE.OTA-SERVERS	-	A	x
OE.AP-KEYS	-	A	x
OE.OPERATOR-KEYS	-	A	x
OE.KEY-GENERATION	-	A	x
OE.CA-KEYS	-	A	x
OE.VA-KEYS	-	A	x
OE.KEY-CHANGE	-	A	x
OE.SECURITY-DOMAINS	-	A	x
OE.QUOTAS	-	A	x
OE.SHARE-CONTROL	-	R2	x

Table 8 Objectives for the Operational Environment

R1: OE.CARD_MANAGEMENT defined in the JCS PP is refined by the objective for the TOE O.CARD_MANAGEMENT in the (U)SIM PP.

R2: OE.SECURE-APPS-CERTIFICATION, OE.BASIC-APPS-VALIDATION, OE.AID-MANAGEMENT, OE.SHARE-CONTROL in the (U)SIM PP are refinements of OE.VERIFICATION in the JCS PP.

The objectives for the operational environment OE.SCP.RECOVERY, OE.SCP.IC in the JCS PP have been moved to objectives for the TOE in this ST, since the SCP defined in JCS PP became part of the TOE of this ST. OE.SCP-SUPPORT in the (U)SIM PP became part of the TOE in this ST.

E: OE.SCP-SUPPORT in the (U)SIM PP is equivalent to OE.SCP.SUPPORT defined in the JCS PP. It applies to SCWS and to GP. In this ST, this objective was moved from the environment to the TOE: O.SCP.SUPPORT (see Section 5.1.6).

5.2.1 OEs from JCS PP

OE.APPLET

No applet 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 in the 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 Security Target.

Application Note:

> For application code loaded post-issuance and verified off-card, the integrity and authenticity evidence is achieved by electronic signature of the application code, after code verification, by the actor who performed verification.

5.2.2 Actors

OE.MOBILE-OPERATOR

The mobile operator shall be a trusted actor responsible for the mobile network and the associated OTA servers.

OE.OTA-ADMIN

Administrators of the mobile operator OTA servers shall be trusted people. They shall be trained to use and administrate those servers. They have the means and the equipments to perform their tasks.

They must be aware of the sensitivity of the assets they manage and the responsibilities associated to the administration of OTA servers.

OE.APPS-PROVIDER

The AP shall be a trusted actor that provides basic or secure application. He must be responsible of his security domain keys.

OE.VERIFICATION-AUTHORITY

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

OE.KEY-ESCROW

The key escrow shall be a trusted actor in charge of the secure storage of the AP initial keys generated by the personalizer.

OE.PERSONALIZER

The personalizer shall be a trusted actor in charge of the personalization process. He must ensure the security of the keys it manages and loads into the card:

- Mobile operator keys including OTA keys (telecom keys either generated by the personalizer or by the mobile operator),
- 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 APSD keys creation and personalisation. He must be responsible for his security domain keys (CASD keys).

5.2.3 Secure places

OE.PRODUCTION

Production and personalization environment in phases 5 and 6 must be trusted and secure.

5.2.4 Policies

VALIDATION AND CERTIFICATION

OE.SECURE-APPS-CERTIFICATION

Secure applications must be evaluated and certified at a security level higher or equal than the one of the current Protection Profile.

OE.BASIC-APPS-VALIDATION

Basic applications must be analysed during the validation process in order to ensure that the rules for correct usage of the TOE are still enforced.

OE.AID-MANAGEMENT

The VA or the MNO 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.

LOADING

OE.OTA-LOADING

Application code, validated or certified depending on the application, is loaded "Over The Air" (OTA) onto (U)SIM Platform using OTA servers. This process should protect the confidentiality and the integrity of the loaded application code.

OE.OTA-SERVERS

The mobile operator must enforce a policy to ensure the security of the applications stored on its servers.

KEYS

OE.AP-KEYS

The SD keys personalizer, the AP and the key escrow must enforce a security policy on SD keys in order to secure their transmission.

OE.OPERATOR-KEYS

The security of the mobile operator keys must be ensured in the environment of the TOE.

OE.KEY-GENERATION

The personalizer must ensure that the generated keys cannot be accessed by unauthorized users.

OE.CA-KEYS

The security domain keys of the CA must be securely generated prior storage in the (U)SIM card.

OE.VA-KEYS

The security domain keys of the VA must be securely generated prior storage in the (U)SIM card.

5.2.5 Platform

OE.KEY-CHANGE

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

5.2.6 GlobalPlatform

OE.SECURITY-DOMAINS

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

OE.QUOTAS

Security domains are subject to quotas of memory at creation.

5.2.7 Applications

OE.SHARE-CONTROL

All applications (basic and secure applications) must have means to identify the applications with whom they share data using the Shareable Interface.

Application Note:

> If an application implementing a Shareable Interface has to share data with a new application, it has to be updated, and thus re-validated, to take into account the identification of this new application (through its AID for instance) before sharing data.

5.3 Security Objectives Rationale

The reader is referred to the Security Objectives Rational of the Protection Profiles:

- The rationale for the TOE of Basic configuration in [PP(U)SIM], Sections 4.3.1.1, 4.3.2.1, 4.3.3 and 4.3.4, and
- the rationale in [PPJCSv3.0], Section 6.3.

Those parts of the rationale, which deviate from the rationale in the PPs are included in this ST and the differences are underlined for easier comparison.

5.3.1 Threats

This Security Target includes the rationale from [PP(U)SIM], Section 4.3.1.1.

All occurrences of OE.SCP.IC, OE.SCP.RECOVERY and OE.SCP.SUPPORT have been changed to O.SCP.IC, O.SCP.RECOVERY and O.SCP.SUPPORT, respectively.

In the [PP(U)SIM], O.CARD_MANAGEMENT replaces OE.CARD_MANAGEMENT originally defined in [PPJCSv3.0].

The coverage of the threats T.INTEG-USER-DATA and T.PHYSICAL has been adjusted as indicated in the subsection below. The coverage in Table 9 and in Table 10 has been adjusted to reflect that.

The newly introduced T.SECURE_DELETION is covered by the security objective O.DELETION. The assumption A.DELETION has been removed.

5.3.1.1 CONFIDENTIALITY

T.CONFID-APPLI-DATA This threat is countered by the security objective for the operational environment regarding bytecode verification (OE.VERIFICATION) and application validation (OE.BASIC-APPS-VALIDATION). 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 countermeasure 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). 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, 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.

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 This threat 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 and OE.BASIC-APPS-VALIDATION, so no application can be run to disclose a piece of code.

The (#.VERIFICATION) security aspect is addressed in this security target 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 This threat 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 countermeasure 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.

5.3.1.2 INTEGRITY

T.INTEG-APPLI-CODE This threat 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 and OE.BASIC-APPS-VALIDATION, 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 This threat is countered by the security objective O.LOAD which ensures that the loading of packages is done securely and thus preserves the integrity of packages 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 This threat is countered by bytecode verification (OE.VERIFICATION), by application validation (OE.BASIC-APPS-VALIDATION) 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 countermeasure 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). 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.

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.INTEG-APPLI-DATA.LOAD This threat is countered by the security objective O.LOAD which ensures that the loading of packages 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 This threat 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 and OE.BASIC-APPS-VALIDATION, 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 This threat is countered by bytecode verification (OE.VERIFICATION) and the isolation commitments stated in the (O.FIREWALL) objective and application validation (OE.BASIC-APPS-VALIDATION). 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 countermeasure 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.

5.3.1.3 IDENTITY USURPATION

T.SID.1 As impersonation is usually the result of successfully disclosing and modifying some assets, this threat 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 This 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.

5.3.1.4 UNAUTHORIZED EXECUTION

T.EXE-CODE.1 Unauthorized execution of a method is prevented by the objective OE.VERIFICATION and OE.BASIC-APPS-VALIDATION. 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 Unauthorized execution of a method fragment or arbitrary data is prevented by the objective OE.VERIFICATION and OE.BASIC-APPS-VALIDATION. 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 This threat is countered by O.NATIVE and OE.BASIC-APPS-VALIDATION which ensures that a Java Card applet can only access native methods indirectly that is, through an API. OE.APPLET also covers this threat by ensuring that no native applets 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).

5.3.1.5 DENIAL OF SERVICE

T.RESOURCES This threat is directly countered by objectives on resourcemanagement (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 singlethreaded 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 security target, 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.

5.3.1.6 CARD MANAGEMENT

T.DELETION This threat is covered by the O.DELETION security objective which ensures that both applet and package deletion perform as expected.

The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

T.SECURE_DELETION This threat is covered by the O.DELETION objective which ensures that deletion through the card manager is secure.

T.INSTALL This threat 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 package into the card is safe.

The objective O.CARD-MANAGEMENT controls the access to card management functions and thus contributes to cover this threat.

5.3.1.7 SERVICES

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

5.3.1.8 Basic TOE Threats

T.PHYSICAL This threat is countered by physical protections which rely on the underlying platform which is defined to be part of the TOE.

The security objectives O.SCP-SUPPORT and O.SCP-IC (from [PPJCSv3.0]) protect sensitive assets of the platform against loss of integrity and confidentiality and especially ensure the TSFs cannot be bypassed or altered.

T.INTEG-USER-DATA The security objective O.SCP-SUPPORT provides functionality to ensure atomicity of sensitive operations, secure low level access control and protection against bypassing of the security features of the TOE. In particular, it explicitly ensures the independent protection in integrity of the platform data.

The security objectives O.DOMAIN-RIGHTS, OE.CA-KEYS, OE.VA-KEYS and OE.AP-KEYS ensure that personalization of the application by its associated security domain is only performed by the authorized AP.

The security objectives from [PPJCSv3.0] covering the threat T.INTEG-APPLI-DATA also cover this threat. Therefore, the objective OE.CODE-EVIDENCE contributes to cover this threat by ensuring the integrity of the user data loaded into the platform.

T.COM_EXPLOIT This threat is covered by the following security objectives:

- 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 (U)SIM card.
- O.COMM_CONFIDENTIALITY prevents from disclosing encrypted data transiting to the (U)SIM card.

T.UNAUTHORIZED_CARD_MNGT This threat is covered by the following security objectives:

- 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 (U)SIM card.
- O.APPLI-AUTH which requires for loading all applications to be authenticated.
- O.DOMAIN-RIGHTS which restricts the modification of an AP security domain keyset to the AP who owns it.

T.LIFE_CYCLE This threat is covered by the security objectives:

- O.CARD-MANAGEMENT that controls the access to card management functions such as the loading, installation, extradition or deletion of applets and prevent attacks intended to modify or exploit the current life cycle of applications
- O.DOMAIN-RIGHTS that restricts the use of an AP security domain keysets, and thus the management of the applications related to this SD, to the AP who owns it.

T.UNAUTHORIZED_ACCESS This threat is covered by the security objective on the operational environment of the TOE OE.SHARE-CONTROL which ensures that sharing objects functionality is strictly controlled to stop data transitive flows between applets and thus stop access to unauthorized data.

5.3.2 Organisational Security Policies

This ST includes the rationale from [PP(U)SIM], Section 4.3.2.1.

The coverage of OSPs is adjusted in Table 11.

5.3.3 Assumptions

This ST includes the rationale from [PP(U)SIM], Section 4.3.3.

The assumption A.DELETION was removed.

The coverage of assumptions is adjusted in Table 13.

5.3.4 SPD and Security Objectives

This ST includes the SPD as defined in [PP(U)SIM], Section 4.3.4, and [PPJCSv3.0], Section 6.4.3, taking account of the redefinition of objectives given in Section 5.1. The coverage tables for threats and objectives shall read as follows.

