



Sm@rtSIM CX Virgo V5.0 Security Target Lite

Version 4.4/2017-07-18



R&D

Author Giesecke+Devrient Mobile Security GmbH
Status Draft
File: ASE_ST_Lite_SmartSIM_CX_Virgo_V5c.docx

Giesecke+Devrient Mobile Security GmbH
Prinzregentenstraße 159
D-81677 München

© Copyright 2017
Giesecke+Devrient Mobile Security GmbH
Prinzregentenstraße 159
D-81677 München

This document as well as the information or material contained is copyrighted. Any use not explicitly permitted by copyright law requires prior consent of Giesecke+Devrient Mobile Security GmbH. This applies to any reproduction, revision, translation, storage on microfilm as well as its import and processing in electronic systems, in particular.

The information or material contained in this document is property of Giesecke+Devrient Mobile Security GmbH and any recipient of this document shall not disclose or divulge, directly or indirectly, this document or the information or material contained herein without the prior written consent of Giesecke+Devrient Mobile Security GmbH.

All copyrights, trademarks, patents and other rights in connection herewith are expressly reserved to Giesecke+Devrient Mobile Security GmbH and no license is created hereby.

All brand or product names mentioned are trademarks or registered trademarks of their respective holders.

Contents

Contents 3

History of this document 6

1 ST Introduction 7

1.1 ST Reference 7

1.2 ST Overview 7

1.3 Typographic Conventions 8

1.4 TOE Overview 8

 1.4.1 TOE Type 8

 1.4.2 TOE Description 9

 1.4.3 TOE Usage 10

 1.4.4 TOE Life Cycle 10

 1.4.5 Actors of the TOE 13

 1.4.6 Usage of the TOE 13

 1.4.7 TOE Security Features 13

 1.4.8 Non-TOE HW/SW/FW Available to the TOE 15

 1.4.9 TOE exclusions 15

2 Conformance Claim 16

2.1 CC Conformance Claim 16

2.2 PP Claim 16

2.3 Package Claim 16

2.4 Conformance Claims Rationale 16

3 Security Aspects 18

4 Security Problem Definition 19

4.1 Assets 19

4.2 Subjects 21

 4.2.1 S.CRS_USER 21

 4.2.2 S.CRS 21

4.3 Threats 22

4.4 Organisational Security Policies 24

4.5 Assumptions 24

5 Security Objectives 26

5.1 Security Objectives for the TOE	26
5.1.1 IDENTIFICATION	27
5.1.2 EXECUTION	27
5.1.3 SERVICES	28
5.1.4 OBJECT DELETION	29
5.1.5 APPLET MANAGEMENT	29
5.1.6 Smart Card Platform	30
5.1.7 Card Management	31
5.1.8 CRS	32
5.2 Security Objectives for the Operational Environment	32
5.2.1 OEs from [PPJCSv3.0]	34
5.2.2 Actors	35
5.2.3 Secure places	36
5.2.4 Policies	36
5.2.5 Platform	36
5.2.6 GlobalPlatform	36
5.2.7 Applications	37
5.3 Security Objectives Rationale	37
5.3.1 Threats	37
5.3.2 Organisational Security Policies	45
5.3.3 Assumptions	45
5.3.4 SPD and Security Objectives	45
6 Security Requirements	53
6.1 Security Functional Requirements for the TOE	53
6.1.1 Java Card System SFRs	58
6.1.2 Basic TOE configuration SFRs	80
6.1.3 Additional SFRs	88
6.2 Security Assurance Requirements	96
6.3 Security Requirements Rationale	96
6.3.1 Objectives	96
6.3.2 Rationale Tables of Security Objectives and SFRs	97
6.3.3 Dependencies	100
6.3.4 Security Assurance Requirements Rationale	101
6.3.5 AVA_VAN.5 Advanced methodical vulnerability analysis	101
6.3.6 ALC_DVS.2 Sufficiency of security measures	101
7 TOE summary specification	102
7.1 TOE Security Functions	102
7.1.1 SF_TRANSACTION	102
7.1.2 SF_ACCESS_CONTROL	102
7.1.3 SF_CRYPTO	103
7.1.4 SF_INTEGRITY	104
7.1.5 SF_SECURITY	104

7.1.6 SF_CONTENT_MANAGEMENT..... 105

7.1.7 SF_SECURE_CHANNEL 107

7.1.8 SF_APPLET_MIGRATION 108

7.1.9 SF_OS_UPDATE 108

7.1.10 SF_CRS..... 109

7.2 Fulfilment of the SFRs..... 110

8 Statement of Compatibility..... 112

8.1 Matching statement 112

8.2 Security Environment..... 112

8.3 Security requirements..... 114

8.4 Assurance requirements 115

8.5 Statement on contradictions..... 115

9 Assurance Measures..... 116

10 Glossary and Acronyms..... 117

10.1 Glossary..... 117

10.2 Acronyms 118

11 Bibliography..... 120

History of this document

4.4	2017-07-18	stut	Final Version
-----	------------	------	---------------

1 ST Introduction

1.1 ST Reference

Title: Security Target Lite Sm@rtSIM CX Virgo V5.0

Reference: Security Target Lite Sm@rtSIM CX Virgo V5.0, Version 4.4/2017-07-18

Author: Giesecke+Devrient Mobile Security GmbH

TOE name: Sm@rtSIM CX Virgo V5.0

TOE guidance: [AGD_PRE], [AGD_OPE]

TOE hardware:

BCM_SPS02 C0 (Certificate: BSI-DSC-CC-0915-2016-MA-03) [BCM-ST] and [BCM-Cert]

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 Sm@rtSIM CX Virgo V5.0.

Chapter 1 is an introduction to this document.

Chapter 2 includes the conformance claims.

Chapter 3 lists the security aspects as defined in the [PPJCSv3.0].

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 7 contains the TOE Summary Specification.

Chapter 8 contains the statement of compatibility between this Composite Security Target (Composite-ST) and the Platform Security Target (Platform-ST, [BCM-ST]).

Chapter 9 describes the Assurance Measures fulfilling the requirements listed in chapter 6.2.

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 an embedded Secure Element (eSE) intended to be soldered in a mobile phone or other mobile devices. 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

Sm@rtSIM CX Virgo V5.0 is the Target of Evaluation (TOE) of this Security Target. The TOE is a composite product that comprises:

- 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.
- The SCP (Smart Card Platform) comprises the IC (Integrated Circuit) and the OS (Operating System).

The TOE boundaries are indicated in Figure 1 by the purple dashed line. The TOE is an embedded Secure Element (eSE).

The number bullets in Figure 1 below demonstrate the correspondence of the TOE and non-TOE elements in the two PPs, to which this ST claims conformance to or is based upon (see chapter 2.2).

	TOE of the PP	Sm@rtSIM CX Virgo TOE
①	The Smart Card Platform (SCP) is a combination of the Security Integrated Circuit (IC) and the native Operating System (OS).	BCM_SPS02 C0 and Sm@rtSIM CX Virgo OS
②	Java Card System (JCRE, JCVM, JCAPI)	Java Card Platform 3.0.4 classic implementation
③	Additional native code, proprietary applications	Native Applications
④	Applets	The TOE includes the CRS applet

Table 1 Correspondance of TOE building blocks in PP and ST

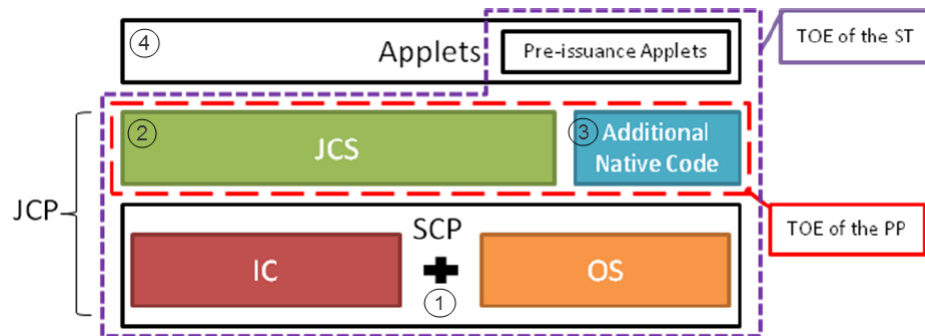


Figure 1 Sm@rtSIM CX Virgo V5.0 TOE boundaries as defined in [PPJCSv3.0], Figure 1

1.4.2 TOE Description

Sm@rtSIM CX Virgo V5.0 is an embedded Secure Element (eSE).

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 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 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 and SCP03).
- Support for contactless cards (ATQ, different implicit selection on different interfaces and channels)
- FeliCa Routing Handler
- Post-issuance personalization of Security Domain according to [GP UICC]
- Application personalization according to [GP UICC]
- GP CASD scenario #1 and #3 according to [GP CS]
- GP CRS based on [GP CCM]

The TOE supports the single wire protocol (SWP).

The TOE supports the OS Update functionality of the underlying secure chip platform (SCP), BCM_SPS02 C0 (Certificate: BSI-DSC--CC-0915-2016) [BCM-ST] and Applet Migration functionality.

The TOE includes the FeliCa Crypto Library.

The TOE supports CASD with scenario #1 with ECC-256.

Application Note (ST author):

For applets that opt-in to the Applet Migration functionality, during the OS Update process, the Applet Migration functionality executed in the JCRE context within the TOE is responsible for the enforcement of the applicable SFRs of the JavaCard Firewall. This Applet Migration functionality is evaluated to maintain the overall security level of the TOE, which remains consistent across the operational phase including Applet Migration.

1.4.3 TOE Usage

The main usage of the TOE is as an embedded Secure Element (eSE) being the basis for applications such as contactless payment and transit to be stored and executed.

An embedded Secure Element (eSE) is a tamper resistant component which is used in a device to provide the security, confidentiality, and multiple application environments required to support various business models.

1.4.4 TOE Life Cycle

The TOE life cycle follows the description in the [PPJCSv3.0]. The TOE life cycle phases are detailed in Figure 2. We refer to [PP0084] for a thorough description of Phases 1 to 7:

- Phases 1 and 2 comprise the product development: Embedded Software (IC Dedicated Software, OS, Java Card System, 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 (composite product integration).
- Phase 6 is dedicated to the product personalisation prior final use.
- Phase 7 is the product operational phase.

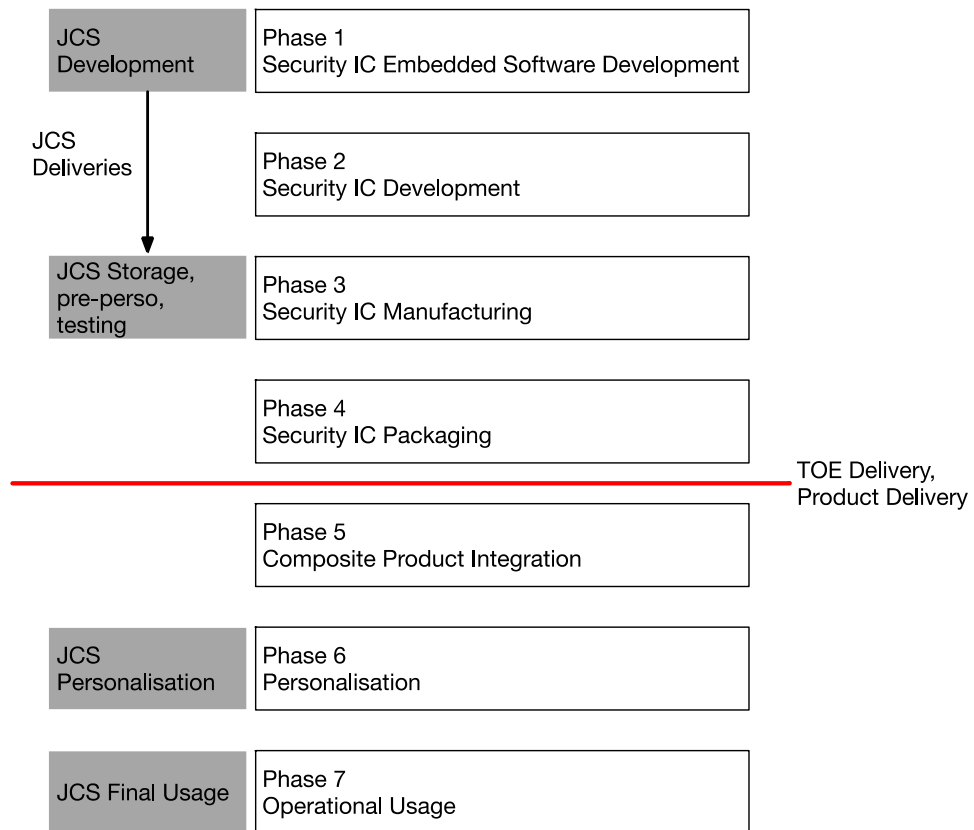


Figure 2 TOE Life Cycle

Phase 1

The Embedded Software Development is performed during Phase 1. This includes Java Card System (JCS) 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 TOE development environment, in particular those locations where the TOE is accessible for installation or testing.

