KONA26CC
Security Target Lite

KEBTechnology
Korea Electronic Banking Technology co.,Ltd.

Table of contents

1. ST Introduction

1.1. ST reference

Document No: Document Title:	SP-01-14 KONA26CC Security Target Lite
Version:	1
Revision:	0
Release date:	August, 3, 2010

Table 1. ST reference

1.2. TOE reference

Name:	KONA26CC
Version:	1
Revision:	1

Table 2. TOE reference

1.3. TOE overview

The TOE Type is a smartcard, which is composed of operative system KONA26CC version 1.1 and IC Samsung S3CC91C revision 0.

The TOE features Java Card 2.2.1, and GlobalPlatform 2.1.1.

It provides the security level of EAL4+ augmented with ALC_DVS.2 and AVA_VAN.5 and allows loading and deleting applications, which are developed by the customers. Thus, it allows for multiple applications to run on a single TOE and provides security features to ensure secure interoperability of applications.

The examples of the application that can be loaded in the TOE are:

- Government applications - National Identification Card, Driver License, Health Card
- Banking applications - Credit/Debit cards, ePurse
- Security Token applications - Public Key Infrastructure(PKI)
- Telecom applications - 3G USIM, JavaSIM

The TOE requires the following components in order to operate:

- Card Acceptance Device(CAD) or Smartcard Reader complies with [7816-3] communication
- Dedicated software that runs over the CAD to allow logical communication with the TOE

1.4. TOE description

1.4.1. TOE Architecture

The TOE is a composition of an IC hardware and a embedded software that controls the IC. These parts are:

- The Samsung S3CC91C IC, including the low level library that control cryptographic operations (Secure Crypto Library ver. 3.5S) and random number generation (DRNG ver. 2.0).

- The Kona OS operative system software ver. 1.2 , that runs on top of the IC and controls the operation of the whole card. The KONA OS allows customers to load and control their own applications while keeping them isolated from the rest of the card.

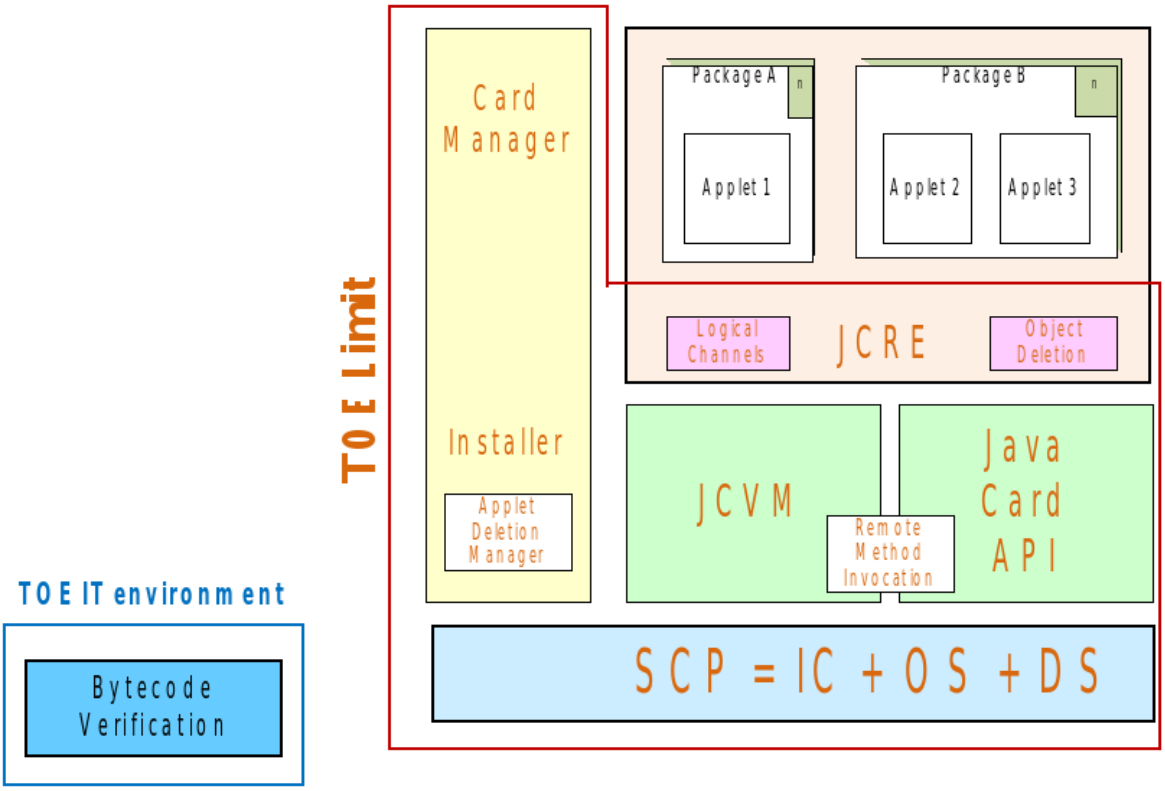

Illustration 1: Scope of TOE

It is important to remark that user applications and bytecode-verifier are not included under the TOE scope and that no additional native applications are installed in the TOE.

The TOE enables the Java Card technology, which consists of the Java Card Virtual Machine (JCVM), the Java Card Runtime Environment (JCRE) and the Java Card Application Programming Interface (JCAPI).

Also, the TOE complies Global Platform, providing the features of GlobalPlatform Environment (OPEN), the Issuer Security Domain (ISD) and Cardholder Verification Method Services.

Therefore, the TOE allows loading of multiple applications, called applets, that interact securely, under the rules defined by TOE provider.

1.4.2. TOE security functionalities

The TOE provides a wide area of physical protection measures and implements the logical security

measures, being them:

- Java Card 2.2.1 Functionalities:
  - Remote Method Invocation
    Supports the remote methods that can be invoked remotely from CAD.
  - Multiple Logical Channel
    Supports Multiple logical channels which allow a terminal to open up to two channels with the smart card, one session per logical channel. (Logical channels functionality is described in detail in ISO [7816-4].)
  - Garbage Collector
    Reclaims deallocated data automatically during the execution of a program.