Threats	Security Objectives	Rational
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.ALARM, O.TRANSACTION, O.CIPHER, O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION, OE.BASIC-APPS-VALIDATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.CONFID-JCS-CODE	OE.VERIFICATION, O.CARD-MANAGEMENT, O.NATIVE, OE.BASIC-APPS-VALIDATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.CONFID-JCS-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.ALARM	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.INTEG-APPLI-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE, OE.BASIC-APPS-VALIDATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.INTEG-APPLI-CODE.LOAD	O.LOAD, O.CARD-MANAGEMENT, OE.CODE-EVIDENCE	[PPJCSv3.0], Section 6.3.1
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.ALARM, O.TRANSACTION, O.CIPHER, O.PIN-MNGT, O.KEY-MNGT, O.REALLOCATION, OE.CODE-EVIDENCE, OE.BASIC-APPS-VALIDATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.INTEG-APPLI-DATA.LOAD	O.LOAD, O.CARD-MANAGEMENT, OE.CODE-EVIDENCE	[PPJCSv3.0], Section 6.3.1
T.INTEG-JCS-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE, OE.BASIC-APPS-VALIDATION	[PPJCSv3.0], Section 6.3.1
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, OE.BASIC-APPS-VALIDATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.SID.1	O.CARD-MANAGEMENT, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.INSTALL, O.SID	[PP(U)SIM], Section 4.3.1
T.SID.2	O.SCP.RECOVERY, O.SCP.SUPPORT, O.SID, O.OPERATE, O.FIREWALL, O.INSTALL	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.EXE-CODE.1	OE.VERIFICATION, O.FIREWALL, OE.BASIC-APPS-VALIDATION	[PPJCSv3.0], Section 6.3.1
T.EXE-CODE.2	OE.VERIFICATION, OE.BASIC-APPS-VALIDATION	[PPJCSv3.0], Section 6.3.1
T.NATIVE	OE.VERIFICATION, OE.APPLET, O.NATIVE, OE.BASIC-APPS-VALIDATION	[PPJCSv3.0], Section 6.3.1
T.RESOURCES	O.INSTALL, O.OPERATE, O.RESOURCES, O.SCP.RECOVERY, O.SCP.SUPPORT	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.DELETION	O.DELETION, O.CARD-MANAGEMENT	[PPJCSv3.0], Section 6.3.1
T.SECURE_DELETION	O.DELETION	Section 5.3.1.6
T.INSTALL	O.INSTALL, O.LOAD, O.CARD-MANAGEMENT	[PPJCSv3.0], Section 6.3.1
T.OBJ-DELETION	O.OBJ-DELETION	[PPJCSv3.0], Section 6.3.1

T.PHYSICAL	O.SCP.IC, O.SCP.SUPPORT	Section 5.3.1;
		[PPJCSv3.0], Section 6.3.1;
		[PP(U)SIM], Section 4.3.1

T.INTEG-USER-DATA	O.DOMAIN-RIGHTS, O.SCP-SUPPORT, OE.CA-KEYS, OE.AP-KEYS, OE.VA-KEYS, OE.CODE-EVIDENCE	Section 5.3.1;
		[PPJCSv3.0], Section 6.3.1;
		[PP(U)SIM], Section 4.3.1

T.COM-EXPLOIT	O.COMM_AUTH, O.COMM_INTEGRITY, O.COMM_CONFIDENTIALITY	[PP(U)SIM], Section 4.3.1
T.UNAUTHORIZED_CARD-MNGT	O.CARD-MANAGEMENT, O.COMM_AUTH, O.COMM_INTEGRITY, O.APPLI-AUTH, O.DOMAIN-RIGHTS	[PP(U)SIM], Section 4.3.1
T.LIFE_CYCLE	O.CARD-MANAGEMENT, O.DOMAIN-RIGHTS	[PP(U)SIM], Section 4.3.1
T.UNAUTHORIZED_ACCESS	OE.SHARE-CONTROL	[PP(U)SIM], Section 4.3.1

Table 9 Threats and Security Objectives – Coverage

Security Objectives	Threats
O.SID	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.1, T.SID.2
O.FIREWALL	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.1, T.SID.2, T.EXECODE.1
O.GLOBAL-ARRAYS-CONFID	T.CONFID-APPLI-DATA, T.SID.1
O.GLOBAL-ARRAYS-INTEG	T.INTEG-APPLI-DATA, T.SID.1
O.NATIVE	T.CONFID-JCS-CODE, T.INTEG-APPLI-CODE, T.INTEG-JCS-CODE, T.NATIVE
O.OPERATE	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES
O.REALLOCATION	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.RESOURCES	T.RESOURCES
O.ALARM	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA
O.CIPHER	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.KEY-MNGT	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.PIN-MNGT	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.TRANSACTION	T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA
O.OBJ-DELETION	T.OBJ-DELETION
O.DELETION	T.DELETION, T.SECURE_DELETION
O.LOAD	T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA.LOAD, T.INSTALL
O.INSTALL	T.SID.1, T.SID.2, T.RESOURCES, T.INSTALL
OE.APPLET	T.NATIVE
OE.VERIFICATION	T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-APPLI-CODE, T.INTEG-APPLI-DATA, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, T.EXE-CODE.1, T.EXE-CODE.2, T.NATIVE
OE.CODE-EVIDENCE	T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.CONFID-JCS-CODE, T.INTEG-JCS-DATA, T.INTEG-USER-DATA
O.SCP.IC	T.PHYSICAL
O.SCP.RECOVERY	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES
O.SCP.SUPPORT	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES, T.PHYSICAL, T.INTEG-USER-DATA
O.CARD-MANAGEMENT	T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE, T.CONFID-JCS-DATA, T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD T.INTEG-APPLI-DATA, T.INTEG-APPLI-DATA.LOAD, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, T.SID.1, T.DELETION, T.INSTALL, T.UNAUTHORIZED_CARD-MNGT, T.LIFE_CYCLE
O.DOMAIN-RIGHTS	T.INTEG-USER-DATA, T.UNAUTHORIZED_CARD_MNGT, T.LIFE_CYCLE
O.APPLI-AUTH	T.UNAUTHORIZED_CARD_MNGT
O.COMM_AUTH	T.COM_EXPLOIT, T.UNAUTHORIZED_CARD_MNGT
O.COMM_INTEGRITY	T.COM_EXPLOIT, T.UNAUTHORIZED_CARD_MNGT
O.COMM_CONFIDENTIALITY	T.COM_EXPLOIT
OE.MOBILE-OPERATOR	
OE.OTA-ADMIN	

OE.APPS-PROVIDER	
OE.VERIFICATION-AUTHORITY	
OE.KEY-ESCROW	
OE.PERSONALIZER	
OE.CONTROLLING-AUTHORITY	
OE.PRODUCTION	
OE.SECURE-APPS-CERTIFICATION	
OE.BASIC-APPS-VALIDATION	T.CONFID-APPLI-DATA, T.CONFID-JCS-CODE, T.INTEG-APPLI-CODE, T.INTEG-APPLI-DATA, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA, T.EXE-CODE.1, T.EXE-CODE.2, T.NATIVE
OE.AID-MANAGEMENT	
OE.OTA-LOADING	
OE.OTA-SERVERS	
OE.AP-KEYS	T.INTEG-USER-DATA
OE.OPERATOR-KEYS	
OE.KEY-GENERATION	
OE.CA-KEYS	T.INTEG-USER-DATA
OE.VA-KEYS	T.INTEG-USER-DATA
OE.KEY-CHANGE	
OE.SECURITY-DOMAINS	
OE.QUOTAS	
OE.SHARE-CONTROL	T.UNAUTHORIZED_ACCESS

Table 10 Security Objectives and Threats – Coverage

Organisational Security Policies	Security Objectives	Rationale
OSP.VERIFICATION	OE.VERIFICATION, O.LOAD, OE.CODE-EVIDENCE, OE.BASIC-APPS-VALIDATION	[PPJCSv3.0], Section 6.3.2
OSP.SECURE-APPS-CERTIFICATION	OE.SECURE-APPS-CERTIFICATION	[PP(U)SIM], Section 4.3.2
OSP.BASIC-APPS-VALIDATION	OE.BASIC-APPS-VALIDATION	[PP(U)SIM], Section 4.3.2
OSP.SHARE-CONTROL	OE.SHARE-CONTROL	[PP(U)SIM], Section 4.3.2
OSP.AID-MANAGEMENT	OE.AID-MANAGEMENT	[PP(U)SIM], Section 4.3.2
OSP.OTA-LOADING	OE.OTA-LOADING	[PP(U)SIM], Section 4.3.2
OSP.OTA-SERVERS	OE.OTA-SERVERS	[PP(U)SIM], Section 4.3.2
OSP.APSD-KEYS	OE.AP-KEYS	[PP(U)SIM], Section 4.3.2
OSP.OPERATOR-KEYS	OE.OPERATOR-KEYS	[PP(U)SIM], Section 4.3.2
OSP.KEY-GENERATION	OE.KEY-GENERATION	[PP(U)SIM], Section 4.3.2
OSP.CASD-KEYS	OE.CA-KEYS	[PP(U)SIM], Section 4.3.2
OSP.VASD-KEYS	OE.VA-KEYS	[PP(U)SIM], Section 4.3.2
OSP.KEY-CHANGE	OE.KEY-CHANGE	[PP(U)SIM], Section 4.3.2
OSP.SECURITY-DOMAINS	OE.SECURITY-DOMAINS	[PP(U)SIM], Section 4.3.2
OSP.QUOTAS	OE.QUOTAS	[PP(U)SIM], Section 4.3.2

Table 11 OSPs and Security Objectives – Coverage

Security Objectives	Organisational Security Policies
O.SID	
O.FIREWALL	
O.GLOBAL-ARRAYS-CONFID	
O.GLOBAL-ARRAYS-INTEG	
O.NATIVE	
O.OPERATE	
O.REALLOCATION	
O.RESOURCES	
O.ALARM	
O.CIPHER	
O.KEY-MNGT	
O.PIN-MNGT	
O.TRANSACTION	
O.OBJ-DELETION	
O.DELETION	
O.LOAD	OSP.VERIFICATION
O.INSTALL	
OE.APPLET	
OE.VERIFICATION	OSP.VERIFICATION
OE.CODE-EVIDENCE	OSP.VERIFICATION
O.SCP.IC	
O.SCP.RECOVERY	
O.SCP.SUPPORT	
O.CARD-MANAGEMENT	
O.DOMAIN-RIGHTS	
O.APPLI-AUTH	
O.COMM_AUTH	
O.COMM_INTEGRITY	
O.COMM_CONFIDENTIALITY	
O.INPUT-VALIDATION	
O.DOS-DETECTION	
O.REPLAY	
OE.MOBILE-OPERATOR	
OE.OTA-ADMIN	
OE.APPS-PROVIDER	
OE.VERIFICATION-AUTHORITY	
OE.KEY-ESCROW	
OE.PERSONALIZER	
OE.CONTROLLING-AUTHORITY	
OE.PRODUCTION	
OE.SECURE-APPS-CERTIFICATION	OSP.SECURE-APPS-CERTIFICATION
OE.BASIC-APPS-VALIDATION	OSP.BASIC-APPS-VALIDATION, OSP.VERIFICATION

OE.AID-MANAGEMENT	OSP.AID-MANAGEMENT
OE.OTA-LOADING	OSP.OTA-LOADING
OE.OTA-SERVERS	OSP.OTA-SERVERS
OE.AP-KEYS	OSP.APSD-KEYS
OE.OPERATOR-KEYS	OSP.OPERATOR-KEYS
OE.KEY-GENERATION	OSP.KEY-GENERATION
OE.CA-KEYS	OSP.CASD-KEYS
OE.VA-KEYS	OSP.VASD-KEYS
OE.KEY-CHANGE	OSP.KEY-CHANGE
OE.SECURITY-DOMAINS	OSP.SECURITY-DOMAINS
OE.QUOTAS	OSP.QUOTAS
OE.SHARE-CONTROL	OSP.SHARE-CONTROL

Table 12 Security Objectives and OSPs – Covertage

The assumption A.DELETION has been removed (replaced by the threat T.SECURE_DELETION) and does therefore not appear in Table 13 and in Table 14.

Assumptions	Security Objectives for the Operational Environment	Rationale
A.APPLET	OE.APPLET	[PPJCSv3.0], Section 6.3.3
A.VERIFICATION	OE.VERIFICATION, OE.CODE-EVIDENCE	[PPJCSv3.0], Section 6.3.3
A.MOBILE-OPERATOR	OE.MOBILE-OPERATOR	[PP(U)SIM], Section 4.3.3
A.OTA-ADMIN	OE.OTA-ADMIN	[PP(U)SIM], Section 4.3.3
A.APPS-PROVIDER	OE.APPS-PROVIDER	[PP(U)SIM], Section 4.3.3
A.VERIFICATION-AUTHORITY	OE.VERIFICATION-AUTHORITY	[PP(U)SIM], Section 4.3.3
A.KEY-ESCROW	OE.KEY-ESCROW	[PP(U)SIM], Section 4.3.3
A.PERSONALIZER	OE.PERSONALIZER	[PP(U)SIM], Section 4.3.3
A.CONTROLLING-AUTHORITY	OE.CONTROLLING-AUTHORITY	[PP(U)SIM], Section 4.3.3
A.PRODUCTION	OE.PRODUCTION	[PP(U)SIM], Section 4.3.3

Table 13 Assumptions and Security Objectives for the Operational Environment – Coverage

The objectives OE.SCP.IC, OE.SCP.RECOVERY, OE.SCP.SUPPORT have been changed to objectives on the TOE and do not appear in Table 14.

Security Objectives for the Operational Environment	Assumption
OE.APPLET	A.APPLET
OE.VERIFICATION	A.VERIFICATION
OE.CODE-EVIDENCE	A.VERIFICATION
OE.MOBILE-OPERATOR	A.MOBILE-OPERATOR
OE.OTA-ADMIN	A.OTA-ADMIN
OE.APPS-PROVIDER	A.APPS-PROVIDER
OE.VERIFICATION-AUTHORITY	A.VERIFICATION-AUTHORITY
OE.KEY-ESCROW	A.KEY-ESCROW
OE.PERSONALIZER	A.PERSONALIZER
OE.CONTROLLING-AUTHORITY	A.CONTROLLING-AUTHORITY
OE.PRODUCTION	A.PRODUCTION
OE.SECURE-APPS-CERTIFICATION	
OE.BASIC-APPS-VALIDATION	
OE.AID-MANAGEMENT	
OE.OTA-LOADING	
OE.OTA-SERVERS	
OE.AP-KEYS	
OE.OPERATOR-KEYS	
OE.KEY-GENERATION	
OE.CA-KEYS	
OE.VA-KEYS	
OE.KEY-CHANGE	
OE.SECURITY-DOMAINS	
OE.QUOTAS	
OE.SHARE-CONTROL	

Table 14 Security Objectives for the Operational Environment and Assumptions – Coverage

6 Extended Components Definition

6.1 Generation of random numbers (FCS_RNG)

This family describes the functional requirements for random number generation used for cryptographic purposes.

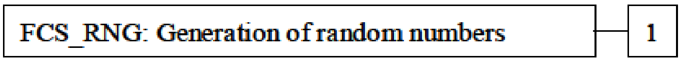
The family ‘Generation of random numbers (FCS_RNG)’ is specified as follows:

FCS_RNG Generation of random numbers

Family behaviour

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

Component levelling:



FCS_RNG.1 Generation of random numbers requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 Quality metric for random numbers

Hierarchical to: No other components

Dependencies: No dependencies

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

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

7 Security Requirements

7.1 Security Functional Requirements for the TOE

All mandatory SFRs of the [PPJCSv3.0] and (U)SIM PP are relevant to the TOE of this Security Target.