Phase 2

The IC Development phase is as described in [BCM-ST].

Phase 3

The IC Manufacturing phase is as described in [BCM-ST].

In this phase, JCS Deliveries and the Loader Keys (required to Load the Embedded Software on the IC) are delivered by the Embedded Software Developer to the IC Manufacturer, who is responsible for integrating them on the ICs during Manufacturing.

Also in this phase, the Embedded Software is securely transferred from the Software Developer to the IC Developer in the form of an encrypted image, together with all pre-personalisation data that would be needed to complete the integration, storage, pre-personalisation and test.

The evaluation includes the delivery process. Delivery and acceptance procedures guarantee the authenticity, the confidentiality and integrity of the exchanged pieces. The TOE delivery involves encrypted and signed exchanges with previous exchange of public keys.

The TOE is obtained by the embedding of this image on ICs. The Manufacturing environment protects the confidentiality and integrity of the Security IC Embedded Software and any related material.

At the end of Phase 3, wafers or sawn wafers are delivered for packaging.

Phase 4

The IC Packaging phase is as described in [BCM-ST].

The TOE delivery occurs during Phase 4 and the next phases are not covered by ALC.

Phase 5

The Composite Product Integration is as described in [BCM-ST].

At the end of this phase, the TOE is a loaded eSE, ready to be delivered for further integration within a device, and ready to be personalized.

Phase 6

The TOE, and specifically the IC Embedded Software, is personalized in Phase 6.

Phase 7

During Phase 7, Operational Usage of the TOE is taking place, and the following services are supported:

- Post-issuance applet loading,
- OS Update: to perform an update of the OS with preservation of any existing OS data and User data
- Applet Migration: to perform to replace pre-issuance applets with preservation of their data

Note:

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

1.4.5 Actors of the TOE

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

- 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 Issuer represented on the eSE and responsible for securing the keys creation and personalization of the Application Provider Security Domain (APSD. 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, is endorsed in this document by the Verification Authority, as follows.
- The **Verification Authority (VA)**, trusted third party represented on the eSE, acting on behalf of the Issuer 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 Usage of the TOE

The eSE defined in this ST is a Java Card platform embedded in a module used as data carriers that are secure against forgery and tampering. Data processing is performed by a piece of software embedded in the chip, called an applet.

This TOE is in Open configuration: it contains pre-issuance Java Card applets and also supports applet loading after personalization through the GP layer of the eSE.

This eSE is integrated in a mobile device and its management is under control of the Issuer which holds the Keys required to authenticate with his dedicated GP Security Domain: the ISD. The authentication mechanism is the GP SCP, which allows the Issuer to:

- Load, Install and Manage Applet
- Load and Install Secure Data Files
- Install and Manage Supplementary Security Domains for dedicated applications (APSD)
- Manage the Controlling Authority Security Domain (CASD)

A proprietary mechanism allows the Issuer to block applet loading post-issuance (during phase 7).

1.4.7 TOE Security Features

Java Card applets can be loaded and instantiated onto the TOE either before card issuance or in post-issuance. 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.

1.4.7.1 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.

1.4.7.2 Application Management

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

- The issuer is initially the only entity authorized to manage applications (loading, instantiation, deletion) through a secure communication channel with the TOE, based on SCP02 or SCP03. However, the Issuer can grant these privileges to the AP through the delegated management functionality of GP.
- Before loading, all applications are verified by a Verification Authority and are associated at load time to a Verification Authority (VA) signature (Mandated DAP) that is verified on card by the on-eSE 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 Issuer 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 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.

1.4.8 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
- Contactless Services according to [GP CS]
- Customer-specific plugins and APIs

1.4.9 TOE exclusions

On-card bytecode verifier is out of the scope of this Security Target, see chapter 4.4.2 [PPJCSv3.0].

2 Conformance Claim

2.1 CC Conformance Claim

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

2.2 PP Claim

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

This Security Target is also referring to the (U)SIM Java Card™ Platform Protection Profile- Basic configuration, [PP(U)SIM].

Even when all its SFRs are included in this ST a strict compliance to the [PP(U)SIM] cannot be made since the TOE is an embedded Secure Element (eSE) and not a USIM card.

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 TOE is demonstrable conformant to [PPJCSv3.0], which references Java Card Platform (JCAPI, JCRE, JCVM) specification version 3.0.1.

The ST conformance to [PPJCSv3.0] is demonstrable with respect to the operations related to applet migration (see application note (ST author) in chapter 1.4.2).

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

During the static Applet Migration only the user in charge of the OS update (owner of the hash table signature key) can authorise the Applet Migration process. Therefore the overall security of the TOE is still intact and in accordance to the objectives of the [PPJCSv3.0].

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 SFR has been introduced: FPT_PHP.3.

The following SFRs from [PPJCSv3.0] are not included, since they are covered by the SFRs that were included from the [PP(U)SIM] (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 [PPJCSv3.0] are not included, because they are covered by the card content management SFRs from the [PP(U)SIM]: 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 [PP(U)SIM] either extend the SFRs originating [PPJCSv3.0] or address exactly the same functionality. Exclusion of the [PPJCSv3.0] SFRs is therefore considered as removal of duplicates.

3 Security Aspects

This Security Target includes all Security Aspects as defined in [PPJCSv3.0]. 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	#.INSTALL
#.CONFID-JCS-CODE	#.SID
#.CONFID-JCS-DATA	#.OBJ-DELETION
#.INTEG-APPLI-CODE	#.DELETION
#.INTEG-APPLI-DATA	#.ALARM
#.INTEG-JCS-CODE	#.OPERATE
#.INTEG-JCS-DATA	#.RESOURCES
#.EXE-APPLI-CODE	#.CIPHER
#.EXE-JCS-CODE	#.KEY-MNGT
#.FIREWALL	#.PIN-MNGT
#.NATIVE	#.SCP
#.VERIFICATION	#.TRANSACTION
#.CARD-MANAGEMENT	

4 Security Problem Definition

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

Additions and refinements to the [PPJCSv3.0] have been made in the ST, following the [PP(U)SIM].

The tables included in subsections below summarize the refinements, additions, and equivalences between the [PPJCSv3.0], the [PP(U)SIM] 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 [PPJCSv3.0].
- A: Additional assets, subjects, threats, OSPs, assumptions, objectives or SFRs specified in the [PP(U)SIM] that are independent from the [PPJCSv3.0] and do not affect it.
- R: Refinement made in [PP(U)SIM] or in this ST. [PP(U)SIM] assets, threats, OSPs, assumptions or objectives are more restrictive than the ones required in [PPJCSv3.0] SPD.
- : The asset, threat, OSP, objective, OE, assumption, or SFR is not included in the designated PP or in this ST.

4.1 Assets

Table 2 indicates the assets included in this Security Target. All assets defined in the Protection Profile ([PPJCSv3.0], Section 5.1.) are relevant for the TOE of this Security Target.

Additions and Refinements are according to the [PP(U)SIM], Section 3.1.

The definition of assets is not repeated here.

ST TOE Asset	[PPJCSv3.0]	Refinement/Addition [PP(U)SIM]
TSF Data		
D.API_DATA	D	
D.CRYPTO	D	
D.JCS_CODE	D	
D.JCS_DATA	D	
D.SEC_DATA	D	
D.GP_CODE	-	A
D.CARD_MNGT_DATA	-	A
D.UPDATE_STATE	-	
D.UPDATE_KEYS	-	
User Data		
D.APP_CODE	D	
D.APP_C_DATA	D	
D.APP_I_DATA	D	
D.PIN	D	
D.APP_KEYS	D	
D.APSD_KEYS	(D.APP_KEYS)	R
D.CASD_KEYS	(D.APP_KEYS)	R
D.ISD_KEYS	(D.APP_KEYS)	R
D.VASD_KEYS	(D.APP_KEYS)	R
CRS		
D.AUTH_KEYS	(D.APP_KEYS)	
D.AUTH_RANDOM	-	
D.CL_REGISTRY	-	

Table 2 Assets

R: The assets D.APSD_KEYS, D.CASD_KEYS, D.ISD_KEYS, D.VASD_KEYS in the [PP(U)SIM] are refinements of the asset D.APP_KEYS in [PPJCSv3.0].

Two assets specific to this TOE are additional to those defined in the PP [PPJCSv3.0]:

D.UPDATE_STATE

This asset corresponds to an OS internal state describing the stage within the OS Update process. The complex nature of the OS Update mechanism allows multiple sessions, resume and rollback in case needed. This state is the warrant for successful progress and completion.

D.UPDATE_KEYS

This asset corresponds to the Loader Keys that are required to operate the OS Update mechanism. The TOE uses these Keys to decrypt and verify the signature of the image loaded on the eSE during the OS Update (source code and data).

Three assets specific to CRS are additional to those defined in the PP [PPJCSv3.0]:

D.AUTH_KEYS

This asset corresponds to the CRS Authentication Key that is required in the CRS authentication scheme.

D.AUTH_RANDOM

This asset corresponds to the CRS Authorization Random that is required in the CRS authentication scheme.

D.CL REGISTRY

GP CL Registry containing applets information dedicated to contactless: install parameters, CL state, CVM settings, and more.

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) are relevant for the TOE of this Security Target.

Additions and Refinements are according to the [PP(U)SIM], Section 3.2.

The definition of the subjects is not repeated here.

ST TOE Subject	[PPJCSv3.0]	Refinement/Addition [PP(U)SIM]
S.ADEL	D	
S.APPLET	D	
S.BCV	D	
S.CAD	D	
S.INSTALLER	D	
S.JCRE	D	
S.JCVM	D	
S.LOCAL	D	
S.MEMBER	D	
S.PACKAGE	D	
S.SD	-	A
S.CRS_USER	-	-
S.CRS	-	-

Table 3 Subjects

Notes:

- 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.
- S.SD covers ISD, CASD, VASD and SSD.

4.2.1 S.CRS_USER

The subject S.CRS_USER is represented by the Issuer.

4.2.2 S.CRS

The subject S.CRS is represented by the CRS applet.

4.3 Threats

T.OVERWRITE_CVM	-	-
T.OVERWRITE_MFF_MFD	-	-
T.OVERWRITE_CL	-	-
T.AUTH_FORGE	-	-
T.DATA_FORGE	-	-

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).

TOE ST Threat	[PPJCSv3.0]	Refinement/Addition [PP(U)SIM]
T.CONFID-APPLI-DATA	D	
T.UNAUTHORIZED_ACCESS	(shareable interface)	R1
T.CONFID-JCS-CODE	D	
T.CONFID-JCS-DATA	D	
T.INTEG-APPLI-CODE	D	
T.INTEG-APPLI-CODE.LOAD	D	
T.INTEG-APPLI-DATA	D	
T.INTEG-USER-DATA	(T.INTEG-APPLI-DATA)	R2
T.INTEG-APPLI-DATA.LOAD	D	
T.INTEG-JCS-CODE	D	
T.INTEG-JCS-DATA	D	
T.SID.1	D	
T.SID.2	D	
T.EXE-CODE.1	D	
T.EXE-CODE.2	D	
T.NATIVE	D	
T.RESOURCES	D	
T.DELETION	D	
T.INSTALL	D	
T.UNAUTHORIZED_CARD_MNGT	(T.INSTALL)	R3
T.LIFE_CYCLE	-	A
T.OBJ-DELETION	D	
T.PHYSICAL	D	
T.SECURE-DELETION	-	-
CRS		
T.OVERWRITE_CVM	-	-
T.OVERWRITE_MFF_MFD	-	-
T.OVERWRITE_CL	-	-
T.AUTH_FORGE	-	-
T.DATA_FORGE	-	-

Table 4 Threats

R1: T.UNAUTHORIZED_ACCESS is a refinement of [PPJCSv3.0] threat T.CONFID-APPLI-DATA to Shareable Objects.

R2: T.INTEG-USER-DATA is a refinement of [PPJCSv3.0] threat T.INTEG-APPLI-DATA. Applies to [PP(U)SIM] assets.

R3: T.UNAUTHORIZED_CARD_MNGT is a refinement of [PPJCSv3.0] threat T.INSTALL.

The following Threat is additional to those defined in the PP [PPJCSv3.0] in order to cover the threats inherent to applet deletion within the TOE, especially with the applet migration mechanism:

A: T.SECURE_DELETION replaces A.DELETION.

T.SECURE DELETION

The attacker exploits security breaches that are introduced through the deletion of an installed applet in the form of broken references to garbage collected code or data or altered 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 threatened asset(s): D.APP I DATA, D.SEC DATA, D.APP KEY, D.PIN and D.CRYPTO.

The following threats are added to this ST in order to cover the threats inherent to CRS:

T.OVERWRITE CVM

The attacker attempts to change an applet's required CVM.

T.OVERWRITE MFF MFD

The attacker attempts to communicate with an applet that is Marked For Freeze or Marked For Delete.

T.OVERWRITE CL

The attacker attempts to make an applet available on CL without required settings.

T.AUTH FORGE

The attacker attempts to use CRS in order to perform an operation on an applet that is restricted to specific CVM, without successful authentication. For instance, the attacker attempts to forge an authentication in required to enable the applet on CL or to use the applet for processing transaction data.