- GlobalPlatform 2.1.1 Functionalities:
  - Issuer Security Domain
    Operates as the mandatory on-card representative of the Card Issuer which has capability of loading, installing, and deleting application that belong either to the Card Issuer or to other Application Provider.
  - Supplementary Security Domain
    Operates as the on-card representative of an Application Provider or Controlling Authority.
  - Public key DAP Verification
    Supports verification of application code integrity and authenticity before the application code is loaded, installed and made available to the Cardholder on behalf of an Application Provider.
  - Mandated DAP Verification
    Supports verification of application code integrity and authenticity before the application code is loaded, installed and made available to the Cardholder on behalf of a Controlling Authority.
  - Secure Channel Protocol 01 and Secure Channel Protocol 02
    Provides a secure communication channel between a card and an off-card entity during an Application Session.
  - CVM interface supporting Global PIN
    Provides support for CVM management which is responsible for Cardholder verification, including velocity checking.
  - Delegated Management
    Provides authority to Supplementary Security Domain to manage Card Contents.

- Cryptographic Algorithms and Functionalities:
  - DES with 64 bits key size
  - Triple DES with 128 bits or 192 bits key size
  - RSA with key length of multiple of 32 bits from 512 bits to 2048 bits
  - RSA CRT with key length of multiple of 32 bits from 512 bits to 2048 bits
  - Hash Algorithm - MD5 and SHA-1

- General Functionalities:
  - Protection against Physical Probing and against malfunctions
  - A non-deterministic random number generator (RNG)
  - Storage data integrity
  - Security alarms in case of detect a security violation
  - Atomicity of critical operations
  - Support for Communication Protocols T=0 and T=1
  - Support various baud rate for Communication Protocols (9600, 19200, 38400, 76800 and 115200) bit/sec

1.4.3. TOE environment

The TOE requires a CAD device and an application that implements communication with the card in order to operate in a correct way. These are not included under the TOE scope.

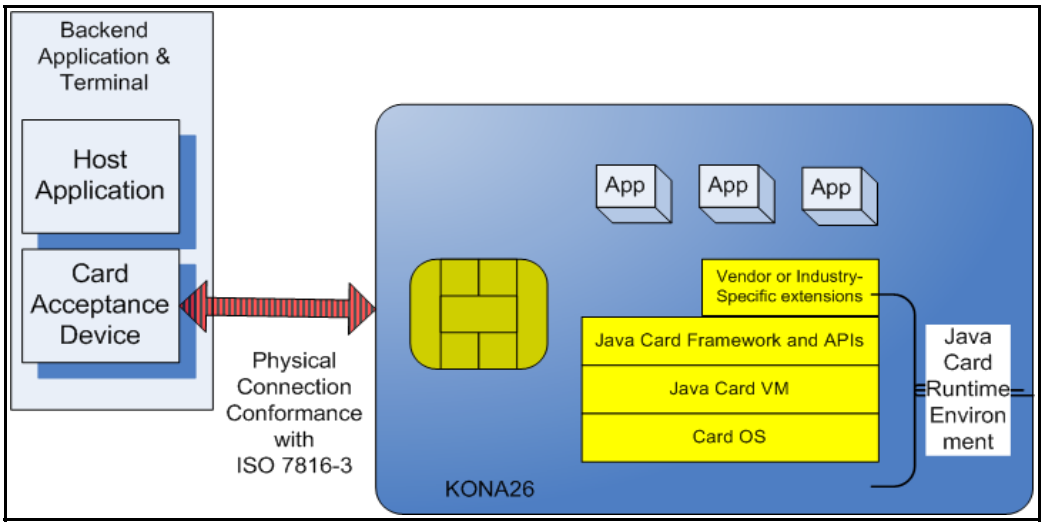

Illustration 2: TOE environment

The examples of TOE intended usage are:

- Government applications - National Identification Card, Driver License, Health Card
- Banking applications - Credit/Debit cards, ePurse
- Security Token applications - Public Key Infrastructure(PKI)
- Telecom applications - 3G USIM, JavaSIM

1.4.4. TOE life Cycle

The TOE lifecycle is composed of the platform fabrication and platform usage as shown the illustration 3, but the evaluation only covers the fabrication phase. It is a Application Provider's responsibility to follow the security recommendations and operation guidance of the TOE in order to keep the TOE secure during the platform usage phase.

In the platform fabrication phase, the following actions are performed:

- The IC developer designs and develops the IC.
- Kona OS is designed and developed by KEBT considering the recommendations and usage guidance provided by the IC manufacturer. Moreover, the Javacard 2.2.1 specification is followed for the JCRE, JCVM and JCAPI and the GlobalPlatform specifications are taken into account for the ISD, the OPEN and the GP API.
- During the IC fabrication phase, Kona OS is masked in Read Only Memory (ROM) of the IC.
- Initialization phase covers the initialization of data used by JavaCard Platform and allows the transfer of management capabilities. During Initialization phase, Issuer Security Domain's Key can be changed upon customer's request and the TOE is set to OP_READY state where the TOE is fully functional.