This Security Target adds the following SFRs in addition to those defined in the Protection Profiles: FPT_PHP.3, FCS_RNG.1.

SFR	JCS PP	(U)SIM PP Basic TOE	ST TOE
CoreG_LC			
FDP_ACC.2.1/FIREWALL	D	x	x
FDP_ACC.2.2/FIREWALL	D	x	x
FDP_ACF.1.1/FIREWALL	D	x	x
FDP_ACF.1.2/FIREWALL	D	x	x
FDP_ACF.1.3/FIREWALL	D	x	x
FDP_ACF.1.4/FIREWALL	D	x	x
FDP_IFC.1.1/JCVM	D	x	x
FDP_IFF.1.1/JCVM	D	x	x
FDP_IFF.1.2/JCVM	D	x	x
FDP_IFF.1.3/JCVM	D	x	x
FDP_IFF.1.4/JCVM	D	x	x
FDP_IFF.1.5/JCVM	D	x	x
FDP_RIP.1.1/OBJECTS	D	x	x
FMT_MSA.1.1/JCRE	D	x	x
FMT_MSA.1.1/JCVM	D	x	x
FMT_MSA.2.1/FIREWALL_JCVM	D	x	x
FMT_MSA.3.1/FIREWALL	D	x	x
FMT_MSA.3.2/FIREWALL	D	x	x
FMT_MSA.3.1/JCVM	D	x	x
FMT_MSA.3.2/JCVM	D	x	x
FMT_SMF.1.1	D	x	x
FMT_SMR.1.1	D	x	x
FMT_SMR.1.2	D	x	x
FCS_CKM.1.1	D	x	x
FCS_CKM.2.1	D	x	x
FCS_CKM.3.1	D	x	x
FCS_CKM.4.1	D	x	x
FCS_COP.1.1	D	x	x
FDP_RIP.1.1/ABORT	D	x	x
FDP_RIP.1.1/APDU	D	x	x
FDP_RIP.1.1/bArray	D	x	x
FDP_RIP.1.1/KEYS	D	x	x
FDP_RIP.1.1/TRANSIENT	D	x	x
FDP_ROL.1.1/FIREWALL	D	x	x
FDP_ROL.1.2/FIREWALL	D	x	x
FAU_ARP.1.1	D	x	x
FDP_SDI.2.1	D	x	x
FDP_SDI.2.2	D	x	x
FPR_UNO.1.1	D	x	x
FPT_FLS.1.1	D	x	x
FPT_TDC.1.1	D	x	x
FPT_TDC.1.2	D	x	x
FIA_ATD.1.1/AID	D	x	x
FIA_UID.2.1/AID	D	x	x
FIA_USB.1.1/AID	D	x	x
FIA_USB.1.2/AID	D	x	x

FIA_USB.1.3/AID	D	x	x
FMT_MTD.1.1/JCRE	D	x	x
FMT_MTD.3.1/JCRE	D	x	x
InstG			
FDP_ITC.2.1/Installer	D	x	Covered by FDP_ITC.2/CCM
FDP_ITC.2.2/Installer	D	x	
FDP_ITC.2.3/Installer	D	x	
FDP_ITC.2.4/Installer	D	x	
FDP_ITC.2.5/Installer	D	x	
FMT_SMR.1.1/Installer	D	x	x
FMT_SMR.1.2/Installer	D	x	x
FPT_FLS.1.1/Installer	D	x	Covered by FPT_FLS.1/CCM
FPT_RCV.3.1/Installer	D	x	x
FPT_RCV.3.2/Installer	D	x	x
FPT_RCV.3.3/Installer	D	x	x
FPT_RCV.3.4/Installer	D	x	x
AdelG			
FDP_ACC.2.1/ADEL	D	x	x
FDP_ACC.2.2/ADEL	D	x	x
FDP_ACF.1.1/ADEL	D	x	x
FDP_ACF.1.2/ADEL	D	x	x
FDP_ACF.1.3/ADEL	D	x	x
FDP_ACF.1.4/ADEL	D	x	x
FDP_RIP.1.1/ADEL	D	x	x
FMT_MSA.1.1/ADEL	D	x	x
FMT_MSA.3.1/ADEL	D	x	x
FMT_MSA.3.2/ADEL	D	x	x
FMT_SMF.1.1/ADEL	D	x	x
FMT_SMR.1.1/ADEL	D	x	x
FMT_SMR.1.2/ADEL	D	x	x
FPT_FLS.1.1/ADEL	D	x	x
ODELG			
FDP_RIP.1.1/ODEL	D	x	x
FPT_FLS.1.1/ODEL	D	x	x
CarG			
FCO_NRO.2.1/CM	D	Refined by FCO_NRO.2.1/SC	x
FCO_NRO.2.2/CM	D		x
FCO_NRO.2.3/CM	D		x
FDP_IFC.2.1/CM	D	Refined by FDP_IFC.2/SC	-
FDP_IFC.2.2/CM	D		-
FDP_IFF.1.1/CM	D	Refined by FDP_IFF.1/SC	-
FDP_IFF.1.2/CM	D		-
FDP_IFF.1.3/CM	D		-
FDP_IFF.1.4/CM	D		-
FDP_IFF.1.5/CM	D		-
FDP_UIT.1.1/CM	D	x	Covered by FDP_UIT.1/CCM
FDP_UIT.1.2/CM	D	x	
FIA_UID.1.1/CM	D	Refined by FIA_UID.1/SC	-
FIA_UID.1.2/CM	D		-

FMT_MSA.1.1/CM	D	x	Covered by FMT_MSA.1/SC
FMT_MSA.3.1/CM	D	x	Covered by FMT_MSA.3/SC
FMT_MSA.3.2/CM	D	x	
FMT_SMF.1.1/CM	D	x	Covered by FMT_SMF.1/SC
FMT_SMR.1.1/CM	D	x	Covered by FMT_SMR.1/Installer
FMT_SMR.1.2/CM	D	x	
FTP_ITC.1.1/CM	D	Refined by FTP_ITC.1/SC	-
FTP_ITC.1.2/CM	D		-
FTP_ITC.1.3/CM	D		-
CMGRG			
FDP_UIT.1.1/CCM	-	A	x
FDP_UIT.1.2/CCM	-	A	x
FDP_ROL.1.1/CCM	-	A	x
FDP_ROL.1.2/CCM	-	A	x
FDP_ITC.2.1/CCM	-	A	x
FDP_ITC.2.2/CCM	-	A	x
FDP_ITC.2.3/CCM	-	A	x
FDP_ITC.2.4/CCM	-	A	x
FDP_ITC.2.5/CCM	-	A	x
FPT_FLS.1.1/CCM	-	A	x
FCS_COP.1.1/DAP	-	R	x
FDP_ACC.1.1/SD	-	A	x
FDP_ACF.1.1/SD	-	A	x
FDP_ACF.1.2/SD	-	A	x
FDP_ACF.1.3/SD	-	A	x
FDP_ACF.1.4/SD	-	A	x
FMT_MSA.1.1/SD	-	A	x
FMT_MSA.3.1/SD	-	A	x
FMT_MSA.3.2/SD	-	A	x
FMT_SMF.1.1/SD	-	A	x
FMT_SMR.1.1/SD	-	A	x
FMT_SMR.1.2/SD	-	A	x
FTP_ITC.1.1/SC	-	R	x
FTP_ITC.1.2/SC	-	R	x
FTP_ITC.1.3/SC	-	R	x
FCO_NRO.2.1/SC	-	R	x
FCO_NRO.2.2/SC	-	R	x
FCO_NRO.2.3/SC	-	R	x
FDP_IFC.2.1/SC	-	R	x
FDP_IFC.2.2/SC	-	R	x
FDP_IFF.1.1/SC	-	R	x
FDP_IFF.1.2/SC	-	R	x
FDP_IFF.1.3/SC	-	R	x
FDP_IFF.1.4/SC	-	R	x
FDP_IFF.1.5/SC	-	R	x
FMT_MSA.1.1/SC	-	A	x
FMT_MSA.3.1/SC	-	A	x

FMT_MSA.3.2/SC	-	A	x
FMT_SMF.1.1/SC	-	A	x
FIA_UID.1.1/SC	-	R	x
FIA_UID.1.2/SC	-	R	x
FIA_UAU.1.1/SC	-	A	x
FIA_UAU.1.2/SC	-	A	x
FIA_UAU.4.1/SC	-	A	x
SCPG			
FPT_PHP.3.1	-	-	A
Extended components			
FCS_RNG.1	-	-	A

Table 15 Security Functional Requirements

7.1.1 Java Card System SFRs

The JCS PP arranges requirements into groups. The following groups defined in [PPJCSv3.0], Section 7.1, apply to the TOE of this Security Target:

Group
Core with Logical Channel (CoreG_LC)
Installation (InstG)
Applet deletion (ADELG)
Object deletion (ODELG)
Secure carrier (CarG)

The TOE does not support the group RMIG related to Remote Method Invocation (RMI).

The JCS PP [PPJCSv3.0], Section 7.1, defines Subjects (prefixed with an "S"), Objects (prefixed with an "O") and Information (prefixed with an "I"). The following Subjects, Objects and Information are relevant for this Security Target:

Subject	Object	Information
S.ADEL	O.APPLET	I.APDU
S.APPLET	O.CODE_PKG	I.DATA
S.BCV	O.JAVAOBJECT	
S.CAD		
S.INSTALLER		
S.JCRE		
S.CJVM		
S.LOCAL		
S.MEMBER		
S.PACKAGE		

The Objects O.REMOTE_MTHD, O.REMOTE_OBJ, O.RMI_SERVICE, O.ROR, and the Information I.RORD defined in JCS PP are related to RMI and therefore not relevant for this Security Target.

The JCS PP further defines security attributes linked to these Subjects, Objects and Information that are listed here without repeating the description (or values) provided in [PPJCSv3.0], Section 7.1:

Security Attribute	
Active Applets	LifeTime
Applet Selection Status	Owner
Applet's version number	Package AID
Context	Registered Applets
Currently Active Context	Resident Packages
Dependent package AID	Selected Applet Context
Identifier	Sharing
LC Selection Status	Static References

The security attributes Class, Remote and Returned References are related to RMI and therefore not relevant for this Security Target.

The JCS PP defines the following operations (prefixed with "OP") and their parameters that are listed here without repeating the description:

Operation
OP.ARRAY_ACCESS(O.JAVAOBJECT, field)
OP.CREATE(Sharing, LifeTime)
OP.DELETE_APPLET(O.APPLET, ...)
OP.DELETE_PCKG(O.CODE_PKG, ...)
OP.DELETE_PCKG_APPLET(O.CODE_PKG, ...)
OP.INSTANCE_FIELD(O.JAVAOBJECT, field)
OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, ...)
OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1, ...)
OP.JAVA(...)
OP.PUT(S1, S2, I)
OP.THROW(O.JAVAOBJECT)
OP.TYPE_ACCESS(O.JAVAOBJECT, class)

The operations OP.GET_ROR, OP.INVOKE, OP.RET_RORD are related to RMI and therefore not relevant for this Security Target.

7.1.1.1 CoreG_LC Security Functional Requirements

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

7.1.1.1.1 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.PACKAGE, S.JCRE, S.JCVM, O.JAVAOBJECT** and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- OP.CREATE,
- OP.INVK_INTERFACE,
- OP.INVK_VIRTUAL,
- OP.JAVA,
- OP.THROW,
- OP.TYPE_ACCESS.

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:

> 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.PACKAGE	LC Selection Status
S.JCVM	Active Applets, Currently Active Context
S.JCRE	Selected Applet Context
O.JAVAOBJECT	Sharing, Context, LifeTime

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

- R.JAVA.1 ([JCRE], §6.2.8): S.PACKAGE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, 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 ([JCRE], §6.2.8): S.PACKAGE 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 ([JCRE], §6.2.8.10): S.PACKAGE may perform OP.TYPE_ACCESS upon an O.JAVAOBJECT whose Sharing attribute has value "SIO" only if O.JAVAOBJECT is being cast into (checkcast) or is being verified as being an instance of (instanceof) an interface that extends the Shareable interface.

- R.JAVA.4 ([JCRE], §6.2.8.6): S.PACKAGE may perform OP.INVK_INTERFACE upon an O.JAVAOBJECT whose Sharing attribute has the value "SIO", and whose Context attribute has the value "Package AID", only if the invoked interface method extends the Shareable interface and one of the following conditions applies:
  a) The value of the attribute Selection Status of the package whose AID is "Package AID" is "Multiselectable",
  b) The value of the attribute Selection Status of the package whose AID is "Package AID" is "Non-multiselectable", and either "Package AID" is the value of the currently selected applet or otherwise "Package AID" does not occur in the attribute Active Applets.
- R.JAVA.5: S.PACKAGE may perform OP.CREATE only if the value of the Sharing parameter is "Standard".

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

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.

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 is done by ensuring that references to objects belonging to a deleted application are considered as a null reference.

In the case of an array type, fields are components of the array ([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 four categories of objects:

- Standard ones, whose both fields and methods are under the firewall policy,
- Shareable interface Objects (SIO), which provide a secure mechanism for inter-applet communication,
- JCRE entry points (Temporary or Permanent), who have freely accessible methods but protected fields,

- Global arrays, having both unprotected fields (including components) 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 ([JCRE], §6.1.3). An object is owned by an applet instance, by the JCRE or by the package library where it has been defined (these latter objects can only be arrays that initialize static fields of packages).