T.DATA FORGE

The attacker attempts to modify the transaction data after they are received by CRS to corrupt the transaction.

4.4 Organisational Security Policies

The TOE complies with all Organizational Security Policies (OSP) defined in the Protection Profile ([PPJCSv3.0], Section 6.3.2).

Additions and Refinements are according to the [PP(U)SIM], Sections 3.4.

ST TOE Organizational Security Policy	[PPJCSv3.0]	Refinement/Addition [PP(U)SIM]
OSP.VERIFICATION	D	
OSP.SHARE-CONTROL	(OSP.VERIFICATION)	
OSP.AID-MANAGEMENT	(OSP.VERIFICATION)	
OSP.APSD-KEYS	-	A1
OSP.KEY-GENERATION	-	A1
OSP.CASD-KEYS	-	A1
OSP.VASD-KEYS	-	A1
OSP.KEY-CHANGE	-	A1
OSP.SECURITY-DOMAINS	-	A2
OSP.QUOTAS	-	A2

Table 5 Organizational Security Policies

OSP.SHARE-CONTROL, and OSP.AID-MANAGEMENT are refinements of OSP.VERIFICATION in [PPJCSv3.0].

A1: The following OSPs are added to this ST, accordingly with [PP(U)SIM], in order to cover the assets that were added to cover the SD Keys specifically:

- OSP.APSD-KEYS, to cover D.APSD_KEYS
- OSP.CASD-KEYS, to cover D.CASD_KEYS
- OSP.VASD-KEYS, to cover D.VASD_KEYS
- OSP.KEY-GENERATION, to cover the generation of the SDs Keys (D.*SD_KEYS)
- OSP.KEY-CHANGE, to cover the update of the D.APSD_KEYS

A2: The following OSPs are added to this ST, accordingly with [PP(U)SIM], in order to cover the specificity of the Card Manager regarding SD management:

- OSP.SECURITY-DOMAINS, to cover the management of SD management in Phase 7
- OSP.QUOTAS, to cover the memory quota at SD creation

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 are defined in the Protection Profile ([PPJCSv3.0], Section 6.3.3).

Additions and Refinements are according to the [PP(U)SIM], Sections 3.5.

ST TOE Assumption	[PPJCSv3.0]	Refinement/Addition [PP(U)SIM]
A.APPLET	D	
	A.DELETION	
A.VERIFICATION	D	
A.APPS-PROVIDER	-	A1
A.VERIFICATION-AUTHORITY	-	A1
A.CONTROLLING-AUTHORITY	-	A1
A.KEY-ESCROW	-	A1
A.PERSONALIZER	-	A1
A.PRODUCTION	-	A2

Table 6 Assumptions

A.DELETION is not an assumption applicable to this TOE but rather relates to a threat countered by the TOE and defined as T.SECURE_DELETION.

A1: The following Assumptions are added to this ST, accordingly with [PP(U)SIM], in order to cover the assets that were added previously:

- A.APPS-PROVIDER, to cover the trusted actor holding the D.APSD_KEYS
- A.VERIFICATION-AUTHORITY, to cover the trusted actor holding the D.VASD_KEYS
- A.CONTROLLING-AUTHORITY, to cover the trusted actor holding the D.CASD_KEYS
- A.KEY-ESCROW, to cover the trusted actor in charge of the initial AP Keys
- A.PERSONALIZER, to cover the trusted actor in charge of loading the TOE Keys and holding the D.ISD_KEYS

A2: The following assumption are additional to those defined in the PP [PPJCSv3.0];, accordingly with [PP(U)SIM], in order to cover the TOE until delivery:

- A.PRODUCTION, to cover the TOE environment

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.

The security objectives defined in the Protection Profile ([PPJCSv3.0], Section 6.1) are included in this Security Target, except O.REMOTE that is not supported by this TOE, as indicated in Table 7.

Additions and Refinements are according to the [PP(U)SIM] (Sections 4.1), formulated to comply with the fact that the TOE is an eSE.

ST TOE Objective	[PPJCSv3.0]	Refinement/Addition [PP(U)SIM]
O.SID	D	
O.FIREWALL	D	
O.GLOBAL-ARRAYS-CONFID	D	
O.GLOBAL-ARRAYS-INTEG	D	
O.NATIVE	D	
O.OPERATE	D	
O.REALLOCATION	D	
O.RESOURCES	D	
O.ALARM	D	
O.CIPHER	D	
O.KEY-MNGT	D	
O.PIN-MNGT	D	
-	Ø.REMOTE	
O.TRANSACTION	D	
O.OBJ-DELETION	D	
O.DELETION	D	
O.LOAD	D	
O.INSTALL	D	
O.DOMAIN-RIGHTS	-	A
O.APPLI-AUTH	(O.LOAD)	R2
O.SCP.IC	(OE.SCP.IC)	R3
O.SCP.RECOVERY	(OE.SCP.RECOVERY)	R3
O.SCP.SUPPORT	(OE.SCP.SUPPORT)	R3
<u>O.SCP.UPDATE</u>		
<u>O.MIGRATION</u>		
<u>O.CRS_AUTH</u>		
<u>O.CRS_REGISTRY</u>		
<u>O.CRS_TRANSACTION</u>		

Table 7 Security Objectives for the TOE

O.REMOTE applies to RMI services and is not supported by this TOE.

The TOE has been extended to include the SCP mechanism compared to [PPJCSv3.0], and the following Objectives are moved from the TOE Environment to the TOE:

- O.SCP.IC is OE.SCP.IC applied to the TOE
- O.SCP.RECOVERY is OE.SCP.RECOVERY applied to the TOE
- O.SCP.SUPPORT is OE.SCP.SUPPORT applied to the TOE

The TOE has been extended with proprietary mechanisms compared to [PPJCSv3.0], and the following Objectives are additional to those defined in the PP [PPJCSv3.0]:

- O.MIGRATION, to define the applet migration security objectives
- O.SCP.UPDATE, to define the OS update security objectives

The TOE has been extended to include CRS compared to [PPJCSv3.0], and the following Objectives have been added to the ST:

- O.CRS_REGISTRY
- O.CRS_AUTH
- O.CRS_TRANSACTION

R2: O.APPLI-AUTH in the [PP(U)SIM] is a refinement of the objective O.LOAD in [PPJCSv3.0].

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 5.1.8).

A: The following Objective is additional to those defined in the PP [PPJCSv3.0], in order to cover the card content management environment:

- O.DOMAIN-RIGHTS

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.

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 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.

O.MIGRATION

The TOE shall ensure that the applet migration from one applet version to another performs as expected.

The TOE shall ensure the mapping of an applet instance and its data is preserved when the applet is migrated (applet packages are replaced).

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).

O.SCP.UPDATE

The TOE shall enforce the OS Update security policies established by the Issuer during the OS Update mechanism.

For instance, the TOE SCP platform shall ensure that the value of D.UPDATE_STATE is consistent with the OS Update flow. The OS Update flow consists of the following sequential steps and D.UPDATE_STATE transitions:

- 1- Pre-update script: sequence of OS Update-specific APDUs that guide the TOE to prepare the migration of pre-issuance applets and their data (post-issuance applets load packages and instances are deleted)
- 2- (Optional) installation of Migration OS (new SCP) performing direct transformation of data
- 3- Load and installation of new SCP
- 4- Post-update script: sequence of OS Update-specific APDUs to have the new SCP finalize the migration of applets and data
- 5- Activation of the new OS

The TOE CSP Platform shall also verify that the pre-update and post-update scripts are properly signed and encrypted using the D.UPDATE_KEYS, in order to guarantee the integrity and confidentiality of the imported code and data.

5.1.7 Card Management

O.DOMAIN-RIGHTS

The 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 content management shall enforce the application security policies established by the 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 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 a GlobalPlattform 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 CRS

O.CRS REGISTRY

The CRS Module of the TOE shall ensure secure management of the Contactless Registry Services: applet availability over CL, applet authentication restrictions, applet CL lifecycle and the CL application state.

O.CRS AUTH

The CRS Module of the TOE shall enforce all applicable restrictions to process a CRS operation on an applet. This covers the verification that the operation is allowed with the current applet's CL registry information (applet lifecycle marker, CL availability status, CVM restrictions, and Transient overwrite capability). The operations are: authorizing an applet in CL, authorizing a transaction, changing the CVM requirements of an applet, marking an applet (for Freeze or for Delete), and personalize CRS applet.

O.CRS TRANSACTION

The CRS Module of the TOE shall ensure secure handling of an online transaction.

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 Profile ([PPJCSv3.0], Section 6.2)

Additions and refinements are according to [PP(U)SIM], Section 4.2, formulated to comply with the fact that the TOE is an eSE.

Objective for the Operational Environment	[PPJCSv3.0]	Refinement/Addition [PP(U)SIM]
OE.APPLET	D	
OE.CARD-MANAGEMENT	D	
-	OE.SCP.IC	
-	OE.SCP.RECOVERY	
-	OE.SCP.SUPPORT	
OE.VERIFICATION	D	
OE.CODE-EVIDENCE	D	
OE.APPS-PROVIDER	-	A1
OE.VERIFICATION-AUTHORITY	-	A1
OE.KEY-ESCROW	-	A1
OE.PERSONALIZER	-	A1
OE.CONTROLLING-AUTHORITY	-	A1
OE.PRODUCTION	-	A1
OE.AID-MANAGEMENT	(OE.VERIFICATION)	R
OE.AP-KEYS	-	A2
OE.KEY-GENERATION	-	A2
OE.CA-KEYS	-	A2
OE.VA-KEYS	-	A2
OE.KEY-CHANGE	-	A2
OE.SECURITY-DOMAINS	-	A3
OE.QUOTAS	-	A3
OE.SHARE-CONTROL	(OE.VERIFICATION)	R
OE.SECURE-APPS-CERTIFICATION	-	
OE.BASIC-APPS-VALIDATION	-	

Table 8 Objectives for the Operational Environment

The TOE has been extended to include the SCP mechanism compared to the PP [JCSv3.0], and the following Objectives were moved from the TOE Environment to the TOE:

- OE.SCP.IC is now covered by the TOE and defined as O.SCP.IC
- OE.SCP.RECOVERY is now covered by the TOE and defined as O.SCP.RECOVERY
- OE.SCP.SUPPORT is now covered by the TOE and defined as O.SCP.SUPPORT

R: OE.AID-MANAGEMENT and OE.SHARE-CONTROL in the [PP(U)SIM] are refinements of OE.VERIFICATION in the PP [JCSv3.0].

A1: The following Objectives for the Operational Environment are added to this ST, accordingly with [PP(U)SIM], in order to cover the assumptions related to :

- OE.APPS-PROVIDER covers A.APPS-PROVIDER
- OE.VERIFICATION-AUTHORITY covers A. VERIFICATION-AUTHORITY
- OE.KEY-ESCROW covers A.KEY-ESCROW
- OE.PERSONALIZER covers A.PERSONALIZER
- OE.CONTROLLING-AUTHORITY covers A.CONTROLLING-AUTHORITY
- OE.PRODUCTION covers A.PRODUCTION

A2: The following Objectives for the Operational Environment are added to this ST, accordingly with [PP(U)SIM], in order to cover the OSPs related to SD Keys specifically:

- OE.KEY-GENERATION covers the OSP.KEY-GENERATION
- OE.AP-KEYS covers the OSP.APSD-KEYS
- OE.CA-KEYS covers the OSP.CASD-KEYS
- OE.VA-KEYS covers the OSP.VASD-KEYS
- OE.KEY-CHANGE covers the OSP.KEY-CHANGE

A3: The following Objectives for the Operational Environment are added to this ST, accordingly with [PP(U)SIM], in order to cover the specificity of the OSPs related to Card Manager regarding SD management:

- OE.SECURITY-DOMAINS covers the OSP.SECURITY-DOMAINS
- OE.QUOTAS, covers the OSP.QUOTAS

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

5.2.1 OEs from [PPJCSv3.0]

OE.APPLET

No applet loaded post-issuance shall contain native methods.

OE.CARD-MANAGEMENT

The card manager shall control the access to card management functions such as the installation, update or deletion of applets. It shall also implement the card issuer's policy on the card.

The card manager is an application with specific rights, which is responsible for the administration of the smart card. This component will in practice be tightly connected with the TOE, which in turn shall rely on the card manager for the effective enforcing of some of its security functions. Typically the card manager shall be in charge of the life cycle of the whole card, as well as that of the installed applications (applets). The card manager should prevent that card content management (updating, installation, deletion) is carried out, for instance, at invalid states of the card or by non-authorized actors. It shall also enforce security policies established by the card issuer.

OE.VERIFICATION

All the bytecodes shall be verified at least once, before the updating, 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-eSE 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-eSE, technical measures must ensure that the verification required in OE.VERIFICATION are performed. On-card bytocode verifier is out of the scope of this Security Target.

Application Note:

For application code loaded post-issuance and verified off-eSE, 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.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:

- 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

5.2.4.1 Validation and Certification

OE.AID-MANAGEMENT

The VA or the Issuer 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.