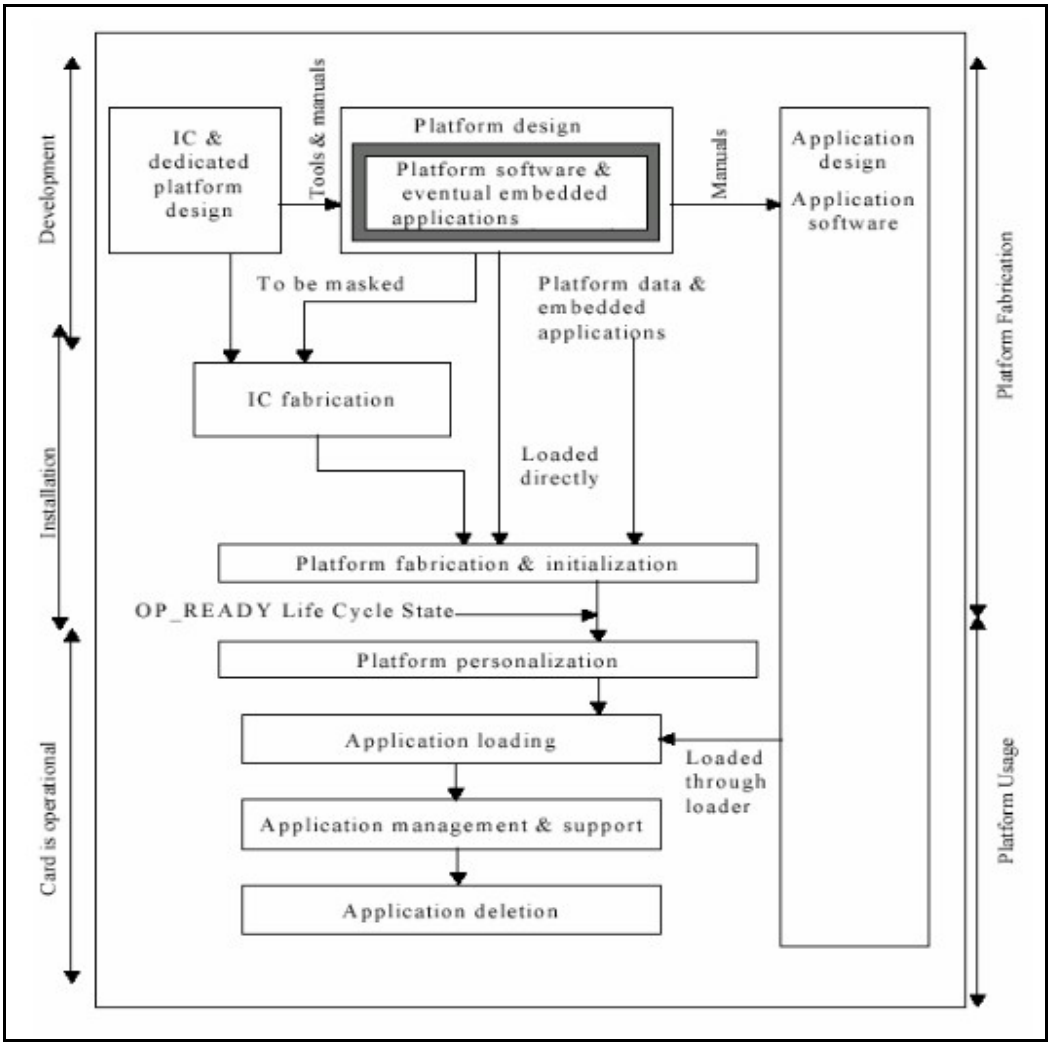

Illustration 3: Composite TOE life cycle

2. Conformance claims

2.1. Conformance to common criteria

This security target claims the following conformance with Common Criteria:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 3, July 2009 conformance.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3, July 2009 extended conformance. The only extended component used is FCS_RND.1 which is defined in [BSI-PP].
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 3, July 2009 conformant conformance.

2.2. Conformance to protection profiles

This security target claims the following conformance with protection profiles:

- A demonstrable conformance wit Java Card System - Standard 2.2 Configuration Protection Profile

version 1.0b, August 2003 conformance rationale [JC-PP]“.

2.2.1. Java Card System - Standard 2.2 Configuration Protection Profile version 1.0b, August 2003 conformance rationale

The TOE type of the current security target is "Java Card 2.2.1 conformant to Global Platform 2.1.1, implemented on Samsung IC, S3CC91C" and protection profile TOE type is "smart card platform enabled with Java Card technology". TOE types are compatible since the security target's TOE is a smart card that is enabled with Java Card technology.

2.2.1.1. Security problem definition compatibility rationale

All threats, organizational security policies and assumptions apply to the TOE. However the A.NATIVE assumption is not applicable to the environment since it is a non-bypassability assumption. Therefore it is removed due to the fact that in CC v3.1 non-bypassability is implicit.

Moreover, the security target defines:

- Additional threats to the security problem definition, which are compatible with threats of the protection profile but they are more focus to generic attacks on the smartcard platform and Global platform software. Being them T.ACCESS, T.OS_OPERATE, T.LEAKAGE and T.FAULT.

- Additional organizational security policies to the security problem definition, which are compatible with organizational security policies of the protection profile but they are applicable to configuration of Global Platform implementations. Being them P.ROLES-3, P.INITIAL_LIFECYCLE_STATE-2, P.CARD_ADMINISTRATOR_PRE-APPROVAL_2b, P.APPLICATION_PROVIDER_PRE-APPROVAL_2b, P.LOAD_FILE_VERIFICATION_3, P.APPLICATION_CODE_VERIFICATION_3, P.SECURE_COMMUNICATION_1.

- Additional assumptions to the security problem definition, which are compatible with assumptions of the protection profile but are related to how keep confidentiality of assets that affect to TOE and are stored off-card. Being them: A.KEY_MANAGEMENT.

Therefore, the security problem definition of the TOE is more restrictive than the security problem definition of the Protection profile.
Security objectives compatibility rationale

All security objectives of the Protection profile are included in the Security Target. However objective O.NATIVE does not appear in the Security Target because it is related to bypassability of security functionalities and in Common Criteria version 3.1 bypassability of security functionalities is implicit and therefore no objective and functional requirement it is required.

Furthermore the security target includes other security objectives that covers the additional components of the security problem definition. Therefore the security objectives of the protection profile are compatible with the security objectives for the TOE.

2.2.1.2. Security Requirements compatibility rationale

All existent CC 3.1 Revision 2 Part 2 SFRs that appear in the protection profile and apply to the TOE also appear in this security target operated in the same way because:

- The SFRs of the FCS class have been chosen and operated according to the TOE implemented cryptographic capabilities.
- The FPT_AMT.1 changes to FPT_TST.1 since it applies to a part of the TOE.
- The SFRs concerning to bytecode verification (which is done off-card in the TOE) are not included. Because in CC3.1r3 SFRs for the environment do not need to be stated.
- SFRs with minor aesthetic changes have been included as they appear in CC3.1r3.

However there are some security functional requirements that are not included because the TOE is conformance with CC 3.1 and PP is conformance with CC 2.1. Consequently, section 2.2.2 of the security target provides a statement of compatibility between PP conformant with CC 2.1 and TOE evaluation conformant with CC 3.1.