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

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

([JCRE], §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).

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

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

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

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:

Subject	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 following additional information flow control SFP rules: none.

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

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

Application Note:

> The storage of temporary Java Card RE-owned objects references is runtime-enforced ([JCRE], §6.2.8.1-3). This policy essentially applies to the execution of bytecode. Native methods, the Java Card RE itself and possibly some API methods are granted specific rights or limitations through the FDP_IFF.1.3/JCVM to FDP_IFF.1.5/JCVM elements.

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 [JCRE], §4 and [JCVM], §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 is performed in accordance with the rules given in [JCRE], §4 and [JCVM], §3.4.

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.
> - An O.JAVAOBJECT whose Sharing attribute value is a global array necessarily has "array of primitive type" as a JavaCardClass security attribute's value.
> - Any O.JAVAOBJECT whose Sharing attribute value is not "Standard" has a PERSISTENT-LifeTime attribute's value.
> - Any O.JAVAOBJECT whose LifeTime attribute value is not PERSISTENT has an array type as JavaCardClass attribute's value.

FMT_MSA.3/FIREWALL Static attribute initialisation

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:

> FDP_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 ([JCRE], §6.1.3). There is one default value for the Selected Applet Context that is the default applet identifier's Context, and one default value for the Currently Active Context that is "Java Card RE".

- The knowledge of which reference corresponds to a temporary entry point object or a global array and which does not is solely available to the Java Card RE (and the Java Card virtual machine).

FMT_MSA.3.2/FIREWALL:

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

FMT_MSA.3/JCVM Static attribute initialisation

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.

7.1.1.1.2 Application Programming Interface

The following SFRs are related to the Java Card API.

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

FCS_CKM.1 Cryptographic key generation

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

Iteration	Algorithm	Key size	List of standards
/TDES	Triple-DES key generation	112, 168 bits	[SP800-67] Sections 3.4.1 and 3.4.2
/AES	AES key generation	128, 192, 256 bits	[FIPS 197] Sections 3.1 and 5
/RSA	RSA and RSA-CRT key generation	512-2048 bits	[FIPS 186-4] Section 5.1 and B.3.3

Application Note:

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

Application Note (ST author):

> For generation of RSA-CRT, RSA, Triple-DES and AES keys the TOE uses the random number generator according to [ISO/IEC 18031].
>
> There is no dedicated Java Card API for generation of symmetric keys (AES, Triple-DES). For this purpose, the javacard.security.RandomData.generateData shall be used.

FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[assignment: cryptographic key distribution method]** that meets the following: **[assignment: list of standards]**.

Iteration	Key distribution method	Standards
/TDES	set methods in [JCAPI] javacard.security class DESKey	none
/AES	set methods in [JCAPI] javacard.security class AESKey	none
/RSA	set methods in [JCAPI] javacard.security classes RSAPrivateCrtKey, RSAPrivateKey and RSAPublicKey	none

Application Note:

> Command SetKEY meets [JCAPI] specification.

FCS_CKM.3 Cryptographic key access

FCS_CKM.3.1 The TSF shall perform **[assignment: type of cryptographic key access]** in accordance with a specified cryptographic key access method **[assignment: cryptographic key access method]** that meets the following: **[assignment: list of standards]**.

Iteration	Key access method	Standards
/TDES	get methods in [JCAPI] javacard.security class DESKey	none
/AES	get methods in [JCAPI] javacard.security class AESKey	none
/RSA	get methods in [JCAPI] javacard.security classes RSAPrivateCrtKey, RSAPrivateKey and RSAPublicKey	none

Application Note:

> The keys can be accessed as specified in [JCAPI] Key class.

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 [JCAPI] javacard.security.Key.clearKey() and overwriting the keys with zeros that meets the following: none.

Application Note:

> The keys are reset as specified in the [JCAPI] Key class, with the method clearKey(). Any access to a cleared key for ciphering or signing throws an exception.

FCS_COP.1 Cryptographic operation

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

Iteration	Operation	Algorithm	Key sizes	Standards
/SHA	hashing	SHA-1, SHA-224, SHA-256, SHA-384, SHA-512	n.a.	[FIPS 180-4] Sections 6.1-6.5
/SIG_RSA	digital signature generation and verification	RSA and RSA-CRT both with (RSASSA-PSS, RSASSA-PKCS1-V1_5) using SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512	512-2048 bits	[FIPS 186-4] Section 5.5, [RFC3447] Section 8
/MAC_TDES	MAC generation and verification	Triple-DES CBC MAC	112, 168 bits	[FIPS 46-3], Chapter 'TRIPLE DATA ENCRYPTION ALGORITHM',[ISO/IEC 9797-1] Sections 6.6.3, 7.1, 7.3
/MAC_AES		AES CBC MAC, CMAC	128, 192, 256 bits	[FIPS 197] Section 5, [ISO/IEC 9797-1] Section 7.1, [SP800-38B], Section 6
/CIPH_TDES	encryption and decryption	Triple-DES in CBC and ECB modes	112, 168 bits	[SP800-67] all normative sections, [SP800-38A] Sections 6.1 and 6.2
/CIPH_AES		AES in CBC and ECB modes	128, 192, 256 bits	[FIPS 197] Section 5, [SP800-38A] Sections 6.1 and 6.2
/CIPH_RSA		RSA	512-2048 bits	[RFC3447] Section 7.2 (RSAES-PKCS1-V1_5, NOPAD)

Application Note:

> The TOE provides a subset of cryptographic operations defined in [JCAPI] (see javacardx.crypto.Cipher and javacard.security packages).

Application Note (ST author):

> AES CMAC is used for MAC calculations as part of SCP80 ([TS102 225], Section 5.1.3).

Random Number Generation (FCS_RNG.1)

The TOE meets the requirement "Quality metric for random numbers (FCS_RNG.1)" as specified below (Common Criteria Part 2 extended).

FCS_RNG.1 Quality metric for random numbers

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a hybrid deterministic random number generator that implements:

The internal state of the RNG consists of a Triple DES key (three times 56 bits) and a DES block (64 bit). Guessing the internal state requires guessing 232 bits.

The RNG provides forward secrecy.

The RNG provides backward secrecy, even if the current internal state is known.

Refinement:

FCS_RNG.1.2 The TSF shall provide random numbers that meet:

The RNG, initialized with a random seed generated from statistically tested TRNG output, generates output for which two strings of bit length 128 are mutually different with probability 1 - 2^128.
Statistical test suites cannot practically distinguish the random number from output sequences of an ideal RNG. The random numbers pass test procedure A as defined in AIS20/31.

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 [JCRE], §5.1.

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

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

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 [JCRE], §7.7, within the bounds of the Commit Capacity ([JCRE], §7.8), and those described in [JCAPI].**

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. Some operations of the API are not conditionally updated, as documented in [JCAPI] (see for instance, PIN-blocking, PIN-checking, update of transient objects).

7.1.1.1.3 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**
- no other actions

upon detection of a potential security violation.

Refinement:

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

- CAP file inconsistency,
- typing error in the operands of a bytecode,
- applet life cycle inconsistency,
- card tearing (unexpected removal of the card out of the CAD) and power failure, abort of a transaction in an unexpected context, (see abortTransaction(), [JCAPI] and [JCRE], §7.6.2)
- violation of the Firewall or JCVM SFPs,
- unavailability of resources,
- array overflow,
- flow control errors
- other runtime errors related to applet's failure (like uncaught exceptions).

Application Note:

> For the TOE in this ST bytecode verification is performed off-card.

FDP_SDI.2 Stored data integrity monitoring and action

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for **integrity errors** on all objects, based on the following attributes: complementary value, EDC.

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall bring the card into a secure state.

Application Note:

> The TOE raises an exception upon detection of integrity error on cryptographic keys, PIN values and their associated security attributes. Cryptographic keys and PIN objects are considered as described in FDP_SDI.2.1.
>
> The TOE monitors integrity errors in the code of the native applications and Java Card applets.
>
> For integrity sensitive application, their data is monitored (D.APP_I_DATA): applications may need to protect information against unexpected modifications, and explicitly control whether a piece of information has been changed between two accesses.

FPR_UNO.1 Unobservability

FPR_UNO.1.1 The TSF shall ensure that unauthorized users or subjects are unable to observe the operation cryptographic operations, comparison operations on key values, PIN values by S.JCRE, S.Applet, S.SD.

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 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 ([JCRE], §6.2.3) or after a proximity card (PICC) activation sequence ([JCRE]). Behavior of the TOE on power loss and reset is described in [JCRE], §3.6 and §7.1. Behavior of the TOE on RF signal loss is described in [JCRE], §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 [JCVM] specification,**
- **the API tokens defined in the export files of reference implementation**,

when interpreting the TSF data from another trusted IT product.

Application Note:

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

7.1.1.1.4 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:

- Package AID,
- Applet's version number,
- Registered applet AID,
- Applet Selection Status ([JCVM], §6.5).

Application Note:

> "Individual users" stand for applets.

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 packages (or applets) that act as subjects of policies. In the Java Card System, every action is always performed by an identified user interpreted here as the currently selected applet or the package that is the subject's owner. Means of identification are provided during the loading procedure of the package and the registration of applet instances.
> - The role Java Card RE defined in FMT_SMR.1 is attached to an IT security function rather than to a "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: Package AID.

FIA_USB.1.2/AID The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: rules defined in FMT_MSA.2/FIREWALL_JCVM and FMT_MSA.3.1/FIREWALL.

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: rules defined in FMT_MSA.3.1/FIREWALL.

Application Note:

> The user is the applet and the subject is the S.PACKAGE. The subject security attribute "Context" shall hold the user security attribute "package AID".

FMT_MTD.1/JCRE Management of TSF data

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. 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 are granted the right to modify the list of registered applets' AIDs (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.

7.1.1.2 InstG Security Functional Requirements

This group consists of the SFRs related to the installation of the applets, which addresses security aspects outside the runtime. The installation of applets is a critical phase, which lies partially out of the boundaries of the firewall, and therefore requires specific treatment. Loading a package or installing an applet is modelled as importation of user data (that is, user application's data) with its security attributes (such as the parameters of the applet used in the firewall rules).

FDP_ITC.2/Installer SFRs are covered by FDP_ITC.2/CCM SFRs, see section 7.1.2.1.1.

FMT_SMR.1/Installer Security roles

FMT_SMR.1.1/Installer The TSF shall maintain the roles: Installer.

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

Application Note (ST author):

> The subject S.SD includes the subject S.INSTALLER.

FPT_FLS.1.1/Installer is covered by FPT_FLS.1.1/CCM, see section 7.1.2.1.1.

FPT_RCV.3/Installer Automated recovery without undue loss

FPT_RCV.3.1/Installer When automated recovery from power loss is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2/Installer For reset, insufficient non-volatile memory, failure in cryptographic safeguarding, package references (versions) mismatch, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

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

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

Application Note:

> FPT_RCV.3.1/Installer:
>
> - This element is not within the scope of the Java Card specification, which only mandates the behavior of the Java Card System in good working order.
>
> FPT_RCV.3.2/Installer:
>
> - Should the installer fail during loading/installation of a package/applet, it has to revert to a "consistent and secure state". The Java Card RE has some clean up duties as well; see [JCRE], 11.1.5 for possible scenarios. This component includes among the listed failures the deletion of a package/applet. See ([JCRE], 11.3.4) for possible scenarios.
> - Other events such as the unexpected tearing of the card, power loss, and so on, are partially handled by the underlying hardware platform (see [PP0035]) and, from the TOE's side, by events "that clear transient objects" and transactional features. See FPT_FLS.1.1, FDP_RIP.1.1/TRANSIENT, FDP_RIP.1.1/ABORT and FDP_ROL.1/FIREWALL.
>
> FPT_RCV.3.3/Installer:
>
> - The SCP ensures the atomicity of updates for fields and objects, and a power-failure during a transaction or the normal runtime does not create the loss of otherwise-permanent data, in the sense that memory on a smart card is essentially persistent with this respect (flash NVM). Data stored on the RAM and subject to such failure is intended to have a limited lifetime anyway (runtime data on the stack, transient objects' contents). According to this, the loss of data within the TSF scope is limited to the same restrictions of the transaction mechanism.

7.1.1.3 AdelG Security Functional Requirements

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

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_PKG and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- OP.DELETE_APPLET,
- OP.DELETE_PCKG,
- OP.DELETE_PCKG_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 Packages
O.CODE_PKG	Package AID, 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 ([JCRE], §11.3.4.1, Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon an O.APPLET only if,
  (1) S.ADEL is currently selected,
  (2) there is no instance in the context of O.APPLET that is active in any logical channel and
  (3) there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance distinct from O.APPLET, or O.JAVAOBJECT is reachable from a package P, or ([JCRE], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.15 ([JCRE], §11.3.4.1, Multiple Applet Instance Deletion): S.ADEL may perform OP.DELETE_APPLET upon several O.APPLET only if,
  (1) S.ADEL is currently selected,
  (2) there is no instance of any of the O.APPLET being deleted that is active in any logical channel and
  (3) there is no O.JAVAOBJECT owned by any of the O.APPLET being deleted such that either O.JAVAOBJECT is reachable from an applet instance distinct from any of those O.APPLET, or O.JAVAOBJECT is reachable from a package P, or ([JCRE], §8.5) O.JAVAOBJECT is remote reachable.
- R.JAVA.16 ([JCRE], §11.3.4.2, Applet/Library Package Deletion): S.ADEL may perform OP.DELETE_PCKG upon an O.CODE_PKG only if,

(1) S.ADEL is currently selected,

(2) no reachable O.JAVAOBJECT, from a package distinct from O.CODE_PKG that is an instance of a class that belongs to O.CODE_PKG, exists on the card and

(3) there is no resident package on the card that depends on O.CODE_PKG.