5.2.4.2 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.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 to storage in the eSE.

OE.VA-KEYS

The security domain keys of the VA must be securely generated prior to storage in the eSE.

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 Rationale of the Protection Profiles:

- [PPJCSv3.0], Section 6.3.1
- [PP(U)SIM], Sections 4.3.1.1, 4.3.2.1, 4.3.3 and 4.3.4

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

- [PPJCSv3.0], Section 6.3.1
- [PP(U)SIM], Section 4.3.1.1.

With the following changes:

- 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 ([PP(U)SIM])

- 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.

- the additional threats and Security Objectives related to CRS (see section 5.3.1.9)

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 OE.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.

The objective O.MIGRATION contributes to cover this threat by ensuring that the Applet Migration mechanism preserves the confidentiality of all the data of an applet during this applet's migration.

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, 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 OE.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 O.SCP.UPDATE contributes to cover this threat by ensuring that the OS Update mechanism preserves the confidentiality of the JCS code during the importation of the new code and during the replacement of the previous code with the new code.

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 OE.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.

The objective O.SCP.UPDATE contributes to cover this threat by ensuring that the OS Update mechanism preserves the confidentiality of the JCS data during the importation of the new data and during the replacement of the previous data with the new data.

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 OE.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.

The objective O.MIGRATION contributes to cover this threat by ensuring that the Applet Migration mechanism preserves the integrity of the code of an applet during this applet's migration.

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 OE.CARD-MANAGEMENT contributes to cover this threat.

The objective O.MIGRATION contributes to cover this threat by ensuring that the Applet Migration mechanism preserves the integrity of the application code loaded for an applet's migration.

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 OE.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.

The objective O.MIGRATION contributes to cover this threat by ensuring that the Applet Migration mechanism preserves the integrity of all the data of an applet during this applet's migration.

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 OE.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 OE.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 objective O.SCP.UPDATE contributes to cover this threat by ensuring that the OS Update mechanism ensures the integrity of the JCS code during the importation of the new code.

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 OE.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 OE.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. 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. 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 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 resource management (O.RESOURCES) for runtime purposes and good working order (O.OPERATE) in a general manner.

Consumption of resources during installation and other card management operations are covered, in case of failure, by O.INSTALL.

It should be noticed that, for what relates to CPU usage, the Java Card platform is single threaded and it is possible for an ill-formed application (either native or not) to monopolize the CPU. However, a smart card can be physically interrupted (card removal or hardware reset) and most CADs implement a timeout policy that prevent them from being blocked should a card fails to answer. That point is out of scope of this 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 OE.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 OE.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.UNAUTHORIZED_CARD_MNGT

This threat is covered by the following security objectives:

- OE.CARD-MANAGEMENT controls the access to card management functions such as the loading, installation, extradition or deletion of applets.
- 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:

- OE.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.

5.3.1.9 CRS Threats

T.OVERWRITE_CVM

This threat is covered by O.CRS_REGISTRY: CRS ensures that authentication restriction (CVM) of CL applets is securely managed.

T.OVERWRITE_MFF_MFD

This threat is covered by O.CRS_REGISTRY: CRS ensures that CL lifecycle marker of CL applets is securely managed.

T.OVERWRITE_CL

This threat is covered by O.CRS_REGISTRY: CRS ensures that CL availability status of CL applets is securely managed.

T.AUTH_FORGE

This threat is covered by O.CRS_AUTH: CRS enforces applet's CRS registry restrictions before allowing the processing a CRS operation. This includes the validation of the CVM restrictions.

T.DATA_FORGE

This threat is covered by O.CRS_TRANSACTION: CRS enforces protection of the online transaction data.

5.3.2 Organisational Security Policies

This Security Target includes the OSPs rationale from:

- [PPJCSv3.0], section 6.3.2

- [PP(U)SIM], Section 4.3.2.1.

The coverage of OSPs is adjusted in Table 11.

5.3.3 Assumptions

This Security Target includes the Assumptions rationale from:

- [PPJCSv3.0], section 6.3.1

- [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 Security Target includes the SPDs and Security Objectives rationale from:

- [PPJCSv3.0], section 6.3.4

- [PP(U)SIM], Section 4.3.4

The rationale is 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	<u>O.SCP.RECOVERY</u> , <u>O.SCP.SUPPORT</u> , OE.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, O.MIGRATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.CONFID-JCS-CODE	OE.VERIFICATION, OE.CARD-MANAGEMENT, O.NATIVE, <u>O.SCP.UPDATE</u>	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.CONFID-JCS-DATA	<u>O.SCP.RECOVERY</u> , <u>O.SCP.SUPPORT</u> , OE.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.ALARM, O.SCP.UPDATE	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.INTEG-APPLI-CODE	OE.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE, O.MIGRATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.INTEG-APPLI-CODE.LOAD	O.LOAD, OE.CARD-MANAGEMENT, OE.CODE-EVIDENCE, O.MIGRATION	[PPJCSv3.0], Section 6.3.1
T.INTEG-APPLI-DATA	<u>O.SCP.RECOVERY</u> , <u>O.SCP.SUPPORT</u> , OE.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, O.MIGRATION	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.INTEG-APPLI-DATA.LOAD	O.LOAD, OE.CARD-MANAGEMENT, OE.CODE-EVIDENCE	[PPJCSv3.0], Section 6.3.1
T.INTEG-JCS-CODE	OE.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE, OE.CODE-EVIDENCE, <u>O.SCP.UPDATE</u>	[PPJCSv3.0], Section 6.3.1
T.INTEG-JCS-DATA	<u>O.SCP.RECOVERY</u> , <u>O.SCP.SUPPORT</u> , OE.CARD-MANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.ALARM, OE.CODE-EVIDENCE	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.SID.1	OE.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	<u>O.SCP.RECOVERY</u> , <u>O.SCP.SUPPORT</u> , 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	[PPJCSv3.0], Section 6.3.1
T.EXE-CODE.2	OE.VERIFICATION	[PPJCSv3.0], Section 6.3.1
T.NATIVE	OE.VERIFICATION, OE.APPLLET, O.NATIVE	[PPJCSv3.0], Section 6.3.1
T.RESOURCES	O.INSTALL, O.OPERATE, O.RESOURCES, <u>O.SCP.RECOVERY</u> , <u>O.SCP.SUPPORT</u>	Section 5.3.1; [PPJCSv3.0], Section 6.3.1
T.DELETION	O.DELETION, OE.CARD-MANAGEMENT	[PPJCSv3.0], Section 6.3.1
T.SECURE_DELETION	O.DELETION	Section 5.3.1.6
T.INSTALL	O.INSTALL, O.LOAD, OE.CARD- MANAGEMENT	[PPJCSv3.0], Section 6.3.1
T.OBJ-DELETION	O.OBJ-DELETION	[PPJCSv3.0], Section 6.3.1

T.PHYSICAL	<u>O.SCP.IC</u> , <u>O.SCP.SUPPORT</u>	Section 5.3.1; [PPJCSv3.0], Section 6.3.1; [PP(U)SIM], Section 4.3.1
T.INTEG-USER-DATA	O.DOMAIN-RIGHTS, <u>O.SCP-SUPPORT</u> , OE.CA-KEYS, OE.AP-KEYS, OE.VA-KEYS, <u>OE.CODE-EVIDENCE</u>	Section 5.3.1; [PPJCSv3.0], Section 6.3.1; [PP(U)SIM], Section 4.3.1
T.UNAUTHORIZED_CARD-MNGT	OE.CARD-MANAGEMENT, O.APPLI-AUTH, O.DOMAIN-RIGHTS	[PP(U)SIM], Section 4.3.1
T.LIFE_CYCLE	OE.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
T.OVERWRITE_CVM	O.CRS_REGISTRY	Section 5.3.1;
T.OVERWRITE_MFF_MFD	O.CRS_REGISTRY	Section 5.3.1;
T.OVERWRITE_CL	O.CRS_REGISTRY	Section 5.3.1;
T.AUTH_FORGE	O.CRS_AUTH	Section 5.3.1;
T.DATA_FORGE	O.CRS_TRANSACTION	Section 5.3.1;

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
O.MIGRATION	<u>T.CONFID-APPLI-DATA, T.INTEG-APPLI-CODE, T.INTEG-APPLI-CODE.LOAD, T.INTEG-APPLI-DATA</u>
O.SCP.IC	T.PHYSICAL
<u>O.SCP.RECOVERY</u>	T.CONFID-APPLI-DATA, T.CONFID-JCS-DATA, T.INTEG-APPLI-DATA, T.INTEG-JCS-DATA, T.SID.2, T.RESOURCES
<u>O.SCP.SUPPORT</u>	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.SCP.UPDATE	<u>T.CONFID-JCS-CODE, T.CONFID-JCS- DATA, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA</u>
OE.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
<u>O.CRS AUTH</u>	<u>T.OVERWRITE CVM, T.OVERWRITE MFF MFD, T.OVERWRITE CL</u>
<u>O.CRS REGISTRY</u>	<u>T.AUTH FORGE</u>
<u>O.CRS TRANSACTION</u>	<u>T.TRANSACTION FORGE</u>
OE.APPLIET	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.INTEG-JCS-CODE, T.INTEG-JCS-DATA, <u>T.INTEG-USER-DATA</u>
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.INTEG-APPLI-CODE, T.INTEG-APPLI-DATA, T.INTEG-JCS-CODE, T.INTEG-JCS-DATA
OE.AID-MANAGEMENT	
OE.AP-KEYS	T.INTEG-USER-DATA
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, <u>OE.BASIC-APPS-VALIDATION</u>	[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.APSD-KEYS	OE.AP-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	
O.MIGRATION	
<u>O.SCP.IC</u>	
<u>O.SCP.RECOVERY</u>	
<u>O.SCP.SUPPORT</u>	
O.SCP.UPDATE	
OE.CARD-MANAGEMENT	
O.DOMAIN-RIGHTS	
O.APPLI-AUTH	
<u>O.CRS_REGISTRY</u>	
<u>O.CRS_AUTH</u>	
<u>O.CRS_TRANSACTION</u>	
O.INPUT-VALIDATION	
O.REPLAY	
OE.APPLIET	
OE.VERIFICATION	OSP.VERIFICATION
OE.CODE-EVIDENCE	OSP.VERIFICATION
OE.APPS-PROVIDER	
OE.VERIFICATION-AUTHORITY	
OE.KEY-ESCROW	
OE.PERSONALIZER	
OE.CONTROLLING-AUTHORITY	
OE.PRODUCTION	
OE.AID-MANAGEMENT	OSP.AID-MANAGEMENT
OE.AP-KEYS	OSP.APSD-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 – Coverage

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.APPLLET	OE.APPLLET	[PPJCSv3.0], Section 6.3.3
A.VERIFICATION	OE.VERIFICATION, OE.CODE-EVIDENCE	[PPJCSv3.0], Section 6.3.3
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.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.AP-KEYS	
OE.KEY-GENERATION	
OE.CA-KEYS	
OE.VA-KEYS	
OE.KEY-CHANGE	
OE.SECURITY-DOMAINS	
OE.QUOTAS	
OE.SHARE-CONTROL	
OE.CARD-MANAGEMENT	

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

6 Security Requirements

6.1 Security Functional Requirements for the TOE

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

Some SFRs of [PPJCSv3.0] are refined by [PP(U)SIM] and the coverage is expressed in the following table.