2.2.1.3. Security Assurance Requirements alignment rationale

The following table shows how to align the evaluation assurance level requested by the Protection profile (EAL4+ augmented with ADV_IMP.2 and AVA_VLA.3) and the evaluation assurance level of the TOE (EAL4+ augmented with ALC_DVS.2 and AVA_VAN.5):

SAR in CC 2.1	Alignment with SAR in CC 3.1
ACM_AUT.1	ALC_CMC.4 (scope of CM), ALC_CMS.4 (capabilities)
ACM_CAP.4	ALC_CMC.4 (scope of CM), ALC_CMS.4 (capabilities)
ACM_SCP.2	ALC_CMC.4 (scope of CM), ALC_CMS.4 (capabilities)
ADV_FSP.2	ADV_FSP.4 (extends with additional requirements)
ADV_HLD.2	ADV_TDS.3 (grouped with modules and subsystems)
ADV_LLD.1	ADV_TDS.3 (grouped with modules and subsystems)
ADV_RCR.1	Affects all ADV class, no direct correspondence
ADV_SPM.1	Not necessary in 3.1, covered by objectives definition
ADO_DEL.2	ALC_DEL.1 (for delivery process), AGD_PRE.1 (for user side)
ADO_IGS.1	AGD_PRE (preparation of TOE at the user's site), ALC_CMC.4 (generation procedures)
AGD_ADM.1	AGD_OPE.1 (includes administration)
AGD_USR.1	AGD_OPE.1 (includes user guidance)
ALC_DVS.1	ALC_DVS.2 (direct correspondence,)
ALC_LCD.1	ALC_LCD.1 (direct correspondence)
ALC_TAT.1	ALC_TAT.1 (direct correspondence)
ATE_COV.2	ATE_COV.2 (analysis demonstrates all TSFI are tested in accordance with the functional specification)
ATE_DPT.1	ATE_DPT.1 (direct correspondence)
ATE_FUN.1	ATE_FUN.1 (direct correspondence)
ATE_IND.2	ATE_IND.2 (direct correspondence)
AVA_MSU.2(PP) AVA_MSU.3(IC)	AGD_OPE.1 (misuses in operation), AGD_PRE.1 (misuses in preparation), AVA_VAN.5 (uses also guidance documentation looking for misuses, augmented)
AVA_VLA.3	AVA_VAN.5 (augmented specifying that attack potential must be high)
ADV_IMP.2	ADV_IMP.1 (direct correspondence)

Table 3. Alignment of SAR

2.2.2. Statement of compatibility between the evaluation CC version and PP CC version

The TOE evaluation is conformant with CC version 3.1 and the PP is conformant with CC version 2.1.

The current security target provides a statement of compatibility between them based on a study of differences between SFRs and SARs of both versions and a study of alignment between the security assurance requirements of both versions (Table 3).

The following table shows the changes between the SFRs in the protection profile and the SFRs in the ST considering CC version differences:

SFR / ITERATION	Description of Change
FDP_ACC.2 / FIREWALL	Changed from CC2.1 to CC3.1r3. In CC3.1r3 this component requires to cover all SFPs.
FDP_ACF.1 / FIREWALL	There is a minor aesthetic change from CC2.1 to CC3.1r3
FDP_IFC.1 / JCVM	None
FDP_IFF.1 / JCVM	Changed from CC2.1 to CC3.1r3. In CC3.1r3 an additional element from CC2.1 which was unused in the protection profile has been removed.
FDP_RIP.1 / OBJECTS	None
FMT_MSA.1 / JCRE	None
FMT_MSA.2 / JCRE	Changed from CC2.1 to CC3.1r3. In CC3.1r3 there is an assignment that details the list of security attributes that require only secure values to be provided.
FMT_MSA.3 / FIREWALL	None
FMT_SMR.1 / JCRE	None
FMT_SMF.1 / JCRE	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3
FPT_SEP.1	Not defined in CC3.1r3. In CC3.1r3 this requirement is implicit of the architecture and therefore it does not appear
FCS_CKM.1	Changed from CC2.1 to CC3.1r3. A removal of the dependency with FMT_MSA.2
FCS_CKM.2	Changed from CC2.1 to CC3.1r3. A removal of the dependency with FMT_MSA.2
FCS_CKM.3	Changed from CC2.1 to CC3.1r3. A removal of the dependency with FMT_MSA.2
FCS_CKM.4	Changed from CC2.1 to CC3.1r3. A removal of the dependency with FMT_MSA.2
FCS_COP.1	Changed from CC2.1 to CC3.1r3. A removal of the dependency with FMT_MSA.2
FDP_ROL.1 / FIREWALL	None
FAU_ARP.1 / JCS	None
FDP_SDI.2	None
FPT_RVM.1	Not defined in CC3.1r3. In CC3.1r3 this requirement is implicit of the architecture and therefore it does not appear
FPT_TDC.1	None
FPT_FLS.1 / JCS	None
FPR_UNO.1	None
FPT_AMT.1	Does not apply to CC3.1r3
FPT_TST.1	Changed from CC2.1 to CC3.1r3. In CC3.1r3 the elements FPT_TST.1.1 and FPT_TST.1.2 where implicitly applied to the TSF. In this ST, the elements FPT_TST.1.1 and FPT_TST Following threats apply to more generic attacks

	on the Smart Card Platform OS and GP software. Threat agent does not use application but observes OS behavior or uses GP specific commands..1.2 have been operated to apply to the TSF.
FMT_MTD.1 / JCRE	None
FMT_MTD.3	Changed from CC2.1 to CC3.1r3. In CC3.1r3 the element FMT_MTD.3 was implicitly applied to TSF data. In this ST, the element the element FMT_MTD.3 has been operated to apply to TSF data.
FIA_ATD.1 / AID	None
FIA_UID.2 / AID	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3
FIA_USB.1	Changed from CC2.1 to CC3.1r3. In CC3.1r3 the element FIA_USB.1.1 requires a list of user appropriate security attributes. Moreover, this component has two additional elements in CC3.1r3 that detail rules for initial association of attributes and for rules for changing attributes.
FPT_FLS.1 / SCP	None
FRU_FLT.1 / SCP	None
FPT_PHP.3 / SCP	None
FPT_SEP.1	Not defined in CC3.1r3. In CC3.1r3 this requirement is implicit of the architecture and therefore it does not appear
FPT_RVM.1	Not defined in CC3.1r3. In CC3.1r3 this requirement is implicit of the architecture and therefore it does not appear
FPT_RCV.3 / SCP	None
FPT_RCV.4 / SCP	None
FDP_ITC.2 / Installer	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3
FMT_SMR.1 / Installer	None
FPT_FLS.1 / Installer	None
FPT_RCV.3 / Installer	None
FRU_RSA.1 / Installer	None
FDP_ACC.2 / JCRMI	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3
FDP_ACF.1 / JCRMI	None
FDP_IFC.1 / JCRMI	None
FDP_IFF.1 / JCRMI	None
FMT_MSA.1 / JCRMI	None
FMT_MSA.3 / JCRMI	None
FMT_REV.1 / JCRMI	Changed from CC2.1 to CC3.1r3. In CC3.1r3 the element FMT_REV.1.1 details the type of resource whose attribute may be revoked.
FMT_SMR.1 / JCRMI	None
FMT_SMF.1 / JCRMI	None
FDP_RIP.1 / ODEL	None
FPT_FLS.1 / ODEL	None
FDP_ACC.2 / ADEL	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3