- R.JAVA.17 ([JCRE], §11.3.4.3, Applet Package and Contained Instances Deletion): S.ADEL may perform OP.DELETE_PCKG_APPLET upon an O.CODE_PKG only if,

  (1) S.ADEL is currently selected,

  (2) no reachable O.JAVAOBJECT, from a package distinct from O.CODE_PKG, which is an instance of a class that belongs to O.CODE_PKG exists on the card,

  (3) there is no package loaded on the card that depends on O.CODE_PKG, and

  (4) for every O.APPLET of those being deleted it holds that: (i) there is no instance in the context of O.APPLET that is active in any logical channel and (ii) there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance not being deleted, or O.JAVAOBJECT is reachable from a package not being deleted, or ([JCRE], §8.5) O.JAVAOBJECT is remote reachable.

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

FDP_ACF.1.4/ADEL [Editorially Refined] The TSF shall explicitly deny access of 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 package.
> - S.ADEL calls the "uninstall" method of the applet instance to be deleted, if implemented by the applet, to inform it of the deletion request. The order in which these calls and the dependencies checks are performed is 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 packages 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/package deletion are described in [JCRE], §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 Packages** to **the Java Card RE**.

FMT_MSA.3/ADEL Static attribute initialisation

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.

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 Packages**.

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.

Application Note (ST author):

> The subject S.ADEL is represented by a Security Domain with Global Delete privilege or any Security Domain performing deletion of its associated Executable Load Files and Applications and their data. A Security Domain with Global Delete privilege has the privilege to delete any Application or Executable Load File from the card regardless of which Security Domain the Application or Executable Load File is associated with.
>
> The subject S.SD includes the subject S.ADEL.

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 package/applet as described in [JCRE], §11.3.4**.

Application Note:

- The TOE provides feedback information to the card manager in case of a potential security violation (see FAU_ARP.1).
- The Package/applet instance deletion is an atomic operation. The "secure state" referred to in the requirement complies with Java Card specification ([JCRE], §11.3.4.)

7.1.1.4 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() are reused. Requirements on de-allocation after the invocation of the method are described in [JCAPI].
> - 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 provides feedback information to the card manager in case of potential security violation (see FAU_ARP.1).

7.1.1.5 CarG Security Functional Requirements

This group includes requirements for preventing the installation of packages that has not been bytecode verified, or that has been modified after bytecode verification.

FCO_NRO.2/CM Enforced proof of origin

FCO_NRO.2.1/CM The TSF shall enforce the generation of evidence of origin for transmitted application packages at all times.

Application Note:

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

FCO_NRO.2.2/CM [Editorially Refined] The TSF shall be able to relate the identity of the originator of the information, and the application package contained in the information to which the evidence applies.

FCO_NRO.2.3/CM The TSF shall provide a capability to verify the evidence of origin of information to **recipient** given that the data origin authentication provided within the context of secure messaging was successful.

Application Note (ST author):

> FCO_NRO.2/CM is related to secure messaging by means of GlobalPlatform Secure Channel Protocol.
>
> In the context of secure messaging, message integrity also provides data origin authentication ([GP2.2.1], Section 10.5). The TOE performs verification of the origin of the package by applying command MAC verification. No evidence is kept on the card for future verifications.

FDP_IFC.2/CM SFRs are covered by FDP_IFC.2/SC SFRs, see section 7.1.2.1.3.

FDP_IFF.1/CM SFRs are covered by FDP_IFF.1/SC SFRs, see section 7.1.2.1.3.

FDP_UIT.1/CM SFRs are covered by FDP_UIT.1/CCM SFRs, see section 7.1.2.1.1.

FIA_UID.1/CM SFRs are covered by FIA_UID.1/SC SFRs, see section 7.1.2.1.3.

FMT_MSA.1/CM is covered by FMT_MSA.1/SC, see section 7.1.2.1.3.

FMT_MSA.3/CM SFRs are covered by FMT_MSA.3/SC SFRs, see section 7.1.2.1.3.

FMT_SMF.1/CM is covered by FMT_SMF.1/SC, see section 7.1.2.1.3.

FMT_SMR.1/CM is covered by FMT_SMR.1/Installer, see section 7.1.1.2.

FTP_ITC.1/CM SFRs are covered by FTP_ITC.1/SC SFRs, see section 7.1.2.1.3.

7.1.1.6 SCPG Security Functional Requirements

In [PPJCSv3.0], the objectives for the smart card platform are defined as objectives for the environment. Since the smart card platform is part of the TOE of this ST, the objectives for the environment were redefined as objectives for the TOE; they subsequently have to be covered by SFRs.

FPT_PHP.3 Resistance to physical attack

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

7.1.2 Basic TOE configuration SFRs

This section describes the SFRs for the Basic TOE configuration.

7.1.2.1 Card Manager (CMGRG)

This section contains the security requirements for the card manager.

7.1.2.1.1 Card Content Management

FDP_UIT.1/CCM Data exchange integrity

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

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

FDP_ROL.1/CCM Basic rollback

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

FDP_ROL.1.2/CCM The TSF shall permit operations to be rolled back within the the bounds of the Commit Capacity ([JCRE], Section 7.8).

FDP_ITC.2/CCM Import of user data with security attributes

FDP_ITC.2.1/CCM The TSF shall enforce the **Security Domain access control policy, the Card content management operation information flow policy and the Secure Channel Protocol information flow control policy** when importing user data, controlled under the SFP, from outside of the TOE.

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

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

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

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

- Package loading is allowed only if, for each dependent package, its AID attribute is equal to a resident package AID attribute, the major (minor) Version attribute associated to the dependent package is lesser than or equal to the major (minor) Version attribute associated to the resident package ([JCVM], §4.5.2).
- the rules defined in FDP_IFF.1.2/SC.

Application Note:

> This Functional Component Instance enforces a security information flow control policy. This TOE enforces the rules defined for importation operations. These rules take into account all user data.

FPT_FLS.1/CCM Failure with preservation of secure state

FPT_FLS.1.1/CCM The TSF shall preserve a secure state when the following types of failures occur: the Security Domain fails to load/install an Executable File / application instance as described in [JCRE], Section 11.1.5[1].

FCS_COP.1/DAP Cryptographic operation

FCS_COP.1.1/DAP The TSF shall perform verification of the DAP signature attached to Executable Load Applications in accordance with a specified cryptographic algorithm

- PKC Scheme: SHA-1 hash and PKCS#1 RSA signature
- or DES Scheme: Single DES plus final Triple DES MAC (Retail MAC)

and cryptographic key sizes

- PKC Scheme: RSA key of minimum length 1024 bits
- DES Scheme: DES key of minimum length 16 bytes

that meet the following:

- Sections C.1.2 and C.6 of [GP]
- PKC Scheme: SSA-PKCS1-v1_5 as defined in PKCS#1
- DES Scheme: ISO 9797-1 as MAC Algorithm 3 with output transformation 3, without truncation, and with DES taking the place of the block cipher.

Application Note (ST author):

> The TOE implements the DES-Scheme for DAP Verification.

7.1.2.1.2 Security Domain

FDP_ACC.1/SD Subset access control

FDP_ACC.1.1/SD The TSF shall enforce the Security Domain access control policy on:

- Subjects: S.INSTALLER, S.ADEL, S.CAD (from [PPJCSv3.0]) and S.SD
- Objects: Delegation Token, DAP Block and Load File
- Operations: GlobalPlatform's card content management APDU commands and API methods (defined in Appendix A of [GP])

FDP_ACF.1/SD Security attribute based access control

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

- Subjects:
    - o S.INSTALLER, defined in [PPJCSv3.0] and represented by the GlobalPlatform Environment (OPEN) on the card, the Card Life Cycle attributes (defined in Section 5.1.1 of [GP]);
    - o S.ADEL, also defined in [PPJCSv3.0] and represented by the GlobalPlatform Environment (OPEN) on the card;

[1] The [PP(U)SIM] references Section 11.3.5 in [JCRE] that is related to Applet Deletion Manager behavior. The reference was changed, since this SFR addresses the loading and installation.

  - S.SD receiving the Card Content Management commands (through APDUs or APIs) with an AID, a set of privileges (defined in Section 6.6.1 of [GP]), a life-cycle status (defined in Section 5.3.2 of [GP]) and a Secure Communication Security level (defined in Section 10.6 of [GP]);
  - S.CAD, defined in [PPJCSv3.0], the off-card entity that communicates with the S.INSTALLER through S.SD;
- Objects:
  - The Delegation Token, in case of Delegated Management operations, with the attributes Present or Not Present;
  - The DAP Block, in case of application loading, with the attributes Present or Not Present;
  - The Load File or Executable File, in case of application loading, installation, extradition or registry update, with a set of intended privileges and its targeted associated SD AID.
- Security attributes:
  - The Life Cycle State attributes of Applications, Security Domains, Executable Load Files, and the Card.
  - The Security Domain and Application privileges.

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

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

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

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

FMT_MSA.1/SD Management of security attributes

FMT_MSA.1.1/SD The TSF shall enforce the Security Domain access control policy to restrict the ability to modify the security attributes

(1) Application management information
(2) Card Life Cycle information
(3) Application Life Cycle information

to the Security Domain and the application instance itself.

FMT_MSA.3/SD Static attribute initialisation

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

FMT_MSA.3.2/SD The TSF shall allow the following roles: none to specify alternative initial values to override the default values when an object or information is created.

Refinement:

Alternative initial values shall be at least as restrictive as the default values defined in FMT_MSA.3.1.

FMT_SMF.1/SD Specification of Management Functions

FMT_SMF.1.1/SD The TSF shall be capable of performing the following management functions: Management functions defined in FMT_SMF.1.1/SC.

FMT_SMR.1/SD Security Roles

FMT_SMR.1.1/SD The TSF shall maintain the roles Issuer, Application Provider, Controlling Authority.

FMT_SMR1.2/SD The TSF shall be able to associate users with roles.

7.1.2.1.3 Secure Channel

FTP_ITC.1/SC Inter-TSF trusted channel

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

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

- **loading;**
- **installation;**
- **extradition;**
- **registry update;**
- **SD personalization;**
- Setup of initial keys and update of existing keys;
- Modifying the life cycle state of an application or Security Domain ([GP], Section 11.10) and modifying the card life cycle state ([GP], Section 11.11).

Application Note (ST author):

> The TOE allows:
>
> - The setup of initial Secure Channel keys according to the Scenario #2.B, Push Model without Application Provider Certificate ([GP UICC], Section 11.3);

- The replacement of existing keys for Secure Channel Protocols, Token Verification, Receipt Generation, and DAP Verification ([GP], Section 11.8).

FCO_NRO.2/SC Enforced proof of origin

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

FCO_NRO.2.2/SC The TSF shall be able to relate the Load File Data Block Signature of the originator of the information, and the **identity** of the information to which the evidence applies.

FCO_NRO.2.3/SC The TSF shall provide a capability to verify the evidence of origin of information to **originator** given **Executable load files**.

Application note (ST author):

> FCO_NRO.2/SC SFRs are related to GlobalPlatform DAP verification.
>
> FCO_NRO.2.2/SC:
>
> - The Load File Data Block Signature is a signature of the Load File Data Block Hash. Each Load File Data Block Signature is combined with its linked Security Domain AID in the TLV structured DAP Block. DAP Blocks are positioned in the beginning of the Load File ([GP], C.3).
>
> FCO_NRO.2.3/SC:
>
> - A DAP Block is included in a Load File if the associated Security Domain has the DAP Verification privilege or a Security Domain with the Mandated DAP Verification privilege is present ([GP2.2.1], Section 11.6.2.3).
> - If the Executable Load File contains a DAP Block (or multiple DAP Blocks), the associated Security Domain performs DAP Verification.

FDP_IFC.2/SC Complete information flow control

FDP_IFC.2.1/SC The TSF shall enforce the **Secure Channel Protocol information flow control policy** on

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

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

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

FDP_IFF.1/SC Simple security attributes

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

- Subjects:
  - S.SD receiving the Card Content Management commands (through APDUs or APIs). This subject can be the ISD, an APSD or a CASD.
  - S.CAD the off-card entity that communicates with the S.SD.
- Information:
  - load file, in case of application loading;
  - applications or SD privileges, in case of application installation or registry update;
  - personalization keys and/or certificates, in case of application or SD personalization.
  - initial/static keys, with the attributes key identifier and key version.

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

- Runtime behavior rules defined by GlobalPlatform for:
  - loading (Section 9.3.5 of [GP]);
  - installation (Section 9.3.6 of [GP]);
  - extradition (Section 9.4.1 of [GP]);
  - registry update (Section 9.4.2 of [GP]);
  - SD personalization rules, pull and push models (Section 11 of [GP UICC]).
- The subject S.INSTALLER shall accept a message only if it comes from the subject S.CAD;
- The subject S.INSTALLER shall accept an application package only if it has successfully verified the integrity and authenticity evidences of the application package.

Application Note (ST author):

> The TOE supports the Scenario #2.B push model according to [GP UICC], Section 11.3.1.

FDP_IFF.1.3/SC The TSF shall enforce the following additional information flow control SFP rules: Secure channel behaviour rules for SCP02 defined by GlobalPlatform in [GP], Appendix E.

FDP_IFF.1.4/SC The TSF shall explicitly authorise an information flow based on the following rules: none.