This Security Target adds the following SFR in addition to those defined in the Protection Profiles:

- FPT_PHP.3
- FDP_ACC.2/MIGRATION
- FDP_ACF.1/MIGRATION
- FDP_UIT.1/MIGRATION
- FMT_MSA.1/MIGRATION
- FPT_FLS.1/MIGRATION
- FDP_ACC.2/UPDATE
- FDP_ACF.1/UPDATE
- FDP_UIT.1/UPDATE
- FPT_FLS.1/UPDATE
- FDP_ACC.2.CRS_AUTH
- FDP_ACC.2/CRS_PERSO
- FDP_ACC.2/CRS_CVM
- FDP_ACF.1/CRS_AUTH
- FDP_ACF.1/CRS_CVM
- FDP_ETC.1/CRS
- FDP_IFC.1/CRS
- FDP_IFF.1/CRS
- FDP_UCT.1/CRS
- FDP_UIT.1/CRS
- FMT_MSA.1/CRS
- FMT_MSA.4/CRS

ST TOE SFR	[PPJCSv3.0]	Refinements/Additions [PP(U)SIM] Basic TOE
CoreG_LC		
FDP_ACC.2.1/FIREWALL	D	
FDP_ACC.2.2/FIREWALL	D	
FDP_ACF.1.1/FIREWALL	D	
FDP_ACF.1.2/FIREWALL	D	
FDP_ACF.1.3/FIREWALL	D	
FDP_ACF.1.4/FIREWALL	D	
FDP_IFC.1.1/JCVM	D	
FDP_IFF.1.1/JCVM	D	
FDP_IFF.1.2/JCVM	D	
FDP_IFF.1.3/JCVM	D	
FDP_IFF.1.4/JCVM	D	
FDP_IFF.1.5/JCVM	D	
FDP_RIP.1.1/OBJECTS	D	
FMT_MSA.1.1/JCRE	D	
FMT_MSA.1.1/JCVM	D	
FMT_MSA.2.1/FIREWALL_JCVM	D	
FMT_MSA.3.1/FIREWALL	D	
FMT_MSA.3.2/FIREWALL	D	
FMT_MSA.3.1/JCVM	D	
FMT_MSA.3.2/JCVM	D	
FMT_SMF.1.1	D	
FMT_SMR.1.1	D	
FMT_SMR.1.2	D	
FCS_CKM.1.1	D	
FCS_CKM.2.1	D	
FCS_CKM.3.1	D	
FCS_CKM.4.1	D	
FCS_COP.1.1	D	
FDP_RIP.1.1/ABORT	D	
FDP_RIP.1.1/APDU	D	
FDP_RIP.1.1/bArray	D	
FDP_RIP.1.1/KEYS	D	
FDP_RIP.1.1/TRANSIENT	D	
FDP_ROL.1.1/FIREWALL	D	
FDP_ROL.1.2/FIREWALL	D	
FAU_ARP.1.1	D	
FDP_SDI.2.1	D	
FDP_SDI.2.2	D	
FPR_UNO.1.1	D	
FPT_FLS.1.1	D	
FPT_TDC.1.1	D	
FPT_TDC.1.2	D	
FIA_ATD.1.1/AID	D	
FIA_UID.2.1/AID	D	
FIA_USB.1.1/AID	D	
FIA_USB.1.2/AID	D	

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

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

FMT_SMF.1.1/SC	FMT_SMF.1.1/CM	R
FIA_UID.1.1/SC	FIA_UID.1.1/CM	R
FIA_UID.1.2/SC	FIA_UID.1.2/CM	R
FIA_UAU.1.1/SC	-	A
FIA_UAU.1.2/SC	-	A
FIA_UAU.4.1/SC	-	A
SCPG		
FPT_PHP.3.1	-	-
MIGRATION		
FDP_ACC.2.1/MIGRATION	-	-
FDP_ACC.2.2/MIGRATION	-	-
FDP_ACF.1.1/MIGRATION	-	-
FDP_ACF.1.2/MIGRATION	-	-
FDP_UIT.1.1/MIGRATION	-	-
FDP_UIT.1.2/MIGRATION	-	-
FMT_MSA.1.1/MIGRATION	-	-
FPT_FLS.1.1/MIGRATION	-	-
UPDATE		
FDP_ACC.2.1/UPDATE	-	-
FDP_ACC.2.2/UPDATE	-	-
FDP_ACF.1.1/UPDATE	-	-
FDP_ACF.1.2/UPDATE	-	-
FDP_UIT.1.1/UPDATE	-	-
FDP_UIT.1.2/UPDATE	-	-
FPT_FLS.1.1/UPDATE	-	-
CRS		
FDP_ACC.2.1/CRS_AUTH	-	-
FDP_ACC.2.2/CRS_AUTH	-	-
FDP_ACC.2.1/CRS_PERSO	-	-
FDP_ACC.2.2/CRS_PERSO	-	-
FDP_ACC.2.1/CRS_CVM	-	-
FDP_ACC.2.2/CRS_CVM	-	-
FDP_ACF.1.1/CRS_AUTH	-	-
FDP_ACF.1.2/CRS_AUTH	-	-
FDP_ACF.1.1/CRS_CVM	-	-
FDP_ACF.1.2/CRS_CVM	-	-
FDP_ETC.1.1/CRS	-	-
FDP_ETC.1.2/CRS	-	-
FDP_IFC.1.1/CRS	-	-
FDP_IFF.1.1/CRS	-	-
FDP_IFF.1.2/CRS	-	-
FDP_IFF.1.3/CRS	-	-
FDP_IFF.1.4/CRS	-	-
FDP_IFF.1.5/CRS	-	-
FDP_UCT.1.1/CRS	-	-
FDP_UIT.1.1/CRS	-	-
FDP_UIT.1.2/CRS	-	-
FMT_MSA.1.1/CRS	-	-
FMT_MSA.4.1/CRS	-	-

Table 15 Security Functional Requirements

6.1.1 Java Card System SFRs

The [PPJCSv3.0] 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 [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.APPLLET	I.APDU
S.APPLLET	O.CODE_PKG	I.DATA
S.BCV	O.JAVAOBJECT	
S.CAD	O.OS	
S.INSTALLER	<u>O.CRS</u>	
S.JCRE		
S.CJVM		
S.LOCAL		
S.MEMBER		
S.PACKAGE		
<u>S.CRS_USER</u>		
<u>S.CRS</u>		
S.SD		

The Objects O.REMOTE_MTHD, O.REMOTE_OBJ, O.RMI_SERVICE, O.ROR, and the Information I.RORD defined in [PPJCSv3.0] are related to RMI and therefore not relevant for this Security Target.

One Object is additional to those defined in the [PPJCSv3.0]:

O.OS: the embedded software corresponding to the Operating System and comprising the JCS.

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

Security Attribute

Active Applets

Applet Selection Status

Applet's version number	Sharing
Context	Static References
Currently Active Context	<u>MigratableStatus</u>
Dependent package AID	<u>MajorVersion</u>
Identifier	<u>MinorVersion</u>
LC Selection Status	<u>CRS_Default Mark</u>
LifeTime	<u>CRS_Applet Mark</u>
Owner	<u>CRS_Applet CL Authorized</u>
Package AID	<u>CRS_Applet CVM</u>
Registered Applets	<u>CRS_Applet CVM Override</u>
Resident Packages	<u>CRS_Applet CMV Required</u>
Selected Applet Context	<u>CRS_CL Enabled</u>

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

The following Security attributes are additional to those defined in the PP [PPJCSv3.0]:

Migratable Status: Security Attribute held by an O.APPLET to distinguish applets that support the Applet Migration mechanism from the others.

Major version: Security Attribute held by O.OS to determine the minor version (x in case of a version x.y).

Minor Version: Security Attribute held by O.OS to determine the minor version (y in case of a version x.y).

CRS Default Mark: Security attribute of CRS which value, "MarkForDelete" or "MarkForFreeze", identifies the default Applet Mark to be applied on installed applets upon re-personalization of CRS.

CRS Applet Mark: security attribute of an applet which value, "none" or "MarkedForFreeze" or "MarkForDelete", indicates the current CRS lifecycle marking applied to the applet.

CRS Applet CL Authorized: security attribute of an applet which value, "enabled" or "disabled", indicates whether the applet is currently authorized on CL by CRS.

CRS Applet CVM: security attribute of an applet which value identifies the minimum requirements for User Authentication required by the applet before it can be enabled in CL or before it can process a transaction.

CRS Applet CVM Override: security attribute of an applet which value, "enabled" or "disabled", indicates if the User can override the applet's CVM requirements (i.e. changing the value of CRS Applet CVM Required attributes).

CRS Applet CMV Required: security status of an applet which value, "enforce" or "ignore", indicates if the CVM restrictions are enforced or ignored.

The [PPJCSv3.0] 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)
OP.MIGRATE_APPLET(O.APPLET)
OP.OS_UPDATE

OP.CRS_PERSO
OP.CRS_MARK_APPLET(O.APPLET)
OP.CRS_CVM_MANAGE(O.APPLET)
OP.CRS_CVM_OVERRIDE(O.APPLET)
OP.CRS_AUTHORIZE_CL(O.APPLET)
OP.CRS_AUTHORIZE_TRANSACTION(O.APPLET)

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

Eight operations are additional to those defined in the PP [PPJCSv3.0]:

OP.MIGRATE_APPLET: a designated applet is going to be replaced by a new version according to the Applet Migration mechanism.

OP.OS_UPDATE: the OS is going to be updated according to the OS Update mechanism.

OP.CRS_PERSO: to update the CRS data including D.AUTH_KEYS and D.AUTH_RANDOM.

OP.CRS_MARK_APPLET: to set an applet's CRS lifecycle marker to "MarkForFreeze" or "MarkForDelete".

OP.CRS_CVM_MANAGE: to enable or disable the overriding of an applet's CVM by setting its security attribute CVM Override accordingly.

OP.CRS CVM OVERRIDE: to change the setting of an applet to enforce or ignore its CVM restrictions.

OP.CRS AUTHORIZE CL: to make an applet's available in CL.

OP.CRS AUTHORIZE TRANSACTION: to have an applet process an online transaction.

6.1.1.1 CoreG_LC Security Functional Requirements

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

6.1.1.1.1 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.

6.1.1.1.2 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 and [SP800-108]
/RSA	RSA key generation	512-2048 bits	[FIPS 186-4] Section 5.1 and B.3.3
/ECC	EC key generation	192-512 bits	[ANSI X9.63-2011]

Application Note:

The asymmetric keys are generated and diversified in accordance with [JC-API] 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 ([JC-API]).

Application Note (ST author):

For generation of RSA, Triple-DES and AES keys the TOE uses the random number generator from BCM_SPS02 C0 [BCM-ST].

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
/ECC	set methods in [JCAPI] javacard.security classes ECPrivateKey, ECPublicKey	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
/ECC	get methods in [JCAPI] javacard.security classes ECPrivateKey, ECPublicKey	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
/SIG_ECC		ECDSA signature generation and verification with Elliptic Curve Digital Signature Algorithmn (ECDSA)	192-512 bits	[ANSI X9.62-2005], section 7.3
/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)
/CIPH_ECC		ECDH	192-512 bits	[ANSI X9.63-2011]

Application Note:

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

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).

6.1.1.1.3 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-eSE.

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.

6.1.1.1.4 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.

6.1.1.2 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 6.1.2.1.1.

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 6.1.2.1.1.

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 [PP0084]) 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.

6.1.1.3 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_PKG_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.)

6.1.1.4 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).

6.1.1.5 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 6.1.2.1.3.

FDP_IFF.1/CM SFRs are covered by FDP_IFF.1/SC SFRs, see section 6.1.2.1.3.

FDP_UIT.1/CM SFRs are covered by FDP_UIT.1/CCM SFRs, see section 6.1.2.1.1.

FIA_UID.1/CM SFRs are covered by FIA_UID.1/SC SFRs, see section 6.1.2.1.3.

FMT_MSA.1/CM is covered by FMT_MSA.1/SC, see section 6.1.2.1.3.

FMT_MSA.3/CM SFRs are covered by FMT_MSA.3/SC SFRs, see section 6.1.2.1.3.

FMT_SMF.1/CM is covered by FMT_SMF.1/SC, see section 6.1.2.1.3.

FMT_SMR.1/CM is covered by FMT_SMR.1/Installer, see section 6.1.1.2.

FTP_ITC.1/CM SFRs are covered by FTP_ITC.1/SC SFRs, see section 6.1.2.1.3.

6.1.1.6 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.

6.1.2 Basic TOE configuration SFRs

This section describes the SFRs for the Basic TOE configuration.

6.1.2.1 Card Manager (CMGRG)

This section contains the security requirements for the card manager.

6.1.2.1.1 Card Content Management (CCM)

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 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 ([JCVML], §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 11.1.5¹**.

FCS_COP.1/DAP Cryptographic operation

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.

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 PKC Scheme and the DES-Scheme for DAP Verification.

6.1.2.1.2 Security Domain (SD)

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:
 - 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]);
 - S.ADEL, also defined in [PPJCSv3.0] and represented by the GlobalPlatform Environment (OPEN) on the eSE;
 - 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-eSE 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.

6.1.2.1.3 Secure Channel (SC)

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 personalization²;
- 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).

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.

² SD personalization is realized by the TOE via CASD.

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 eSE 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-eSE 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 personalization³.
 - initial/static keys, with the attributes key identifier and key version.

³ SD personalization is realized by the TOE via CASD.

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 (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.

FDP_IFF.1.3/SC The TSF shall enforce the following additional information flow control SFP rules: Secure channel behaviour rules for SCP02 and SCP03 defined by GlobalPlatform in [GP], Appendix E and D.

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-eSE and the off-eSE 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-eSE entity associated with the Security Domain.

Application Note:

..... The authorized identified roles could be the Issuer (off-eSE) or a SD (on-eSE).

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 (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 and SCP03 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 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.**

6.1.3 Additional SFRs

This section contains the SFRs for three additional features supported by the TOE:

- Applet Migration
- OS Update
- CRS

6.1.3.1 Applet Migration

FDP_ACC.2/MIGRATION Complete access control

FDP_ACC.2.1/MIGRATION The TSF shall enforce **the Migration access control SFP** on **O.JAVAOBJECT** and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- OP.MIGRATE_APPLET.

FDP_ACC.2.2/MIGRATION 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/MIGRATION Security attribute based access control

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

Object	Security Attributes
O.APPLET	Migratable Status, Security Domain

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

- **OP.MIGRATE_APPLET is restricted to O.APPLETS with Security Attribute Migratable Status indicating they can be migrated**

- OP.MIGRATE_APPLET requires that user data is protected by a signed hash

FDP_UIT.1/MIGRATION Data exchange integrity

FDP_UIT.1.1/MIGRATION The TSF shall enforce the **Migration access control SFP** to receive user data in a manner protected from **modification, deletion, and insertion** errors.