FDP_ACF.1 / ADEL	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3
FMT_MSA.1 / ADEL	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3
FMT_MSA.3 / ADEL	Changed from CC2.1 to CC3.1r3. There is a minor aesthetic change from CC2.1 to CC3.1r3
FMT_SMR.1 / ADEL	None
FMT_SMF.1 / ADEL	None
FDP_RIP.1 / ADEL	None
FPT_FLS.1 / ADEL	None
FCO_NRO.2 / CM	None
FIA_UID.1 / CM	None
FDP_IFC.2 / CM	None
FDP_IFF.1 / CM	None
FDP_UIT.1 / CM	None
FMT_MSA.1 / CM	None
FMT_MSA.3 / CM	None
FMT_SMR.1 / CM	None
FMT_SMF.1 / CM	None
FTP_ITC.1 / CM	There is a minor aesthetic change from CC2.1 to CC3.1r3
FDP_ACC.1 / CMGR	None
FDP_ACF.1 / CMGR	None
FMT_MSA.1 / CMGR	None
FMT_MSA.3 / CMGR	None
FMT_SMR.1 / CMGR	None
FMT_SMF.1 / CMGR	None
FIA_UID.1 / CMGR	None

Table 4. Comparative of changes between SFR in CC versions

The following enumeration shows the changes between versions made to security assurance requirements:

1. ALC, ACM and ADO class has been merged in one, called ALC that covers completely the requirements of their predecessors
2. ASE class has been modified to allow different levels of assurance for ASE_OBJ, ASE_REQ and ASE_TSS.
3. ADV has suffered the maximum change, that can be grouped in:
    1. Adding a new feature security architecture (ADV_ARC) at auto-protection, secure initialization, anti-tampering and non-bypassability level
    2. Restyling of ADV_HLD (subsystems) and ADV_LLD (modules) in a new family ADV_TDS, that it allows TOE design requirements
    3. Suppression of ADV_RCR representation correspondence
    4. The scope of ADV_IMP and ADV_INT have been kept
4. ACO class has been added and it applies when the TOE is composed of several TOEs already evaluated
5. AGD class has been modified and adapted to create a guidance for operational state and preparative steps (AGD_OPE and AGD_PRE)
6. ATE class keeps the scope of its predecessor in terms of families (ATE_COV, ATE_DPT, ATE_FUN and ATE_IND)

7. AVA class has been modified. It groups the previous families (AVA_CCA, AVA_MSU, AVA_SOF and AVA_VLA). The attack potential has been modified and now there are four levels “Basic”, “Enhanced-Basic”, “Moderate” and “High”.

The statements above-mentioned demonstrate that changes in the SFRs do not affect the functionality of the product. Also, changes in the SARs do not affect the assurance coverage since the changes add extra requirements.

2.3. Conformance to packages

This security target claims conformity with EAL4 +, augmented with:

- AVA_VAN.5 **Advanced methodical vulnerability analysis**
- ALC_DVS.2 Sufficiency of security measures

2.4. Security problem definition

2.4.1. Threats

This section describes the security threats to the TOE:

- T.PHYSICAL
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.CONFID-JCS-CODE
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.CONFID-APPLI-DATA
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.CONFID-JCS-DATA
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.INTEG-APPLI-CODE
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.INTEG-JCS-CODE
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.INTEG-APPLI-DATA
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.INTEG-JCS-DATA
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.SID.1
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.SID.2
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.EXE-CODE.1
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

- T.EXE-CODE.2
  - This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.NATIVE
◦This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.RESOURCES
◦This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.INTEG-APPLI-CODE.2
◦This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.INTEG-APPLI-DATA.2
◦This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.INSTALL
◦This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.EXE-CODE-REMOTE
◦This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.DELETION
◦This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.OBJ-DELETION
This threat is defined in Java Card System - Standard 2.2 Configuration Protection Profile.

•T.RND
Deficiency of Random Numbers .
An attacker may predict or obtain information about random numbers generated by the TOE for instance because of a lack of entropy of the random numbers provided.
An attacker may gather information about the produced random numbers which might be a problem because they may be used for instance to generate cryptographic keys.
Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the TOE without specific knowledge about the TOE's generator. Malfunctions or premature aging are also considered which may assist in getting information about random numbers.

•T.ACCESS
◦Unauthorized access to sensitive information stored in memories in order to disclose or to corrupt the TOE data. This includes any consequences of bad or incorrect user authentication by the TOE.

•T.OS_OPERATE
◦An attacker modifies the correct Software behavior by unauthorized use of TOE or use of incorrect or unauthorized instructions or commands or sequence of commands, in order to obtain an unauthorized execution of the TOE code.

•T.LEAKAGE
◦An attacker may exploit information which is leaked from the TOE during usage of the Smart Card in order to disclose the Software behavior and Application Data handling (TSF data or User data).
No direct contact with the Smart Card Internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. One example is the Differential Power Analysis (DPA)

•T.FAULT
◦An attacker may cause a malfunction of TSF by applying environmental stress in order to (1) deactivate or modify security features or functions of the TOE or (2) deactivate or modify security functions of the Smart Card. This may be achieved by operating the Smart Card outside the normal operating conditions.

2.4.2. Organizational security policies

This section describes the organizational security policies applied to the TOE:

• P.VERIFICATION
◦This policy is described in Java Card System - Standard 2.2 Configuration Protection Profile, it has been renamed from OSP.VERIFICATION to P.VERIFICATION.

◦ P.ROLES-3
◦The TOE shall recognize the following roles associated with:
- Card Administrator,
- Application Provider

• P.INITIAL_LIFECYCLE_STATE-2
◦Card shall be moved in OP_READY state before any GP function or service is used.
Card shall be issued to Cardholders with the card set to SECURED Life-Cycle state
A security domain shall be moved into the PERSONALIZED life cycle state before any security domain User or Application uses the services of that Security Domain.

• P.CARD_ADMINISTRATOR_PRE-APPROVAL_2b
◦The Card Administrator may allow privileged Application Providers to perform CCMFs.
The Card Administrator shall preauthorize every CCMF (except delete of the Application Provider's own Applications) performed by a privileged Application Provider.
The Card Administrator shall request a confirmation for each delegated CCMF that has taken place.
The Card Administrator shall always be allowed to perform any CCMF for any Application.