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

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

Application Note:

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

FMT_MSA.1/SC Management of security attributes

FMT_MSA.1.1/SC The TSF shall enforce the **Secure Channel Protocol (SCP) information flow control policy** to restrict the ability to **modify** the security attributes Secure Channel static keys, the Secure Channel security level and the Secure Channel protocol of a Security Domain to an authenticated off-card entity associated with the Security Domain.

Application Note:

> The authorized identified roles could be the card issuer (off-card) or a SD (on-card).

FMT_MSA.3/SC Static attribute initialisation

FMT_MSA.3.1/SC The TSF shall enforce the **Secure Channel Protocol (SCP) information flow control policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/SC The TSF shall allow the following roles: none to specify alternative initial values to override the default values when an object or information is created.

Refinement:

Alternative initial values shall be at least as restrictive as the default values defined in FMT_MSA.3.1.

FMT_SMF.1/SC Specification of Management Functions

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

- **Management functions specified in GlobalPlatform specifications [GP]:**
  - **loading (Section 9.3.5 of [GP]);**
  - **installation (Section 9.3.6 of [GP]);**
  - **extradition (Section 9.4.1 of [GP]);**
  - **registry update (Section 9.4.2 of [GP]);**
  - **SD personalization rules, pull and push models (Section 11 of [GP UICC]).**
  - Modifying the life cycle state of an Application or Security Domain ([GP], Sections 5.3.1, 5.3.2 and 11.10),
  - Setup of initial keys and update of existing keys. ([GP], Section 11.8, [GP UICC], Section 8.7, and [TS102 226], Section 8.2.1.5).

Application Note:

> All management functions related to SCP02 secure channel are relevant.

Application Note (ST author):

> The SET STATUS command is used to modify the life cycle state of an Application or Security Domain.
>
> The PUT KEY command is used to update Secure Channel keys, DAP Verification keys and Delegated Management keys.

FIA_UID.1/SC Timing of identification

FIA_UID.1.1/SC The TSF shall allow

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

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

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

Application Note:

> The GlobalPlatform TSF mediated actions listed in [GP2.2] such as selecting an application, requestion data, initializing, etc.

FIA_UAU.1/SC Timing of authentication

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

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

FIA_UAU.4/SC Single-use authentication mechanisms

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

7.2 Security Assurance Requirements

The Evaluation Assurance Level is EAL4 augmented with ALC_DVS.2 and AVA_VAN.5.

7.3 Security Requirements Rationale

The reader is referred to the Security Requirements Rational in the Protection Profiles:

- [PPJCSv3.0], Section 7.3.1 and 7.3.2,
- [PP(U)SIM], Section 5.3.1 and 5.3.2.

Those parts of the rationale which deviate from the rationale in the PPs are included below and the differences are underlined for easier comparison.

7.3.1 Objectives

The rationale for the following objective has been adjusted:

In the rationale of O.SID, O.FIREWALL, O.RESOURCES, O.ALARM, O.LOAD and O.INSTALL all occurances of the SFRs (1) are replaced by the SFRs (2):

(1) Original SFR	(2) Covered by SFR
FDP_IFC.2/CM	FDP_IFC.2/SC
FDP_IFF.1/CM	FDP_IFF.1/SC
FDP_UIT.1/CM	FDP_UIT.1/SC
FIA_UID.1/CM	FIA_UID.1/SC
FMT_MSA.1/CM	FMT_MSA.1/SC
FMT_MSA.3/CM	FMT_MSA.3/SC
FMT_SMF.1/CM	FMT_SMF.1/SC
FMT_SMR.1/CM	FMT_SMR.1/Installer
FTP_ITC.1/CM	FTP_ITC.1/SC
FDP_ITC.2/Installer	FDP_ITC.2/CCM
FPT_FLS.1.1/Installer	FPT_FLS.1.1/CCM

The column (2) SFR either extends the corresponding column (1) SFR or it addresses exactly the same TSF.

O.CIPHER This security objective is directly covered by FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4, FCS_COP.1 and FCS_RNG.1. The SFR FPR_UNO.1 contributes in covering this security objective and controls the observation of the cryptographic operations which may be used to disclose the keys.

The rationale for the following objectives for the SCP is included here, since they were moved from the environment to the TOE boundaries.

O.SCP.RECOVERY This objective is covered by FPT_FLS.1 and FAU_ARP.1. FPT_FLS.1 states that the TOE shall preserve a secure state in those cases defined in FAU_ARP.1, one of which refers to card tearing and power failure.

O.SCP.SUPPORT This objective is covered as follows: Non-bypassability by FDP_SDI.2 (because data are secured against modification), low-level-cryptographic support by FCS_COP.1 and low-level transaction mechanism by FDP_ROL.1/FIREWALL (because it makes the operation OP.JAVA atomic). Non-bypassability and memory domain separation shall be investigated in ADV_ARC as of CC version 3.

O.SCP.IC This objective is covered by FPT_PHP.3 (resistance against physical attacks).

7.3.2 Rationale Tables of Security Objectives and SFRs

This ST includes the rationale tables Table 7 and Table 8 from [PP(U)SIM], Section 5.3.2, and [PPJCSv3.0], Section 7.3.2. It extends these rationale tables by the following the following tables Table 16 and Table 17. FCS_COP.1 includes all iterations of FCS_COP.1.1 in Section 7.1.1.1.2 and FCS_COP.1.1/DAP in Section 7.1.2.1.1.

Security Objectives	Security Functional Requirements	Rationale
O.SCP.RECOVERY	FPT_FLS.1, FAU_ARP.1	Section 7.3.1
O.SCP.SUPPORT	FDP_SDI.2, FDP_ROL.1/FIREWALL, FCS_COP.1.1, FCS_COP.1.1/DAP	Section 7.3.1
O.SCP.IC	FPT_PHP.3	Section 7.3.1
O.CIPHER	FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.3,FCS_CKM.4, FCS_COP.1, FPR_UNO.1, FCS_RNG.1	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1
O.SID	FIA_ATD.1/AID, FIA_UID.2/AID, FMT_MSA.1/JCRE, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.1/SC, FMT_MSA.3/SC, FDP_ITC.2/CCM, FMT_SMF.1/SC, FMT_SMF.1/ADEL, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FIA_USB.1/AID, FMT_MSA.1/JCVM, FMT_MSA.3/JCVM	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1
O.FIREWALL	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FMT_SMR.1/Installer, FMT_MSA.1/SC, FMT_MSA.3/SC, FMT_MSA.3/FIREWALL, FMT_SMR.1, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL, FMT_MSA.1/JCRE, FDP_ITC.2/CCM, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FMT_SMF.1/ADEL, FMT_SMF.1/SC, FMT_SMF.1, FMT_MSA.2/FIREWALL_JCVM, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_MSA.1/JCVM, FMT_MSA.3/JCVM	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1
O.OPERATE	FAU_ARP.1, FDP_ROL.1/FIREWALL, FIA_ATD.1/AID, FPT_FLS.1/ADEL, FPT_FLS.1, FPT_FLS.1/ODEL, FPT_FLS.1/CCM, FDP_ITC.2/CCM, FPT_RCV.3/Installer, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FPT_TDC.1, FIA_USB.1/AID	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1
O.RESOURCES	FAU_ARP.1, FDP_ROL.1/FIREWALL, FMT_SMR.1/Installer, FMT_SMR.1, FMT_SMR.1/ADEL, FPT_FLS.1/CCM, FPT_FLS.1/ODEL, FPT_FLS.1, FPT_FLS.1/ADEL, FPT_RCV.3/Installer, FMT_SMR.1/Installer, FMT_SMF.1/ADEL, FMT_SMF.1/SC, FMT_SMF.1, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1
O.ALARM	FPT_FLS.1/CCM, FPT_FLS.1, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL, FAU_ARP.1	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1
O.LOAD	FCO_NRO.2/CM, FDP_IFC.2/SC, FDP_IFF.1/SC, FDP_UIT.1/CCM, FIA_UID.1/SC, FTP_ITC.1/SC	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1
O.INSTALL	FDP_ITC.2/CCM, FPT_RCV.3/Installer, FPT_FLS.1/CCM	[PPJCSv3.0], Section 7.3.3.1; Section 7.3.1

Table 16 SCP objectives / SFR mapping

Security Functional Requirements	Security objectives of the PP	Additional Security Objectives in this ST
FDP_SDI.2	Table 8 in [PPJCSv3.0]	O.SCP.SUPPORT
FDP_ROL.1/FIREWALL		
FCS_COP.1		
FMT_SMR.1/Installer	Table 8 in [PPJCSv3.0]	O.RESOURCES
FDP_IFC.2/SC	Table 8 in [PP(U)SIM]	O.LOAD
FDP_IFF.1/SC		O.LOAD
FDP_UIT.1/CCM		O.LOAD

FIA_UID.1/SC		O.LOAD
FMT_MSA.1/SC		O.SID, O.FIREWALL
FMT_MSA.3/SC		O.SID, O.FIREWALL
FMT_SMF.1/SC		O.SID, O.FIREWALL, O.RESOURCES
FTP_ITC.1/SC		O.LOAD
FDP_ITC.2/CCM		O.FIREWALL, O.OPERATE, O.INSTALL
FPT_FLS.1/CCM		O.OPERATE, O.RESOURCES, O.ALARM, O.INSTALL
FPT_FLS.1	Table 8 in [PPJCSv3.0]	O.SCP.RECOVERY
FAU_ARP.1		
FPT_PHP.3	-	O.SCP.IC
FCS_RNG.1	Table 7 in [PPJCSv3.0]	O.CIPHER

Table 17 SFRs and Security Objectives

7.3.3 Dependencies

The reader is referred to the relevant chapters in the Protection Profiles:

- [PPJCSv3.0], Sections 7.3.2 and 7.3.3,
- [PP(U)SIM], Sections 5.3.2 and 5.3.3.

Those parts that deviate from the Dependencies in the PPs are included below and the differences are underlined for easier comparison.

7.3.3.1 SFRs and SARs Dependencies

This ST includes the dependencies tables Table 9 and Table 10 from [PP(U)SIM], Section 5.3.3, and [PPJCSv3.0], Section 7.3.3. The SFRs Dependencies tables are extended by the following the following Table 18. The SARs Dependencies tables are not extended.

Security Functional Requirement	Dependencies	Satisfied Dependencies
FCS_PHP.3	No Dependencies	-
FCS_RNG.1	No Dependencies	-

Table 18 SFRs Dependencies

7.3.4 Security Assurance Requirements Rationale

See Section 5.3.4 in [PP(U)SIM].

7.3.5 AVA_VAN.5 Advanced methodical vulnerability analysis

See Section 5.3.5 in [PP(U)SIM].

7.3.6 ALC_DVS.2 Sufficiency of security measures

See Section 5.3.6 in [PP(U)SIM].

8 TOE summary specification

8.1 TOE Security Functions

8.1.1 SF_TRANSACTION

This security function provides atomic transactions according to the Java Card Transaction and Atomicity mechanism with commit and rollback capability ([JCRE], Section 7) for updating persistent objects in flash memory. The update operation either successfully completes or the data is restored to its original pre-transaction state if the transaction does not complete normally. The TransactionException is thrown if the commit capacity is exceeded during a transaction. The rollback operation restores the original values of the persistent objects and clears the dedicated transaction area. The TOE permits rollback of any access in the sense of [JCRE], Section 6.2.8, and creation of objects via the JCAPI new or makeTransient calls. Context switches do not alter the state of a transaction in progress.

8.1.2 SF_ACCESS_CONTROL

This security function is in charge of the FIREWALL access control SFP and the JCVM information flow control SFP. Based on security attributes [Sharing, Context, Lifetime], it enforces applet isolation by means of the JCRE firewall ([JCRE], Section 6.1), controls the access to global data containers shared by all applet instances and controls object access across contexts ([JCRE], Section 6.2).

The Firewall access control policy and the JCVM information flow control policy are enforced at runtime.

The JCRE allocates and manages a context for each Java API package containing applets. The JCRE maintains its own context as a special system privilege so that it can perform operations that are denied to contexts of applets.

To grant for the FIREWALL access control SFP and the JCVM information flow control SFP:

1. The TOE enforces the Firewall access control SFP and the JCVM information flow policy to control the flow of information between subjects. It maintains the roles Java Card VM for enforcing the applet firewall, and Java Card RE for allocating and managing contexts for packages containing applets and its own Java Card RE context.
2. The TOE restricts the ability to modify the list of registered applets and packages to the Java Card RE (S.JCRE) and maintains the following list of security attributes for individual users: the AID and version number of each package, the AID and version number of each registered applet, and whether a registered applet is currently selected.
3. The TOE requires each user to be identified before allowing any TSF-mediated actions on behalf of that user. Subjects acting on behalf of a specific user are identified by the package AID.

4. The TOE accepts only secure values for security attributes of subjects and objects defined in the Firewall access control SFP and the JCVM information flow control SFP.
5. The ability to modify the Currently Active Context and the Active Applets is restricted to the Java Card VM (S.JCVM). The ability to modify the Selected Applet Context is restricted to the Java Card RE (S.JCRE).
6. The TOE provides Inter-TSF data consistency. The TOE uses rules defined in [JCVM] specification (given in FPT_TDC.1.2) when interpreting the TSF data from another trusted IT product.

8.1.3 SF_CRYPTO

This security function controls all the operations related to the cryptographic key management and cryptographic operations.