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

FMT_MSA.1/MIGRATION Management of security attributes

FMT_MSA.1.1/MIGRATION The TSF shall enforce the **Migration access control SFP** to restrict the ability to **modify** the security attributes **Context of an O.JAVAOBJECT to SD (on-eSE)**.

Application note:

During the pre-flight phase of the Migration of an applet, the Context of its O.JAVAOBJECTS is transitioned to its context (Security Domain where the applet instance was created). During the post-flight phase of the Migration of an applet, the Context of its O.JAVAOBJECTS is transitioned to the migrated applet (same class AID as original).

FPT_FLS.1/MIGRATION Failure with preservation of secure state

FPT_FLS.1.1/MIGRATION: The TSF shall preserve a secure state when the following types of failures occur: **the applet data exportation phase or the applet data importation phase are interrupted or fail**.

6.1.3.2 OS UPDATE

FDP_ACC.2/UPDATE Complete access control

FDP_ACC.2.1/UPDATE The TSF shall enforce the **OS Update access control SFP** on **O.OS** and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- OP.OS_UPDATE.

FDP_ACC.2.2/UPDATE 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/UPDATE Security attribute based access control

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

Object	Security Attributes
O.OS	Major Version, Minor Version

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

- **OP.OS_UPDATE** is only available to the IC Embedded Software developer through the IC Loader
- **OP.OS_UPDATE** allows upgrading the OS to a higher version (greater Minor version with same Major version, or greater Major version with any Minor version)
- **OP.OS_UPDATE** allows downgrading the OS to a previous Minor Version, but the Major version has to remain unchanged
- **OP.OS_UPDATE** shall verify that incoming data are signed using the **D.UPDATE_KEYS**

Application Note:

The IC Embedded Software Developer, as defined in [BCM-ST], is the user authorized for using the IC Loader, which is the underlying security mechanism of OS update. The **D.UPDATE_KEYS** corresponds to the **ECDSA_verify_Loader** asset in [BCM-ST].

FDP_UIT.1/UPDATE Data exchange integrity

FDP_UIT.1.1/UPDATE The TSF shall enforce the **OS Update access control SFP** to **receive** user data in a manner protected from **modification, deletion, and insertion** errors.

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

Application Note:

These requirements are partly enforced by the IC Loader, which is the underlying security mechanism of OS update. As defined in [BCM-ST], the Loader enforces a

trusted channel between the external user (IC Embedded Software developer) and the TOE, offering guarantees of confidentiality and integrity during data exchange.

FPT_FLS.1/UPDATE Failure with preservation of secure state

FPT_FLS.1.1/UPDATE: The TSF shall preserve a secure state when the following types of failures occur: **the importation of the new image, the OS data migration preparation phase, the OS copy phase or the OS data restoration phase are interrupted or fail.**

Application Note:

The transition between each phases of the OS Update flow is managed with D.UPDATE_STATE. Until completion of a phase, the transition is not complete and a roll-back is performed, preserving the secure state corresponding to the current D.UPDATE_STATE. At completion of a phase, D.UPDATE_STATE is transitioned.

6.1.3.3 CRS

6.1.3.3.1 Class FDP: User data protection

FDP_ACC.2/CRS_AUTH Complete access control

FDP_ACC.2.1/CRS_AUTH The TSF shall enforce the **CRS Authentication access control SFP** on **S.CRS_USER** and all operations among subjects and objects covered by the SFP.

Refinement:

- The operations involved in the policy are:
- OP.CRS_AUTHORIZE_CL
 - OP.CRS_AUTHORIZE_TRANSACTION.

FDP_ACC.2.2/CRS_AUTH 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_ACC.2/CRS_PERSO Complete access control

FDP_ACC.2.1/CRS_PERSO The TSF shall enforce the **Secure Channel Protocol information flow control policy** on **S.SD** and all operations among subjects and objects covered by the SFP.

Refinement:

- The operations involved in the policy are:

- OP.CRS_PERSO when used to update D.AUTH_KEYS,
- OP.CRS_CVM_MANAGE,
- OP.CRS_MARK_APPLET when used.

FDP_ACC.2.2/CRS_PERSO 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_ACC.2/CRS_CVM Complete access control

FDP_ACC.2.1/CRS_CVM The TSF shall enforce the **CRS CVM Override Access Control SFP** on S.CRS_USER and all operations among subjects and objects covered by the SFP.

Refinement:

The operation involved in the policy is:

- OP.CRS_CVM_OVERRIDE.

FDP_ACC.2.2/CRS_CVM 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/CRS_AUTH Security attribute based access control

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

Object	Security Attributes
O.CRS	CRS_Applet CVM Override, CRS_Applet CVM Required, CRS_Applet CVM, CRS_Applet Mark, CRS_Applet CL Authorized

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

- If CRS_Applet Mark is "MarkForFreeze" or "MarkForDelete", then OP.CRS_AUTHORIZE_CL and OP.CRS_AUTHORIZE_TRANSACTION are not permitted, otherwise the other rules apply:
- If CRS_Applet CVM Required is set to "ignore" or if S.CRS_User verified the required CRS_Applet CVM, then OP.CRS_AUTHORIZE_CL and OP.CRS_AUTHORIZE_TRANSACTION are allowed, otherwise the other rules apply:
- If OP.CRS_AUTHORIZE_CL is used to set CRS_Applet CL Authorized to "disable", the operation is allowed, otherwise the other rules apply:
- the Operation shall not be allowed.

Application Note:

The CRS User verification status is part of the input parameters of the operations.

FDP_ACF.1/CRS_CVM Security attribute based access control

FDP_ACF.1.1/CRS_CVM The TSF shall enforce the CRS CVM Override access control SFP to objects based on the following:

Object	Security Attributes
O.CRS	CRS_Applet CVM Override, CRS_Applet CVM Required, CRS_Applet CVM, CRS_Applet Mark

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

- If CRS_Applet Mark is "MarkForFreeze" or "MarkForFreeze", then OP.CRS_CVM_OVERRIDE is not permitted, otherwise the other rule applies:
- If CRS_Applet CVM Override is "disabled", then OP.CRS_CVM_OVERRIDE is not permitted, otherwise the other rule applies:
- OP.CRS_CVM_OVERRIDE is authorized if it is used to set CRS_Applet CVM Required to "enforce", otherwise the other rule applies:
- OP.CRS_CVM_OVERRIDE is authorized if it is used to set CRS_Applet CVM Required to "ignore" and if S.CRS_User verified the required CRS_Applet CVM.

Application Note:

The CRS User verification status is part of the input parameters of the operations.

FDP_ETC.1/CRS Export of user data without security attributes

FDP_ETC.1.1/CRS The TSF shall enforce the CRS Data information flow control policy when exporting user data, controlled under the SFP, outside of the TOE.

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

Refinement:

The SFR applies to the Transaction Data exported by S.CRS at completion of the operation OP.CRS_AUTHORIZE_TRANSACTION.

FDP_IFC.1/CRS Subset information flow control

FDP_IFC.1.1/CRS The TSF shall enforce the CRS Data information flow control policy on Operations OP.CRS_PERSO when used to update the D.AUTH_RANDOM, OP.CRS_AUTHORIZE_CL and OP.CRS_AUTHORIZE_TRANSACTION.

FDP_IFF.1/CRS Simple security attributes

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

- **Subjects:**
 - S.CRS receiving the CRS Personalization APDU commands.
 - S.CRS receiving the S.CRS_USER authentication status.
 - S.CRS exporting the transaction data.
- **Information:**
 - Authorization Random,
 - User Authentication Status,
 - Transaction Data_

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

- The subject S.CRS shall accept a new Authorization Random only if it has successfully verified its integrity and confidentiality evidences.
- The subject S.CRS shall accept an Authentication Status for the S.CRS_USER only if it has successfully verified its integrity evidence.

FDP_IFF.1.3/CRS The TSF shall enforce the following additional CRS DATA information flow control SFP rules: transaction data resulting from OP.CRS_AUTHORIZE_TRANSACTION shall be exported with a valid signature to preserve their integrity.

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

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

Application Note:

The operations covered by this SFR are: OP.CRS_PERSO, OP.CRS_AUTHORIZE_CL and OP.CRS_AUTHORIZE_TRANSACTION.

The integrity evidence is verified with D.AUTH_KEYS MAC Key.

The confidentiality evidence is verified with D.AUTH_KEYS ENC Key.

Transaction data integrity is enforced with the use of the Signing Key of S.SD in the role of the CASD.

FDP_UCT.1/CRS Basic data exchange confidentiality

FDP_UCT.1.1/CRS 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 unauthorised disclosure.

Application Note:

The assets protected by this SFP for CRS include D.AUTH_KEYS and D.CL_REGISTRY.

FDP_UIT.1/CRS Data exchange integrity

FDP_UIT.1.1/CRS 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/CRS The TSF shall be able to determine on receipt of user data, whether **modification, deletion, insertion, replay** has occurred.

Application Note:

The assets protected by this SFP for CRS include D.AUTH_KEYS and D.CL_REGISTRY.

6.1.3.3.2 Class FMT: Security management

FMT_MSA.1/CRS Management of security attribute

FMT_MSA.1.1/CRS The TSF shall enforce the **Secure Channel Protocol information flow control policy** to restrict the ability to **change default value of the security attributes CRS_Default Mark, CRS_Applet CVM, CRS_Applet CVM Override and CRS_Applet CVM Required to S.CAD and S.SD**.

Application Note:

The default value of these security attributes is set during installation of the related applet.

FMT_MSA.4/CRS Security attribute value inheritance

FMT_MSA.4.1/CRS The TSF shall use the following rules to set the value of security attributes:

- The subject S.INSTALLER shall set the default value during applet's installation for the following security attributes:
 - CRS_Default Mark to either "MarkForFreeze" or "MarkForDelete",
 - CRS_Applet CVM for each applet individually,
 - CRS_Applet CVM Override for each applet individually,
 - CRS_Applet CVM Required for each applet individually.
- The default value for CRS_CL Enabled of all applets is always "disabled".
- CRS_CL Enabled is automatically set to "disabled" after having been "enabled" for an expiration of a fixed timer.

- The default value of CRS_Applet Mark is always “none” for each individual applet.
- Upon successful update of the D.AUTH_RANDOM using the operation OP.CRS_PERSONO, the security attribute CRS_Applet Mark of all installed applets is set to CRS_Default Mark.

6.2 Security Assurance Requirements

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

6.3 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.

6.3.1 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 occurrences 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.

The rationale for the following objectives for the SCP is included here, since they were added to the TOE or 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).

O.MIGRATION This objective is directly covered by FDP ACC.2 /MIGRATION, FDP ACF.1/MIGRATION, FDP UIT.1/MIGRATION, FMT MSA.1/MIGRATION, FPT_FLS.1/MIGRATION. These SFRs cover the applet migration restrictions and flow.

O.SCP.UPDATE This objective is directly covered by the SFR implementing the OS Update security policies FDP ACC.2 /UPDATE and FDP ACF.1/UPDATE. In addition, the integrity and confidentiality during data exchange is enforced with the trusted path established with FDP UIT.1/UPDATE. The flow consistency is covered by FPT_FLS.1/UPDATE.

O.CRS_REGISTRY This objective is directly covered by the SFRs implementing secure management of the CRS security attributes: secure import is ensured by integrity and confidentiality enforcement (FDP UCT.1/CRS, FDP UIT.1/CRS, FDP IFC.1/CRS, FDP IFF.1/CRS, FDP IFC.2/SD, FDP IFF.1/SC), secure storage for reliable data is ensured by FDP SDI.2 and secure management is ensured by FMT MSA.1/CRS and FMT MSA.4/CRS.

O.CRS AUTH This objective is directly covered by the SFRs implementing the Access Control SFPs and Information flow FCPs:

- CRS Data Information FCP (FDP IFC.1/CRS, FDP IFF.1/CRS),
- CRS Personalization and associated SCP Information FCP (FDP UIT.1/CRS, FDP ACC.2/CRS_PERSO) and SD access control SFP (FDP ACF.1/SD),
- CRS Authentication access control SFP (FDP ACC.2/CRS_AUTH, FDP ACF.1/CRS_AUTH)
- CRS Override access control SFP (FDP ACC.2/CRS_CVM, FDP ACF.1/CRS_CVM)

O.CRS TRANSACTION This objective is directly covered by the secure handling of the transaction related SFRs: secrets are manipulated with enforcement of privacy (FPR UNO.1) and integrity (FDP SDI.2), User authentication status import and transaction data export are secure (FDP IFC.1/CRS, FDP ETC.1/CRS, FDP IFF.1/CRS).

6.3.2 Rationale Tables of Security Objectives and SFRs

The reader is referred to the rationale tables Table 7 and Table 8 from [PP(U)SIM], Section 5.3.2, and [PPJCSv3.0], Section 7.3.2. This ST extends these rationale tables by the following tables Table 16 and Table 17. FCS COP.1 includes all iterations of FCS COP.1.1 in Section 6.1.1.1.2 and FCS COP.1.1/DAP in Section 6.1.2.1.1.