• P.APPLICATION_PROVIDER_PRE-APPROVAL_2b
◦The Application Provider allows another (privileged) Application Provider to perform CCMFs for its own Applications as well as personalizing and managing some Application(s) specific data (or keys).

• P.LOAD_FILE_VERIFICATION_3
◦Integrity and authenticity of the Load File shall be verified and shall always be carried out successfully prior to Application Load File installation. This shall take place on-card.

• P.APPLICATION_CODE_VERIFICATION_3
◦Byte code verification and other forms of Application Code Verification is a requirement and shall always be carried out successfully prior to Application Load File on-card installation. This shall take place off card and shall be confirmed by using a Security Domain with Mandated DAP Verification privilege. Application Code Verification shall at least include the algorithms necessary to establish that the Application would pass all omitted runtime checks.

• P.SECURE_COMMUNICATION_1
◦Only the minimum security requirements for GP commands as defined by GPCS are required.

• P.CRYPTO
◦The TOE must provide the following cryptographic functionality to application providers:
- RSA public key asymmetric cryptography
- TRIPLE-DES symmetric cryptography
- RSA signatures.
- SHA-1 hashes
- MD5 hashes

2.4.3. Assumptions

This section describes the assumptions made for the TOE:

•A.VERIFICATION
◦This assumption is described in Java Card System - Standard 2.2 Configuration Protection Profile.

•A.APPLET
◦This policy is described in Java Card System - Standard 2.2 Configuration Protection Profile.

•A.KEY_MANAGEMENT
◦It is assumed that cryptographic keys, which are stored outside the TOE and which are used for secure communication and authentication between Smart Card and terminals are protected in their own (off-card) storage environment.

3. Security objectives

3.1. Security objectives for the TOE

This section describes the security objectives for the TOE.

• O.SID
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.OPERATE
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.RESOURCES
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.FIREWALL
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.REALLOCATION
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.SHRD_VAR_CONFID
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.SHRD_VAR_INTEG
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.ALARM
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.TRANSACTION
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

•O.CIPHER
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.PIN-MNGT
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.KEY-MNGT
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

•O.INSTALL
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.LOAD
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.DELETION
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.OBJ-DELETION
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

•O.REMOTE
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• O.SCP.RECOVERY
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile as OE.SCP.RECOVERY. Since the TOE includes the underlying platform, this objective applies to the TOE instead of the environment.

•O.SCP.SUPPORT
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile as OE.SCP.SUPPORT. Since the TOE includes the underlying platform, this objective applies to the TOE instead of the environment.

• O.SCP.IC
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile as OE.SCP.IC. Since the TOE includes the underlying platform, this objective applies to the TOE instead of the environment.

• O.CARD-MANAGEMENT
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile as OE.CARD-MANAGEMENT. Since the TOE includes the card manager, this objective applies to the TOE instead of the environment.

• O.RND
◦The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have sufficient entropy.
◦ The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.
• O.PROTECT_DATA
◦The TOE shall ensure that sensitive information stored in memories is protected against unauthorized disclosure and any corruption or unauthorized modification.
Moreover, the TOE shall ensure that sensitive information stored in memories is protected against unauthorized access.
The TOE has to provide appropriate security mechanisms to avoid fraudulent access to any sensitive data, such as passwords, cryptographic keys or authentication data.
This is obvious for secret information, but also applies to access controlled information.

• O.OS_OPERATE
◦The TOE must ensure continued correct operation of its security functions.
Especially, the TOE must prevent the unauthorized use of TOE or use of incorrect or unauthorized instructions or commands or sequence of commands.
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

• O.SIDE_CHANNEL
◦The TOE must provide protection against disclosure of confidential data (User Data or TSF data) stored and/or processed in the Smart Card IC:
- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines),
- by measurement and analysis of the time between events found by measuring signals (for example on the power, clock, or I/O lines).
Especially, the software must be designed to avoid interpretations of signals extracted, intentionally or not, from the hardware part of the TOE (for instance, Power Supply, Electro Magnetic emissions).

• O.FAULT_PROTECT

◦The TOE must ensure its correct operation even outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent errors. The environmental conditions may include voltage, clock frequency, temperature, or external energy fields that can be applied on all interfaces of the TOE (physical or electrical).

• O.ROLES-3
◦The TOE shall recognize the following roles associated with:
- Card Administrator
- Application Provider

• O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦The TOE shall allow the Card Administrator to allow privileged Application Providers to perform CCMFs. The Card Administrator shall pre-authorize every CCMF (except delete of the Application Provider's own Applications) performed by a privileged Application Provider. The Card Administrator shall request a confirmation for each delegated CCMF that has taken place. The Card Administrator shall always be allowed to perform any CCMF for any Application.

• O.APPLICATION_PROVIDER_PRE-APPROVAL_2b
◦The TOE shall allow the Application Provider to allow another (privileged) Application Provider to perform CCMFs for its own Applications as well as personalizing and managing some Application(s) specific data (or keys).

• O.LOAD_FILE_VERIFICATION-3
◦The TOE shall provide means to verify the integrity and authenticity of the Load File. This verification shall always be carried out successfully prior to Application Load File installation. This shall take place on-card.

• O.APPLICATION_CODE_VERIFICATION-3
◦Byte code verification and other forms of Application Code Verification is a requirement and shall always be carried out successfully prior to Application Load File on-card installation. This shall take place off card and shall be confirmed by using a Security Domain with Mandated DAP Verification privilege. Application Code Verification shall at least include the algorithms necessary to establish that the Application would pass all omitted runtime checks.

• O.SECURE_COMM_1
◦The TOE shall provide the minimum security requirements for GP commands as defined by GPCS.

3.2. Security objectives for the the environment

This section describes the security objectives for the TOE environment.

• OE.VERIFICATION
◦This objective is described in Java Card System - Standard 2.2 Configuration Protection Profile.

• OE.APPLET
◦This objective is described in Java Card System – Standard 2.2 Configuration Protection Profile.

• OE.KEY_MANAGEMENT
◦During the TOE usage, the terminal or system in interaction with the TOE, shall ensure the protection of their own keys by operational means and/or procedures

• OE.ACTORS_3
◦After Card manufacturing and initialization, the card administrator shall move the Card in the OP_READY state before any GP function or service is used.
The card Issuer shall issue the card to the Cardholders with the card set to SECURED Life-Cycle state.
A security domain shall be moved into the PERSONALIZED life cycle state before any security domain User or Application uses the services of that Security Domain.