1. Key generation refers to the generation of a cryptographic key or key pair to be used in cryptographic algorithms. The algorithms supported by the TOE that require a secret or private key are RSA-CRT, RSA, Triple-DES and AES. Key generation involves generation of a secret value that is used as a secret key for a symmetric algorithm (AES or Triple-DES), or a prime generation seed for RSA.
2. The random number generator provided by the TOE complies with [ISO/IEC 18031]. Besides of its use in key generation, applications may use the methods of the Java Card API javacard.security.RandomData class for generation of random numbers.
3. The TOE provides key destruction for Triple-DES, AES, RSA and RSA-CRT keys by the following means:
   - Applications may use the Java Card API method Key.clearKey() for key destruction.
   - An authenticated off-card entity may use the PUT KEY command within a Secure Channel Session to zeroize the DAP key(s) and the Delegated Management Token and Receipt keys.
   - All keys (and the Global PIN) are zeroized by setting the Issuer Security Domain life cycle state to TERMINATED. An authenticated off-card entity may use the SET STATUS command for this purpose.
   - The TOE zeroizes the session keys when closing the corresponding Secure Channel Session or upon card reset.
   - In order to delete the DAP Verification key the Security Domain containing this key must be deleted. This operation deletes all keys contained in that Security Domain.
4. Key distribution is provided via the Java Card API set methods of javacard.security classes AESKey, DESKey, RSAPrivateCrtKey, RSAPrivateKey and RSAPublicKey.
5. Key access is provided via the Java Card API get methods of classes AESKey, DESKey, RSAPrivateCrtKey, RSAPrivateKey and RSAPublicKey.
6. The TOE provides cryptographic key operations and hashing operations by the following means:
   - Encryption and decryption with Triple-DES and AES in CBC and ECB modes, RSA and RSA-CRT is provided via the Java Card API methods. AES is implemented according to [FIPS 197], Triple-DES according to [SP800-67], the CBC and ECB modes of operation according to [SP800-38A], and the RSA cipher according to

[PKCS1]. Triple-DES is used for Secure Channel and sensitive data encryption and decryption according to [PKCS1], E.4.6 and E.4.7.

- Digital signature generation and verification using RSA and RSA-CRT is provided to applications via the Java Card API methods defined in the javacard.security.Signature class. The implementation of the algorithm is according to [PKCS1] for RSASSA-PKCS1-v1_5 and RSASSA-PSS.
- Applications may use the methods of the Java Card API class javacard.security.MessageDigest for hashing with SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512. A Security Domain uses SHA-1 for Load File Data Block Hash generation. The hash algorithms are implemented according to [FIPS 180-4].
- MAC generation with Triple-DES in CBC mode and AES MAC is supported via the Java Card API javacard.security.Signature. The full Triple-DES MAC generation according to [ISO/IEC 9797-1] is used for authentication cryptogram generation and verification for the GlobalPlatform Secure Channel Protocol ([GP2.2.1], E.4.2, B.1.2.1).

7. The TOE supports the DES Scheme for DAP Verification according to [GP2.2.1], C.6.2. The Load File Data Block Signature is a Retail MAC of the Load File Data Block Hash generated according to [GP2.2.1], B.1.2.2.

8.1.4 SF_INTEGRITY

This security function provides a means to check the integrity of checksum-protected data stored in flash memory. This mechanism initializes the checksum of cryptographic keys, PIN values and their associated security attributes.

The TOE monitors cryptographic keys, PIN values and their associated security attributes stored within the TSF for integrity errors by checksum verification. Upon detection of a data integrity error on cryptographic keys, PIN values and their associated security attributes the TOE will throw an exception and prevent the usage of the affected key or PIN or switch to an endless loop. This is a secure state.

8.1.5 SF_SECURITY

This security function ensures a secure state of information, the non-observability of operations on the information and the unavailability of previous information content upon deallocation/allocation.

The TSF ensures resistance to physical tampering using features against probing and an active shield detecting integrity violation.

Sensitive data are locked upon the following operations as defined in [JCRE]:

- Deletion of package and/or applications,
- Deletion of objects.

The sensitive temporary buffers (transient object, bArray object, APDU buffer, and cryptographic buffer) are securely cleared after their usage with respect to their life-cycle and interface as defined in [JCRE]. Transient objects and persistent objects are erased at the allocation for a new object.

1. The TOE throws an exception, locks the card session or reinitializes the Java Card System and its JCRE data upon detection of a potential security violation and preserves a secure state. Security violations result in an immediate reset. The JCRE context is the currently active context after card reset.

2. The TOE ensures that an attacker is unable to observe cryptographic operations / comparison operations on key values / PIN values.
3. The TOE ensures that any previous information content of a resource is made unavailable upon deallocation of the resource from the bArray object, any reference to an object instance created during an aborted transaction and the cryptographic buffer. Upon allocation of the APDU buffer any previous information content is made unavailable. Upon deallocation of a resource from the cryptographic buffer (D.CRYPTO) or from any transient object, any previous information content of the resource is made unavailable.
4. The TOE detects physical tampering of the TSF with sensors for operating voltage, clock frequency, temperature and electromagnetic radiation. It is resistant to physical tampering of the TSF. If the TOE detects with the above mentioned sensors that it is not supplied within the specified limits, a security reset is initiated and the TOE is not operable until the supply is back in the specified limits. The design of the hardware protects it against analysing and physical tampering.
5. The TOE hides information about IC power consumptions and command execution time, to ensure that no confidential information can be derived from this data.

8.1.6 SF_CONTENT_MANAGEMENT

Content management is the capability for the loading, installation, extradition, registry update, card content removal and Security Domain personalization. These operations are performed by a privileged Security Domain that applies a secure communication policy. Security Domains are privileged Applications that hold cryptographic keys used to support Secure Channel Protocol operations and/or to authorize card content management functions.

This security function provides the following capabilities:

1. Content changes are permitted according to the privileges that have been assigned to the acting Security Domain and according to the security services and management features it provides.
   The commands for Remote Load File loading, Application installation, Load File removal, Application removal, Application locking/unlocking and Application information retrieval are processed under the control of a Security Domain with card content management capabilities such as the Issuer Security Domain or any Security Domain with Delegated Management privileges or Authorized Management.
   The TOE restricts the ability to modify the Registered Applets and Resident Packages to the JCRE.
   Confidential Security Domain personalization is supported via the STORE DATA command and the DGI to be submitted to the Security Domain with the mandatory data set for CASD and APSD.
2. The TOE provides DAP Verification and Delegated Management as authorization and control features during card content loading and installation. DAP Verification assures that Application code loaded on the card is checked for integrity and authenticity; Delegated Management assures that the off-card entity (Application Provider or Controlling Authority) has been authorized to perfrom a card content management operation. The TOE implements the dedicated flow control and runtime behaviour as described in [GP2.2.1], Section 9.3.5. A Security Domain with Token Verification privilege verifies the

Delegated Management Tokens of Delegated Management functions (loading, installation, extradition, and deletion) to check if an off-card entity has been authorized to perform card content changes. A Security Domain that supports DAP Verification verifies the Load File Data Block Signature during the loading of the Load File and prior to further processing of the Load File Data Block.
The TOE uses the security attributes associated with the loaded packages, installed applets, or imported user data and ensures that they are interpreted as intended. The Installer ensures that package loading is allowed only if for each dependent package its AID attribute is equal to a resident package AID attribute, the major (minor) Version attribute associated to the former is equal (less than or equal) to the major (minor) Version attribute associated to the latter ([JCVM], §4.5.1).

3. The TOE maintains the roles Issuer, Application Provider, Controlling Authority and Verification Authority via dedicated Security Domains. The issuer of the (U)SIM card is the MNO represented by the Issuer Security Domain on the card. The Application Provider is represented by the supplementary Security Domain APSD used to manage confidential loading and personalization of applications. The Controlling Authority is represented by the CASD offering confidential personalization service to authenticated application providers. The Verification Authority is represented by a Security Domain with Mandated DAP Verification privilege. The roles Installer and Applet Deletion Manager support the behaviour specified in [JCRE], Section 11. Content installation comprises the INSTALL [for install] command or is part of the combined load, install and make selectable process that comprises a first INSTALL [for load, install and make selectable] command, one or more LOAD commands and a last INSTALL [for load, install and make selectable] command processed by the Security Domain. Content removal (Executable Load File and/or Application removal) comprises the DELETE command processed by the receiving Security Domain.
4. The Applet Deletion Manager is a function of the Java Card RE. The Applet Deletion Manager is responsible for safe deletion of applets (single applet instance or multiple applet instances) and packages (applet/library package or applet package and contained instances). Applet instance deletion involves the removal of the applet object instance and the objects owned by the applet instance and associated Java Card RE structures. Applet/library package deletion involves the removal of all the card resident components of the CAP file, including code and any associated Java Card RE management structures. Deletion of the applet package and the contained applet instances involves the removal of the card-resident code and Java Card RE structures associated with the applet package, and all the applet instances and objects in the context of the package and associated JCRE structures.
5. Card content management operations may be cleanly aborted when the card is reset or when power is removed from the card, when another applet is selected on this logical channel or in case of insufficient flash memory, failure in cryptographic safeguarding, or upon package reference (versions) mismatch. The TOE preserves a secure state or provides the ability to return to a secure state particularly in the following cases:
   - When the card content management operation loading, installation, or deletion fails.
   - When automated recovery from power loss is not possible.
   - When the object deletion functions fail to delete all the unreferenced objects owned by the applet that requested the execution of the method.

Upon installation failure the TOE performs a rollback and ensures that objects created during the execution of the install method can never be accessed by any applet on the card. In particular, any reference in CLEAR_ON_RESET transient space to an object created during an unsuccessful applet installation will be reset as a null reference ([JCRE], Section 11.1.5).

6. The TOE ensures that any previous information content of a resource is made unavailable upon the deallocation of the resource from applet instances and/or packages and from the objects owned by the context of an applet instance which triggered the execution of the method javacard.framework.JCSystem.requestObjectDeletion() or if deletion operations performed by the Applet Deletion Manager occur.

8.1.7 SF_SECURE_CHANNEL

This security function provides a secure communication channel between a card and an off-card entity during an Application Session.

1. An off-card entity may initiate secure communication with the TOE by the following means: SCP02, SCP80.
   Secure Channels allow the card content to be transmitted to the TOE in a manner protected from modification, deletion, insertion and replay.
   A Secure Channel Session is open after a successful initiation, including the authentication of the off-card entity by the on-card Application. For SCP02 the commands INITIALIZE UPDATE and EXTERNAL AUTHENTICATE are provided by the TOE for this purpose.
   The TOE checks if the off-card entity has been successfully authenticated and a Secure Channel Session initiated before performing content management operations. Application selection, secure channel initiation, request data with the GET DATA command on behalf of the user can be performed before the user is identified and authenticated.
   Applications use the Secure Channel Protocol(s) supported by their associated Security Domain.
2. The Secure Channel Protocol provides mutual authentication, integrity and data origin authentication and confidentiality of transmitted data and application packages. For SCP02 mutual authentication is by means of cryptographic exchange between the card and the off-card entity initiated by the off-card entity; it implies the generation of session keys derived from static key(s) maintained by the Security Domain. Message integrity and data origin authentication is by applying MAC calculation across the header and data field of an APDU command using the generated Secure Channel session MAC key. Confidentiality of message data is assured by encryption using the Secure Channel session ENC key. SCP80 works with pre-shared static keys.
3. The information flow between the communicating entities and command execution may be granted or prohibited based on the kind of command, its parameters and security status information. The security status is determined by the value of the relevant security attributes. A Secure Channel Protocol operates according to the established Security Level. A mandatory minimum Security Level is set at the initialization of the Secure Channel Session. A different Current Security Level may be set for an individual command or response.
4. The TOE prevents reuse of authentication data related to a secure communication channel.