Security Objectives	Security Functional Requirements	Rationale
<u>O.SCP.RECOVERY</u>	FPT_FLS.1, FAU_ARP.1	Section 6.3.1
<u>O.SCP.SUPPORT</u>	FDP_SDI.2, FDP_ROL.1/FIREWALL, FCS_COP.1.1, FCS_COP.1.1/DAP	Section 6.3.1
<u>O.SCP.IC</u>	FPT_PHP.3	Section 6.3.1

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 6.3.1
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 6.3.1
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 6.3.1
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 6.3.1
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 6.3.1
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 6.3.1
O.INSTALL	FDP_ITC.2/CCM, FPT_RCV.3/Installer, FPT_FLS.1/CCM	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.GLOBAL-ARRAYS-CONFID	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FDP_RIP.1/bArray, FDP_RIP.1/APDU, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.GLOBAL-ARRAYS-INTEG	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.NATIVE	FDP_ACF.1/FIREWALL	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.REALLOCATION	FDP_RIP.1/ABORT, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ADEL, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.CIPHER	FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4, FCS_COP.1, FPR_UNO.1	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1

O.KEY-MNGT	<u>FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4, FCS_COP.1, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_SDI.2, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT</u>	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.PIN-MNGT	<u>FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FPR_UNO.1, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_ROL.1/FIREWALL, FDP_SDI.2, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL</u>	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.TRANSACTION	<u>FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT, FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_RIP.1/OBJECTS</u>	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.OBJ-DELETION	<u>FDP_RIP.1/ODEL, FPT_FLS.1/ODEL</u>	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.DELETION	<u>FDP_ACC.2/ADEL, FDP_ACF.1/ADEL, FDP_RIP.1/ADEL, FPT_FLS.1/ADEL, FPT_RCV.3/Installer, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL</u>	[PPJCSv3.0], Section 7.3.3.1; Section 6.3.1
O.DOMAIN-RIGHTS	<u>FDP_ACC.1/SD, FDP_ACF.1/SD, FMT_MSA.1/SD, FMT_MSA.3/SD, FMT_SMF.1/SD, FMT_SMR.1/SD, FMT_MSA.1/SC, FMT_MSA.3/SC, FMT_SMF.1/SC, FIA_UID.1/SC, FIA_UAU.1/SC, FIA_UAU.4/SC, FTP_ITC.1/SC, FCO_NRO.2/SC, FDP_IFC.2/SC, FDP_IFF.1/SC</u>	[PP(U)SIM], Section 5.3.2; Section 5.3.1
O.APPLI-AUTH	<u>FDP_ROL.1/CCM, FPT_FLS.1/CCM, FCS_COP.1/DAP</u>	[PP(U)SIM], Section 5.3.2; Section 5.3.1
<u>O.MIGRATION</u>	<u>FDP_ACC.2 /MIGRATION, FDP_ACF.1/MIGRATION, FDP_UIT.1/MIGRATION, FMT_MSA.1/MIGRATION, FPT_FLS.1/MIGRATION</u>	Section 6.3.1
<u>O.SCP.UPDATE</u>	<u>FDP_ACC.2 /UPDATE, FDP_ACF.1/UPDATE, FDP_UIT.1/UPDATE, FPT_FLS.1/UPDATE</u>	Section 6.3.1
<u>O.CRS_REGISTRY</u>	<u>FDP_UCT.1/CRS, FDP_UIT.1/CRS, FDP_IFC.1/CRS, FDP_IFF.1/CRS, FDP_IFC.2/SD, FDP_IFF.1/SC, FDP_SDI.2, FMT_MSA.1/CRS, FMT_MSA.4/CRS.</u>	Section 6.3.1
<u>O.CRS_AUTH</u>	<u>FDP_IFC.1/CRS, FDP_IFF.1/CRS, FDP_UIT.1/CRS, FDP_ACC.2/CRS_PERSO, FDP_ACF.1/SD, FDP_ACC.2/CRS_AUTH, FDP_ACF.1/CRS_AUTH, FDP_ACC.2/CRS_CVM, FDP_ACF.1/CRS_CVM</u>	Section 6.3.1
<u>O.CRS_TRANSACTION</u>	<u>FPR_UNO.1, FDP_SDI.2, FDP_IFC.1/CRS, FDP_ETC.1/CRS, FDP_IFF.1/CRS.</u>	Section 6.3.1

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]	<u>O.SCP.SUPPORT</u>
FDP_ROL.1/FIREWALL		<u>O.CRS_REGISTRY</u>
FCS_COP.1		<u>O.CRS_TRANSACTION</u>

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.CRS_REGISTRY
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]	<u>O.SCP.RECOVERY</u>
FAU_ARP.1		
FPT_PHP.3	-	<u>O.SCP.IC</u>
FDP_ACC.2/MIGRATION	-	<u>O.MIGRATION</u>
FDP_ACF.1/MIGRATION		
FDP_UIT.1/MIGRATION		
FMT_MSA.1/MIGRATION		
FPT_FLS.1/MIGRATION		
FDP_ACC.2/UPDATE		<u>O.SCP.UPDATE</u>
FDP_ACF.1/UPDATE		
FDP_UIT.1/UPDATE		
FPT_FLS.1/UPDATE		
FDP_ACF.1/SD		
FPR_UNO.1	Table 8 in [PPJCSv3.0]	<u>O.CRS_TRANSACTION</u>
FDP_ACC.2/CRS_AUTH	-	<u>O.CRS_AUTH</u>
FDP_ACC.2/CRS_PERSO	-	
FDP_ACC.2/CRS_CVM	-	
FDP_ACF.1/CRS_AUTH	-	
FDP_ACF.1/CRS_CVM	-	
FDP_ETC.1/CRS	-	
FDP_IFC.1/CRS	-	<u>O.CRS_REGISTRY</u>
FDP_IFF.1/CRS	-	<u>O.CRS_AUTH</u> <u>O.CRS_TRANSACTION</u>
FDP_UIT.1/CRS	-	<u>O.CRS_REGISTRY</u> <u>O.CRS_AUTH</u>
FDP_UCT.1/CRS	-	<u>O.CRS_REGISTRY</u>
FMT_MSA.1/CRS	-	
FMT_MSA.4/CRS		
FDP_IFC.2/SD		
FDP_SDI.2		

Table 17 SFRs and Security Objectives

6.3.3 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.

6.3.3.1 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
<u>FCS_PHP.3</u>	<u>No Dependencies</u>	-

Table 18 SFRs Dependencies

6.3.4 Security Assurance Requirements Rationale

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

6.3.5 AVA_VAN.5 Advanced methodical vulnerability analysis

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

6.3.6 ALC_DVS.2 Sufficiency of security measures

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

7 TOE summary specification

7.1 TOE Security Functions

7.1.1 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.

7.1.2 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.

7.1.3 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, Triple-DES, AES and ECC. 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 or the use of elliptic curves over finite fields for elliptic curve cryptography (ECC).
2. The random number generator provided by the BCM_SPS02 C0 is a PTG.2 [BCM-ST]. 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 ECC keys by the following means:
 - Applications may use the Java Card API method Key.clearKey() for key destruction.
 - An authenticated off-eSE 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-eSE 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, RSAPublicKey, ECPrivateKey and ECPublicKey.
5. Key access is provided via the Java Card API get methods of classes AESKey, DESKey, RSAPrivateCrtKey, RSAPrivateKey, RSAPublicKey, ECPrivateKey and ECPublicKey.
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 ECC is provided via the Java Card API methods. AES is implemented according to [FIPS 197] and [SP800-108], Triple-DES according to [SP800-67], the CBC and ECB modes of operation according to [SP800-38A], RSA cipher according to [PKCS1] and ECC according to [ANSI X9.63-2011]. 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 ECC is provided to applications via the Java Card API methods defined in the `javacard.security.Signature` class. The implementation of the RSA algorithm is according to [PKCS1] for RSASSA-PKCS1-v1_5 and RSASSA-PSS and of the ECC algorithm according to [ANSI X9.62-2005].
 - 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.

7.1.4 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.

7.1.5 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.

7.1.6 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-eSE entity (Application Provider or Controlling Authority) has been authorized to perform 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-eSE 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 eSE is the ISSUER 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.

7.1.7 SF_SECURE_CHANNEL

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

1. An off-eSE entity may initiate secure communication with the TOE by the following means: SCP02 and SCP03..
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-eSE entity by the on-eSE Application. For SCP02 and SCP03 the commands INITIALIZE UPDATE and EXTERNAL AUTHENTICATE are provided by the TOE for this purpose.
The TOE checks if the off-eSE 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 and SCP03 mutual authentication is by means of cryptographic exchange between the eSE and the off-eSE entity initiated by the off-eSE 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.

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.

7.1.8 SF_APPLET_MIGRATION

This security function provides a secure mechanism to migrate an embedded Applet. Applet migration is a feature that allows the user in possession of the OS update keys (hash table signature key) to update a pre-issuance applet to a newer or older version while keeping the (personalization) data of the applet.

1. The applet, in order to be migrated, must be Migratable (its MigratableStatus indicates this capability). Which means it is possible to export its data to secure data store of their associate Security Domain (Export Status) and later import them to the migrated applet after the applet code migration took place.

The applet migration process requires following sequences of operation:

- Applet data export
- On-board deletion
- Loading of new applet package(s).
- On-board installation
- Applet data import.

In case of interruption or failure in the process, the mechanism shall recover a safe state to resume the migration.

2. The new applet package should have an AID different from the current package AID.

In addition, the Class AIDs should be kept unchanged.

3. The integrity and confidentiality of the new applet packages imported in the TOE shall be ensured before the applet code and data are replaced.

7.1.9 SF_OS_UPDATE

This security function provides a secure mechanism to update the OS and all pre-issuance applications.

OS update is performed by the IC Embedded Software developer using interfaces if of the IC Loader (F.Flash Loader security function in [BCM-ST] section 7.8).

1. The OS Update mechanism enforces a verification over the OS version, comparing the version of the existing OS and the version of the OS being loaded. It is possible to upgrade the OS to a higher version (greater Minor version with same Major version, or greater Major version with any Minor version), but also to downgrade under special conditions: it is only possible to load an OS with the same Major version (Minor version could be smaller).

The OS update mechanism also enforces a verification of the ECDSA signature by the Security IC Embedded Software developer.

2. The incoming user data are signed to guarantee the origin and the integrity of incoming user data (Code and Data of the OS and applets included in the update).

They are also encrypted by Security IC Embedded Software developer to maintain the confidentiality of all incoming user data.

3. During the processing of an OS update, the operation of the eSE is limited. OS Update is split in a set of consecutive phases and it is possible to resume a phase that did not complete. The OS update phases are the following:

Migration OS Load: importation of the new image

- Clean-up and Transformation path selected: OS Data migration preparation
- OS Copy
- OS Update Completion: data restoration

7.1.10 SF_CRIS

This security function provides a secure mechanism for Contactless Registry Services (acc. to [GP CS]).

This security function is in charge of the the CRS Authentication access control, the CRS CVM Override access control, the CRS Data information flow control and the Secure Channel Protocol information flow control policy and the Security Domain access control policy.

1. The CRS Authentication access control policy enforces that no authorized transaction is permitted when an applet is marked as frozen or for deletion. This policy can be disabled for any applet.
2. The CRS CVM Override access control policy enforces that the CRS Authentication access control policy can only be disabled for a marked applet by the CVM
3. The CRS Data information flow control policy controls the exporting of user data outside the TOE especially for the personalization phase. Authorisation Random and Authentication Status is only accepted if it has successfully verified its integrity by D.AUTH_KEYS MAC Key and confidentiality evidence with D.AUTH_KEYS ENC Key. Transaction data is secured with a valid signature with the use of the Signing Key of the security Domain in the role of the CASD to preserve their integrity.
4. The Secure Channel Protocol information flow control policy and the Security Domain access control policy enforces the secure receiving of user data protected from unauthorised disclosure. On receipt of user data modification, deletion, insertion and replay of this data can be detected.

7.2 Fulfilment of the SFRs

SFR	TOE Summary Specification
[PPJCSv3.0] 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
[PP(U)SIM] 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
	SF_CRYPT.7
FCS_COP.1/DAP	
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
FDP_ACC.2 /MIGRATION	SF_APPLET_MIGRATION.1
FDP_ACF.1/MIGRATION	SF_APPLET_MIGRATION.1
FDP_UIT.1/MIGRATION	SF_APPLET_MIGRATION.3
FMT_MSA.1/MIGRATION	SF_APPLET_MIGRATION.2
FPT_FLS.1/MIGRATION	SF_APPLET_MIGRATION.1
FDP_ACC.2 /UPDATE	SF_OS_UPDATE_1
FDP_ACF.1/UPDATE	SF_OS_UPDATE_1
FDP_UIT.1/UPDATE	SF_OS_UPDATE_2
FPT_FLS.1/UPDATE	SF_OS_UPDATE_3
FDP_ACC.2/CRS_AUTH	SF_CRS_1
FDP_ACC.2/CRS_PERSO	SF_CRS_4
FDP_ACC.2/CRS_CVM	SF_CRS_2
FDP_ACF.1/CRS_AUTH	SF_CRS_1
FDP_ACF.1/CRS_CVM	SF_CRS_2
FDP_ETC.1/CRS	SF_CRS_3
FDP_IFC.1/CRS	SF_CRS_3
FDP_IFF.1/CRS	SF_CRS_3
FDP_UCT.1/CRS	SF_CRS_4
FDP_UIT.1/CRS	SF_CRS_4
FMT_MSA.1/CRS	SF_CRS_4
FMT_MSA.4/CRS	SF_CRS_4

Table 19 Mapping of SFRs to mechanisms of TOE

8 Statement of Compatibility

This is a statement of compatibility between this Composite Security Target (Composite-ST) and the Platform Security Target (Platform-ST, [BCM-ST]).