• OE.APPLICATION
◦Applications loaded during personalization phases (after the TOE is in OP-READY life cycle state) or Applications loaded after post issuance, are verified and tested off-card before loading.

3.3. Security objectives rationale

This section traces back TOE security objectives to threats and OSPs. Also, TOE environment security objectives are traced back to threats, OSPs and assumptions.

Table 3. Security objectives - Security problem definition mapping

	O.INSTALL	O.LOAD	OE.VERIFICATION	O.CARD-MANAGEMENT	OE.APPLET	O.SHRD_VAR_INTEG	O.SHRD_VAR_CONFID	O.FIREWALL	O.OPERATE	O.ALARM	O.REALLOCATION	O.RESOURCES	O.SID	O.SCP.IC	O.SCP.RECOVERY	O.SCP.SUPPORT	O.CIPHER	O.KEY-MNGT	O.PIN-MNGT	O.TRANSACTION	O.DELETION	O.REMOTE	O.OBJ-DELETION	O.RND
T.PHYSICAL														X										
T.CONFID-JCS-CODE			X	X																				
T.INTEG-APPLI-CODE			X	X																				
T.INTEG-JCS-CODE			X	X																				
T.CONFID-JCS-DATA			X	X				X	X	X			X		X	X								
T.INTEG-JCS-DATA			X	X				X	X	X			X		X	X								
T.CONFID-APPLI-DATA			X	X			X	X	X	X	X		X		X	X	X	X	X	X				
T.INTEG-APPLI-DATA			X	X		X		X	X	X	X		X		X	X	X	X	X	X				
T.SID.1	X			X		X	X	X					X											
T.SID.2	X							X	X				X		X	X								
T.EXE-CODE.1			X					X																
T.EXE-CODE2			X																					
T.NATIVE			X		X																			
T.RESOURCES	X								X			X			X	X								
T.INSTALL	X	X		X																				
T.INTEG-APPLI-CODE.2		X		X																				
T.INTEG-APPLI-DATA.2		X		X																				
T.DELETION				X																	X			
T.EXE-CODE-REMOTE																						X		
T.OBJ-DELETION																							X	
T.RND																								X

P.VERIFICATION			X																			
P.CRYPTO																X						
A.APPLET					X																	
A.VERFICATION			X																			

	O.PROTECT_DATA	*O.OS_OPERATE*	*O.SIDE_CHANNEL*	*O.FAULT_PROTECT*	*O.ROLES-3*	*O.CARD_ADMINISTRATOR_PRE-APPROVAL_3*	*O.APPLICATION_PROVIDER_PRE-APPROVAL_2B*	*O.LOAD_FILE_VERIFICATION-3*	*O.APPLICATION_CODE_VERIFICATION-3*	*O.SECURE_COMM_1*	*OE.KEY_MANAGEMENT*	*OE.ACTORS_3*	*OE.APPLICATION*
A.VERIFICATION													X
A.KEY_MANAGEMENT											X		
T.INTEG-JCS-DATA		X											
T.CONFID-APPLI-DATA		X											
T.INTEG-APPLI-DATA		X											
T.SID.2		X											
T.RESOURCES		X											
T.ACCESS	X												
T.OS_OPERATE		X											
T.LEAKAGE			X										
T.FAULT				X									
P.ROLES-3					X								
P.INITIAL_LIFECYCLE_STATE-2												X	
P.CARD_ADMINISTRATOR_PRE-APPROVAL_2b						X							
P.APPLICATION_PROVIDER_PRE-APPROVAL_2b							X						
P.LOAD_FILE_VERIFICATION_3								X					
P.APPLICATION_CODE_VERIFICATION_3									X				
P.SECURE_COMMUNICATION_1										X			

- *T.PHYSICAL - O.SCP.IC*
  - Directly covered by the strength of the underlying IC.

- *T.CONFID-JCS-CODE - OE.VERIFICATION*

◦Threats concerning the confidentiality of code are directly countered by OE.VERIFICATION.

•*T.CONFID-JCS-CODE - O.CARD-MANAGEMENT*
◦Since the card manager controls the access to card management functions, confidentiality of JCS code is preserved.

•*T.INTEG-APPLI-CODE - OE.VERIFICATION*
◦Threats concerning the integrity of code are directly countered by OE.VERIFICATION.

•*T.INTEG-APPLI-CODE - O.CARD-MANAGEMENT*
◦Since the card manager controls the access to card management functions, integrity of application code is preserved.

•*T.INTEG-JCS-CODE - OE.VERIFICATION*
◦Threats concerning the integrity of code are directly countered by OE.VERIFICATION.

•*T.INTEG-JCS-CODE - O.CARD-MANAGEMENT*
◦Since the card manager controls the access to card management functions, integrity of JCS code is preserved.

•*T.CONFID-JCS-DATA - OE.VERIFICATION*
◦Threats concerning the confidentiality of code are directly countered by OE.VERIFICATION.

•*T.CONFID-JCS-DATA - O.CARD-MANAGEMENT*
◦Since the card manager controls the access to card management functions, confidentiality of JCS data is preserved.

•*T.CONFID-JCS-DATA - O.FIREWALL*
◦The threats concerning confidentiality of data are countered by the isolation commitments stated in the O.FIREWALL objective.

•*T.CONFID-JCS-DATA - O.OPERATE*
◦Since the firewall is dynamically enforced and it shall never stop operating as stated in O.OPERATE objective, the objective protects confidentiality of JCS data.

•*T.CONFID-JCS-DATA - O.ALARM*
◦As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, O.ALARM protects confidentiality of JCS data.

•*T.CONFID-JCS-DATA - O.SID*
◦Since each subject must be uniquely identified when operating, this objective protects confidentiality of JCS data.

•*T.CONFID-JCS-DATA - O.SCP.RECOVERY*
◦Helps in covering this threat by allowing the TOE to complete the interrupted operation successfully, or to recover to a consistent and secure state after a cut of power so JCS data confidentiality is preserved.

•*T.CONFID-JCS-DATA - O.SCP.SUPPORT*
◦Helps in covering this threat by providing functionalities that support the well-functioning of the TSFs of the TOE and by controlling the access to information proper of the TSFs.