8.2 Fulfilment of the SFRs

SFR	TOE Summary Specification
JCS PP SFRs	
FDP_ACC.2/FIREWALL	SF_ACCESS_CONTROL.1
FDP_ACF.1/FIREWALL	SF_ACCESS_CONTROL.1
FDP_IFC.1/JCVM	SF_ACCESS_CONTROL.1
FDP_IFF.1/JCVM	SF_ACCESS_CONTROL.1
FMT_MSA.1/JCRE	SF_ACCESS_CONTROL.5
FMT_MSA.1/JCVM	SF_ACCESS_CONTROL.5
FMT_MSA.2/FIREWALL_JCVM	SF_ACCESS_CONTROL.4
FMT_MSA.3/FIREWALL	SF_ACCESS_CONTROL.4
FMT_MSA.3/JCVM	SF_ACCESS_CONTROL.4
FMT_SMF.1	SF_ACCESS_CONTROL.5
FMT_SMR.1	SF_ACCESS_CONTROL.1
FCS_COP.1.1	SF_CRYPTO.6
FCS_CKM.1	SF_CRYPTO.1, 2
FCS_CKM.2	SF_CRYPTO.4
FCS_CKM.3	SF_CRYPTO.5
FCS_CKM.4	SF_CRYPTO.3
FDP_RIP.1/OBJECTS	SF_SECURITY.3
FDP_RIP.1.1/ABORT	SF_SECURITY.3
FDP_RIP.1.1/APDU	SF_SECURITY.3
FDP_RIP.1.1/bArray	SF_SECURITY.3
FDP_RIP.1.1/KEYS	SF_SECURITY.3
FDP_RIP.1.1/TRANSIENT	SF_SECURITY.3
FDP_ROL.1/FIREWALL	SF_TRANSACTION
FAU_ARP.1	SF_SECURITY.1
FDP_SDI.2	SF_INTEGRITY
FPR_UNO.1	SF_SECURITY.2
FPT_FLS.1	SF_SECURITY.1
FPT_TDC.1	SF_ACCESS_CONTROL.6
FIA_ATD.1/AID	SF_ACCESS_CONTROL.2
FIA_UID.2/AID	SF_ACCESS_CONTROL.3
FIA_USB.1/AID	SF_ACCESS_CONTROL.2, 3, 4
FMT_MTD.1/JCRE	SF_ACCESS_CONTROL.2
FMT_MTD.3/JCRE	SF_ACCESS_CONTROL.4
FMT_SMR.1/Installer	SF_CONTENT_MANAGEMENT.3
FPT_RCV.3/Installer	SF_CONTENT_MANAGEMENT.5
FDP_ACC.2/ADEL	SF_CONTENT_MANAGEMENT.4
FDP_ACF.1/ADEL	SF_CONTENT_MANAGEMENT.4
FDP_RIP.1/ADEL	SF_CONTENT_MANAGEMENT.6
FMT_MSA.1/ADEL	SF_CONTENT_MANAGEMENT.1
FMT_MSA.3/ADEL	SF_CONTENT_MANAGEMENT.4
FMT_SMF.1/ADEL	SF_CONTENT_MANAGEMENT.1
FMT_SMR.1/ADEL	SF_CONTENT_MANAGEMENT.3
FPT_FLS.1/ADEL	SF_CONTENT_MANAGEMENT.5
FDP_RIP.1/ODEL	SF_CONTENT_MANAGEMENT.6
FPT_FLS.1/ODEL	SF_CONTENT_MANAGEMENT.5

FCO_NRO.2/CM	SF_SECURE_CHANNEL.2
(U)SIM PP Basic TOE SFRs	
FDP_UIT.1/CCM	SF_CONTENT_MANAGEMENT.2 SF_SECURE_CHANNEL.1
FDP_ROL.1/CCM	SF_CONTENT_MANAGEMENT.5
FDP_ITC.2/CCM	SF_CONTENT_MANAGEMENT.2
FPT_FLS.1/CCM	SF_CONTENT_MANAGEMENT.5
FCS_COP.1/DAP	SF_CRYPTO.7
FDP_ACC.1/SD	SF_CONTENT_MANAGEMENT.2
FDP_ACF.1/SD	SF_CONTENT_MANAGEMENT.2
FMT_MSA.1/SD	SF_SECURE_CHANNEL.3
FMT_MSA.3/SD	SF_SECURE_CHANNEL.3
FMT_SMF.1/SD	SF_CONTENT_MANAGEMENT.1
FMT_SMR.1/SD	SF_CONTENT_MANAGEMENT.3
FTP_ITC.1/SC	SF_SECURE_CHANNEL.2
FCO_NRO.2/SC	SF_CONTENT_MANAGEMENT.2
FDP_IFC.2/SC	SF_SECURE_CHANNEL.3
FDP_IFF.1/SC	SF_SECURE_CHANNEL.3
FMT_MSA.1/SC	SF_SECURE_CHANNEL.3
FMT_MSA.3/SC	SF_SECURE_CHANNEL.3
FMT_SMF.1/SC	SF_SECURE_CHANNEL.1, 3
FIA_UID.1/SC	SF_SECURE_CHANNEL.1
FIA_UAU.1/SC	SF_SECURE_CHANNEL.1
FIA_UAU.4/SC	SF_SECURE_CHANNEL.4
Additional SFRs	
FPT_PHP.3	SF_SECURITY.4, 5
FCS_RNG.1	SF_CRYPTO.2

Table 19 Mapping of SFRs to mechanisms of TOE

9 Assurance Measures

This chapter describes the Assurance Measures fulfilling the requirements listed in chapter 7.2.

The following table lists the Assurance measures and references the corresponding documents describing the measures.

Assurance Measures	Description
AM_ADV	The representing of the TSF is described in the documentation for functional specification, in the documentation for TOE design, in the security architecture description and in the documentation for implementation representation.
AM_AGD	The guidance documentation is described in the operational user guidance documentation and in the documentation for preparative procedures.
AM_ALC	The life cycle support of the TOE during its development and maintenance is described in the life cycle documentation including configuration management, delivery procedures, development security as well as development tools.
AM_ATE	The testing of the TOE is described in the test documentation.
AM_AVA	The vulnerability assessment for the TOE is described in the vulnerability analysis documentation.

Table 20 References of Assurance measures

10 Glossary and Acronyms

10.1 Glossary

Term	Definition
AID	See [PPJCSv3.0], Appendix 4: Glossary
APDU	
APDU buffer	
Applet	
Applet deletion manager	
BCV	
CAD	
CAP file	
Class	
Context	
Current context	
Currently selected applet	
Default applet	
DPA	
Embedded Software	
Firewall	
Installer	
Interface	
Java Card RE	
Java Card RE Entry Point	
Java Card RMI	
Java Card System	
Java Card VM	
Logical channel	
NVRAM	
Object deletion	
Package	
PCD	
PICC	
RAM	
SCP	
Shareable interface	
SIO	
Subject	
SWP	
Transient object	
User	
ITSEF	See [PP(U)SIM], Annex A.1 Definitions
(U)SIM Java Card Platform developer	
Application developer	
Certification body	
Issuer	
Mobile operator	
Smart Card IC Provider	
Smart Card manufacturer	
Smart Card personalizer	
TOE issuer	
Verification Authority	

Term	Definition
Validation laboratory	
Controlling Authority	
Composite-ST	The Composite Security Target. This document.
DESFire	MIFARE® DESFire® EV1 library of the ST33G1M2 platform, part of the TOE of the Platform-ST.
IC Dedicated Software	IC proprietary software embedded in the Security IC and developed by STMicroelectronics.
Platform-ST	The hardware platform Security Target: [ST33-ST]
Security IC Embedded Software	Software embedded in a Security IC and developed by G&D. The Security IC Embedded Software is designed in Phase 1 and embedded into the Security IC in Phase 5 of the Security IC product life-cycle.

10.2 Acronyms

Acronym	Term
3GPP	3rd Generation Partnership Project
ACP	Access Control Policy
ADELG	Applet Deletion Group
AID	Applet IDentifier
AM	Authorized Management
AP	Application Provider
APDU	Application Protocol Data Unit
API	Application Programming Interface
APSD	Application Provider Security Domain
BIP	Bearer Independent Protocol
CA	Controlling Authority
CAD	Card Acceptance Device
CASD	Controlling Authority Security Domain
CC	Common Criteria
CEM	Common Evaluation Methodology
CMAC	Cipher-based MAC (acc. to NIST SP 800-38)
CRS	Contactless Registry Services (acc. to [GP CS])
EAL	Evaluation Assurance Level
GP	GlobalPlatform
GSM	Global System for Mobile communications
HW/SW/FW	Hardware/Software/Firmware
IC	Integrated Circuit
JCRE	Java Card Runtime Environment
JCS	Java Card System
JCS PP	Java Card™ System Protection Profile
JCVM	Java Card Virtual Machine
IMSI	International Mobile Subscriber Identity
ISD	Issuer Security Domain
ITSEF	Information Technology Security Evaluation Facility
MNO	Mobile Network Operator
n.a.	not applicable
OS	Operating System
OSP	Organizational Security Policy
OTA	Over The Air
PP	Protection Profile
RAM	Remote Application Management
RMI	Remote Method Invocation
SAR	Security Assurance Requirement
SCP	Smart Card Platform

Acronym	Term
	Secure Channel Protocol
SCWS	Smart Card Web Server
SD	Security Domain
SF	Security Function
SFR	Security Functional Requirement
SIM	Subscriber Identity Module
SPD	Security Problem Definition
SSD	Supplementary Security Domain
ST	Security Target
TOE	Target Of Evaluation
TSF	TOE Security Functions
UMTS	Universal Mobile Telecommunications System
USIM	Universal Subscriber Identity Module
(U)SIM PP	(U)SIM Java Card™ Platform Protection Profile
VA	Verification Authority

11 Bibliography

[AGD_PRE] SkySIM CX Hercules V2.0 Preparative Procedures, Version 1.2 / Status 2014-12-17, Giesecke & Devrient GmbH

[AGD_OPE] SkySIM CX Hercules V2.0 Operational Guidance Common Document, Version 1.1 / Status 2014-12-17, Giesecke & Devrient GmbH
SkySIM CX Hercules V2.0 Operational Guidance for the Application Developer, Version 1.1 / Status 2014-12-16, Giesecke & Devrient GmbH
SkySIM CX Hercules V2.0 Operational Guidance for the Application Provider, Version 1.1 / Status 2014-12-16, Giesecke & Devrient GmbH
SkySIM CX Hercules V2.0 Operational Guidance for the Controlling Authority, Version 1.0 / Status 2014-12-12, Giesecke & Devrient GmbH
SkySIM CX Hercules V2.0 Operational Guidance for the Mobile Network Operator, Version 1.1 / Status 2014-12-16, Giesecke & Devrient GmbH
SkySIM CX Hercules V2.0 Operational Guidance for Personaliser, Version 1.0 / Status 2014-12-12, Giesecke & Devrient GmbH
SkySIM CX Hercules V2.0 Operational Guidance for the Terminal, Version 1.0 / Status 2014-12-12, Giesecke & Devrient GmbH
SkySIM CX Hercules V2.0 Operational Guidance for the Verification Authority, Version 1.1 / Status 2014-12-16, Giesecke & Devrient GmbH

[AIS20/AIS31] A proposal for: Functionality classes for random number generators, Version 2.0, 18 September 2011

[CC P1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012

[CC P2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012

[CC P3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

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

[FIPS 46-3] National Institute of Standards and Technology, Data Encryption Standard, Federal Information Processing Standards Publication FIPS PUB 46-3, October 1999

[FIPS 180-4] National Institute of Standards and Technology, Secure Hash Standard, Federal Information Processing Standards Publication FIPS PUB 180-4, March 2012

[FIPS 186-4] Federal Information Processing Standards Publication FIPS PUB 186-4 DIGITAL SIGNATURE STANDARD (DSS) (with Change Notice), U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology, July 2013

[FIPS 197] Federal Information Processing Standards Publication 197, ADVANCED ENCRYPTION STANDARD (AES), U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology, November 26, 2001

[GP], [GP2.2] GlobalPlatform Card Specification, Version 2.2, March 2006

[GP2.2.1] GlobalPlatform Card Specification, Version 2.2.1, January 2011

[GP CCM] GlobalPlatform, Card Confidential Card Content Management, Card specification v2.2 – Amendment A, Version 1.0, October 2007

[GP CS] GlobalPlatform, Contactless Services – Card Specification v2.2 – Amendment C, Version 1.0, February 2010

[GP UICC] GlobalPlatform Card, UICC Configuration, Version 1.0.1, January 2011

[ISO/IEC 9797-1] ISO/IEC 9797-1: Information technology – Security techniques –Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, 1999

[ISO/IEC 18031] ISO/IEC 18031: Information technology — Security techniques — Random bit generation, 2011

[JCAPI] Java Card Platform Classic Edition, Version 3.0.4, Application Programming Interface, Sun Microsystems, Inc., September 2011.

[JCRE] Java Card Platform Classic Edition, Version 3.0.4, Runtime Environment Specification, Sun Microsystems, Inc., September 2011

[JCVM] Java Card Platform Classic Edition, Version 3.0.4, Virtual Machine Specification, Sun Microsystems, Inc., September 2011

[JIL] Certification of “open” smart card products, Version 1.1 (for trial use), 4 February 2013, Joint Interpretation Library

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

[PKCS1] PKCS #1: RSA Encryption Standard – An RSA Laboratories Technical Note, Version 1.5, Revised November 1, 1993

[PP0035] Common Criteria Protection Profile, Security IC Platform, BSI-PP-0035-2007, Version 1.0, June 2007

[PP(U)SIM] Common Criteria Protection Profile, (U)SIM Java Card Platform, Basic and SCWS Configurations, Evolutive Certification Scheme for (U)SIM cards, Version 2.0.2, June 17, 2010, PU-2009-RT-79

[PPJCSv3.0] Common Criteria Protection Profile Java Card™ System Open Configuration, Version 3.0, May 2012, ANSSI-CC-PP-2010/03

[RFC2104] Network Working Group, HMAC: Keyed-Hashing for Message Authencation, RFC 2104, February 1997

[RFC3447] Network Working Group, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, RFC 3447, February 2003

[ST33-ST] ST33G1M2 Platform Maskset K8H0A version F, with firmware revision 9, optional cryptographic library NESLIB 4.1, and optional technology MIFARE® DESFire® EV1 3.7 and 3.8 – Security Target – Public Version, Rev 02.01, April 2014

[SP800-38A] National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, Special Publication 800-38A, December 2001.

[SP800-38B] National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, Special Publication 800-38B, May 2005.

[SP800-67] National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, Special Publication 800-67, version 1.2, July 2011

[TS102 223] ETSI TS 102 223 V10.6.0 (2012-03), Smart Cards; Card Application Toolkit (CAT) (Release 10)

[TS102 225] ETSI TS 102 225 V10.0.0 (2012-03), Smart Cards; Secured packet structure for UICC based applications (Release 10)

[TS102 226] ETSI TS 102 226 V10.0.0 (2012-03), Smart Cards; Remote APDU structure for UICC based applications (Release 10)

[TS102 241] ETSI TS 102 241 V9.2.0 (2012-03), Smart Cards; UICC Application Programming Interface (UICC API) for Java Card™ (Release 9)

[TS131 111] ETSI TS 131 111, Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) (Release 6)

[TS131 130] ETSI TS 131 130 V11.0.0 (2013-04), Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; (U)SIM Application Programming Interface (API); (U)SIM API for Java™ Card (3GPP TS 31.130 version 11.0.0 Release 11)