8.1 Matching statement

The TOE relies on the certified BCM_SPS02 C0 Microcontroller and the following specific security functionality it provides:

- Secure Processing System (SPS) hardware accelerators for AES, HMAC, CRC and TDES cryptographic operations.
- True Random Number Generator (TRNG) for seeding the ISO/IEC 18031 RNG.
- Cryptographic library that implements the composite-TOEs RSA signature generation and verification, RSA key generation, ECDH Key Exchange, ECDSA signature generation and verification, EC Key pair generation, AES and DES in various operation modes, Secure Hash Computation and CRC computation (CRC16C and CRC32C).

The rationale of the Platform-ST has been used to identify the relevant SFRs, TOE objectives, threats and OSPs.

8.2 Security Environment

Lable in Platform-ST	Lable in Composite-ST
T.Phys_Manipulation T.Phys-Probing T.Malfunction T.Leak-Inherent T.Leak-Forced T.Abuse-Func T.RNG T.Mem-Access	T.PHYSICAL
P.Process-TOE	OE.PRODUCTION
P.Crypto-Service	OSP.VERIFICATION, OSP.APSD-KEYS, OSP.KEY-GENERATION, OSP.CASD-KEYS, OSP.VASD-KEYS, OSP.KEY-CHANGE, OSP.SECURITY-DOMAINS
A.Process-Sec-IC	OE.PRODUCTION
A.Resp-Appl	O.FIREWALL, O.SID, O.OPERATE, OE.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.KEY-MNGT, O.PIN- MNGT, O.TRANSACTION, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.REALLOCATION, O.DOMAIN-RIGHTS
O.Phys-Manipulation	O.SCP.IC, O.SCP.SUPPORT
O.Phys-Probing	O.SCP.IC, O.SCP.SUPPORT
O.Malfunction	O.SCP.IC, O.SCP.RECOVERY, O.SCP.SUPPORT

O.Leak-Inherent	O.SCP.IC, O.SCP.SUPPORT
O.Leak-Forced	O.SCP.IC, O.SCP.SUPPORT
O.Abuse-Func	O.SCP.IC, O.SCP.SUPPORT
O.Identification	O.SID
O.RND	O.SCP.SUPPORT
O.TDES	O.SCP.IC, O.SCP.SUPPORT
O.AES	O.SCP.IC, O.SCP.SUPPORT
O.SHA	O.SCP.IC, O.SCP.SUPPORT
O.Mem-Access	O.SCP.SUPPORT
O.RSA	O.SCP.IC, O.SCP.SUPPORT
O.ECC	O.SCP.IC, O.SCP.SUPPORT
O.HMAC	O.SCP.IC, O.SCP.SUPPORT
OE.Resp-Appl	O.FIREWALL, O.SID, O.OPERATE, OE.CARD-MANAGEMENT, O.SCP.RECOVERY, O.SCP.SUPPORT, O.CIPHER, O.KEY-MNGT, O.PIN-MNGT, O.TRANSACTION, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.REALLOCATION, O.DOMAIN-RIGHTS
OE.Process-Sec-IC	OE.PRODUCTION

Table 20 Mapping of Threats, OSP's, Assumptions and Security Objectives

OE.Process-Sec-IC is related to the capability of loading the Security IC Embedded Software into the NVM under the composite product manufacturer authority. This capability is not available in User configuration, when the TOE is delivered to Phase 6, for Personalization. The Composite ST requires the production and personalization environment to be trusted and secure (OE.PRODUCTION). The associated TOE guidance documents provide information on the delivery procedures and the protections that support a controlled loading.

A.Resp-Appl assumes that all user data owned by the Security IC Embedded Software is treated as defined for its specific application context. The [PPJCSv3.0] and the [PP(U)SIM] demand that user data must be protected from unauthorized disclosure and modification (see [PPJCSv3.0], Chapter 5 and 6 and [PP(U)SIM] Chapter 3 and 4) so all those objectives which cover the threats T.CONFID-APPLI-DATA, T.INTEG-APPLI-DATA and T.INTEG-USER-DATA fulfill the assumption.

O.SCP.RECOVERY matches to O.Malfunction because this allows the TOE to eventually complete the interrupted operation successfully.

O.SCP.SUPPORT matches all listed objectives of the Platform-ST because they provide functionality that supports the well-functioning of the TSFs of the TOE (avoiding they are bypassed or altered). O.RND particularly provides a required low-level-security cryptographic function to the Java Card System.

O.SCP.IC matches the above listed objectives of the Platform-ST except O.RND because they describe features against physical attacks.

8.3 Security requirements

Relevant Platform-SFR	Correspondence in Composite ST
FRU_FLT.2, FPT_FLS.1	FPT_FLS.1.1, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL, FPT_FLS.1/CCM, FPT_RCV.3/Installer The TOE relies on the mechanisms and information about the operating conditions monitored during its execution provided by the chip platform.
FPT_PHP.3	FPT_PHP.3, FPR_UNO.1, FPT_RCV.3/Installer The TOE relies on the countermeasures and protection mechanisms of the chip platform for resistance to physical attacks.
FDP_SDI.2	FDP_SDI.2
FPT_TST.2	Internal self-test features of the BCM platform are not accessible by the Composite TOE.
FDP_ITT.1	FDP_IFC.1/JCVM
FDP_IFC.1	FDP_IFC.1/JCVM
FDP_SDC.1	FDP_SDI.2.1
FMT_LIM.1	Internal test features of the BCM platform are not accessible by the Composite TOE.
FMT_LIM.2	Internal test features of the BCM platform are not accessible by the Composite TOE.
FAU_SAS.1	Test process before TOE Delivery is not used by the composite TOE.
FCS_COP.1/AES	FCS_COP.1/MAC_AES, FCS_COP.1/CIPH_AES The Platform cryptographic library is used by the TOE in all AES computations.
FCS_COP.1/DES	FCS_COP.1/MAC_TDES, FCS_COP.1/CIPH_TDES The Platform cryptographic library is used by the TOE in all DES computations
FCS_COP.1/SHA	FCS_COP.1/SHA The Platform cryptographic library is used by the TOE in all SHA computations.
FCS_COP.1/RSA	FCS_COP.1/RSA, FCS_COP.1/SIG_RSA, FCS_COP.1/CIPH_RSA The Platform cryptographic library is used by the TOE in all RSA computations.
FCS_COP.1/ECC	FCS_COP.1/ECC The Platform cryptographic library is used by the TOE in all ECC computations.
FCS_CKM.1/RSA	FCS_CKM.1.1/RSA The Platform cryptographic library is used by the TOE in all RSA key generation.

FCS_CKM.1/ECC	FCS_CKM.1.1/ECC The Platform cryptographic library is used by the TOE in all ECC key generation.
FCS_CKM.4/TDES	FCS_CKM.4.1
FCS_CKM.4/AES	FCS_CKM.4.1
FDP_ACC.1	FDP_ACC.1/SD
FDP_ACF.1	FDP_ACF.1/SD
FMT_MSA.1	FMT_MSA.1/SD
FMT_MSA.3	FMT_MSA.3/SD
FMT_SMF.1	FMT_SMF.1/SD
FDP_ACF.1	FDP_ACF.1/UPDATE This SFR is implemented by the IC Platform.

Table 21 Mapping of SFRs

8.4 Assurance requirements

The Composite-ST requires EAL 4 according to Common Criteria V3.1R4 augmented by ALC_DVS.2 and AVA_VAN.5.

The Platform-ST requires EAL 5 according to Common Criteria V3.1 R4 augmented by: ALC_DVS.2 and AVA_VAN.5.

Since EAL 5 covers all assurance requirements of EAL 4 all non augmented parts of the Composite-ST will match to the Platform-ST assurance requirements.

8.5 Statement on contradictions

Overall there is no conflict between security requirements of this Composite-ST and the Platform-ST.

9 Assurance Measures

This chapter describes the Assurance Measures fulfilling the requirements listed in chapter 6.2.

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 22 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.
IC Dedicated Software	IC proprietary software embedded in the Security IC and developed by Broadcom.
Platform-ST	The hardware platform Security Target: [BCM-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 (acc. to [GP CCM])
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
ECC	Elliptic Curve Cryptography
eSE	Embedded Secure Element
GP	GlobalPlatform
HW/SW/FW	Hardware/Software/Firmware
IC	Integrated Circuit
JCRE	Java Card Runtime Environment
JCS	Java Card System
JCVM	Java Card Virtual Machine
IMSI	International Mobile Subscriber Identity
ISD	Issuer Security Domain
ITSEF	Information Technology Security Evaluation Facility
n.a.	not applicable
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
RAM	Remote Application Management
RMI	Remote Method Invocation
SAR	Security Assurance Requirement
SCP	Smart Card Platform Secure Channel Protocol
SCWS	Smart Card Web Server
SD	Security Domain
SF	Security Function

Acronym	Term
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
USIM	Universal Subscriber Identity Module
(U)SIM PP	(U)SIM Java Card™ Platform Protection Profile
VA	Verification Authority

11 Bibliography

- [AGD_PRE] Sm@rtSIM CX Virgo V5.0 Preparative Procedures, Giesecke+Devrient Mobile Security GmbH
- [AGD_OPE] Sm@rtSIM CX Virgo V5.0 Operational Guidance Common Document, Giesecke+Devrient Mobile Security GmbH
 Sm@rtSIM CX Virgo V5.0 Operational Guidance for the Application Developer, Giesecke+Devrient Mobile Security GmbH
 Sm@rtSIM CX Virgo V5.0 Operational Guidance for the Application Provider, Giesecke+Devrient Mobile Security GmbH
 Sm@rtSIM CX Virgo V5.0 Operational Guidance for the Controlling Authority, Giesecke+Devrient Mobile Security GmbH
 Sm@rtSIM CX Virgo V5.0 Operational Guidance for the Issuer, Giesecke+Devrient Mobile Security GmbH
 Sm@rtSIM CX Virgo V5.0 Operational Guidance for Personaliser, Giesecke+Devrient Mobile Security GmbH
 Sm@rtSIM CX Virgo V5.0 Operational Guidance for the Verification Authority, Giesecke+Devrient Mobile Security GmbH
- [ANSI X9.62-2005] American National Standard Institute, ANSI X9.6-2005, Public Key Cryptography for the Financial Services Industry - The Elliptic Curve Digital Signature Algorithm (ECDSA), 2005
- [ANSI X9.63-2011] American National Standard Institute, ANSI X9.63-2011, Public Key Cryptography for the Financial Services Industry - Key Agreement and Key Transport Using Elliptic Curve Cryptography, 2011
- [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 1.0.1, January 2011
- [GP CS] GlobalPlatform, Contactless Services – Card Specification v2.2 – Amendment C, Version 1.1.1, July 2014
- [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
- [JCRM] 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
- [PP0084] Common Criteria Protection Profile, Security IC Platform Protection Profile with Augmentation Packages, BSI-PP-0084-2014, Version 1.0, 13.01. 2014
- [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 Authentication, RFC 2104, February 1997
- [RFC3447] Network Working Group, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1, RFC 3447, February 2003
- [BCM-ST] Security Target for the BCM_SPS02, Version 1.2, 2016-11-28
- [BCM-Cert] BSI Certification Report, BSI-DSZ-CC-0915- 2016 for BCM_SPS02 Secure Processing System with IC Dedicated Software, Version 1.0 from Broadcom Corporation, 25.02.2016; Assurance Continuity Maintenance Report BSI-DSZ-CC-0915-2016-MA-0, BCM_SPS02 Secure Processing System with Firmware version 002.010 or 002.020 from Broadcom Corporation, 01.06.2016; BSI-DSZ-CC-0915-2016-MA-01 (01.06.2016); BSI-DSZ-CC-0915-2016-MA-02 (27.07.2016); ; BSI-DSZ-CC-0915-2016-MA-03 (02.12.2016)
- [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
- [SP800-108] National Institute of Standards and Technology, Key Derivation using Pseudorandom Functions (SP 800-108) Validation System (KBKDFVS), April 23, 2013
- [TS102 226] ETSI TS 102 226 V10.0.0 (2012-03), Smart Cards; Remote APDU structure for UICC based applications (Release 10)