•*T.INTEG-JCS-DATA - OE.VERIFICATION*
◦Since this OE.VERIFICATION enforces that all the bytecodes shall be verified at least once in order to ensure that each bytecode is valid at execution time, the objectives protects the integrity of JCS data.

•*T.INTEG-JCS-DATA - O.CARD-MANAGEMENT*
◦Since the card manager controls the access to card management functions, integrity of JCS data is preserved.

•*T.INTEG-JCS-DATA - O.FIREWALL*

◦Since the TOE shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and TSFs, this objective protects integrity of JCS data

•*T.INTEG-JCS-DATA - O.OPERATE*
◦Since the firewall is dynamically enforced and it shall never stop operating as stated in O.OPERATE objective, the objective protects integrity of JCS data.

•*T.INTEG-JCS-DATA - O.ALARM*
◦As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, the objective protects integrity of JCS data.

•*T.INTEG-JCS-DATA - O.SID*
◦Since each subject must be uniquely identified when operating, this objective protects integrity of JCS data.

•*T.INTEG-JCS-DATA - O.SCP.RECOVERY*
◦Helps in covering this threat by allowing the TOE to return to a safe state after a cut of power so JCS data integrity is preserved.

•*T.INTEG-JCS-DATA - O.SCP.SUPPORT*
◦Helps by supporting the well-functioning of the TSFs of the TOE (avoiding they are bypassed or altered) and by controlling the access to information proper of the TSFs so JCS data integrity is preserved.
•*T.INTEG-JCS-DATA - O.OS_OPERATE*
◦Since O.OS_OPERATE requires to ensure continue correct operation of the TOE regarding incorrect usage or power loss with recovery of a secure state, an attacker is not able to alter Java Card System or API data.

•*T.CONFID-APPLI-DATA - OE.VERIFICATION*
◦Since all bytecodes shall be verified at least once to ensure that each bytecode is valid at execution time as stated in the OE.VERIFICATION, the objective protects confidentiality of application data.
•
•*T.CONFID-APPLI-DATA - O.CARD-MANAGEMENT*
◦Since the card manager controls the access to card management functions, confidentiality of application data is preserved.

•*T.CONFID-APPLI-DATA - O.SHRD_VAR_CONFID*
◦Since any data container that is shared by all applications is always cleaned as stated in O.SHRD_VAR_CONFID, the objective protects confidentiality of application data.

•*T.CONFID-APPLI-DATA - O.FIREWALL*
◦Since the TOE shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and TSFs, O.FIREWALL protects confidentiality of application data.

•*T.CONFID-APPLI-DATA - O.OPERATE*
◦Since the firewall is dynamically enforced and it shall never stop operating as stated in O.OPERATE objective, the objective protects confidentiality of application data.

•*T.CONFID-APPLI-DATA - O.ALARM*
◦As the firewall is a software tool automating critical controls, the objective O.ALARM asks for it to provide clear warning and error messages, O.ALARM protects confidentiality of application data.

•*T.CONFID-APPLI-DATA - O.REALLOCATION*
◦Since the re-allocation of a memory block for runtime areas of the Java Card VM shall not disclose any information that was previously stored in that block, O.REALLOCATION protects confidentiality of application data.

•*T.CONFID-APPLI-DATA - O.SID*
◦Since each subject must be uniquely identified when operating, O.SID protects confidentiality of application data.

•*T.CONFID-APPLI-DATA - O.SCP.RECOVERY*
◦Helps in covering this threat by allowing the TOE to return to a safe state after a cut of power so application

data confidentiality is preserved.

- *T.CONFID-APPLI-DATA - O.SCP.SUPPORT*
  - Helps by supporting the well-functioning of the TSFs of the TOE (avoiding they are bypassed or altered) and by controlling the access to information proper of the TSFs so application data confidentiality is preserved.

- *T.CONFID-APPLI-DATA - O.CIPHER*
  - This objective protects confidentiality of application data by providing cryptographic functions.

- *T.CONFID-APPLI-DATA - O.KEY-MNGT*
  - Since the cryptographic Key objects are managed securely, the unauthorized access to Key objects belonging to another application is protected.

- *T.CONFID-APPLI-DATA - O.PIN-MNGT*
  - Since the PIN objects are managed securely, the unauthorized access to PIN objects belonging to another application is protected.

- *T.CONFID-APPLI-DATA - O.TRANSACTION*
  - Since atomic execution of a set of operations is provided, the unauthorized access to data belonging to another application is protected.

- *T.CONFID-APPLI-DATA – O.OS_OPERATE*
  - Since O.OS_OPERATE require*s* to ensure continue correct operation of the TOE regarding incorrect usage or power loss with recovery of a secure state, an attacker is not able to alter another application's data.

- *T.INTEG-APPLI-DATA - OE.VERIFICATION*
  - Since all bytecodes shall be verified at least once to ensure that each bytecode is valid at execution time as stated in the OE.VERIFICATION, the objective protects integrity of application data.
- *T.INTEG-APPLI-DATA - O.CARD-MANAGEMENT*
  - Since the card manager controls the access to card management functions, integrity of application data is preserved.

- *T.INTEG-APPLI-DATA - O.SHRD_VAR_INTEG*
  - Since writing access to a data memory area that is shared by all applications shall be ensured to only the currently selected application, this objective prevents from altering another application's data.

- *T.INTEG-APPLI-DATA - O.FIREWALL*
  - Since the TOE shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and TSFs, this objective protects integrity of application data.

- *T.INTEG-APPLI-DATA - O.OPERATE*
  - Since the firewall is dynamically enforced and it shall never stop operating as stated in O.OPERATE objective, the objective protects integrity of application data.

- *T.INTEG-APPLI-DATA - O.ALARM*
  - This objective provides security alarm upon detection of altering (part of) another application's data.

- *T.INTEG-APPLI-DATA - O.REALLOCATION*
  - Since the re-allocation of a memory block for runtime areas of the Java Card VM shall not disclose any information that was previously stored in that block, this objective protects integrity of application data.

- *T.INTEG-APPLI-DATA - O.SID*
  - Since each subject must be uniquely identified when operating, this objective protects integrity of application data.
- *T.INTEG-APPLI-DATA - O.SCP.RECOVERY*
  - Helps in covering this threat by allowing the TOE to return to a safe state after a cut of power so application data integrity is preserved.
-

•*T.INTEG-APPLI-DATA - O.SCP.SUPPORT*
◦Helps by supporting the well-functioning of the TSFs of the TOE (avoiding they are bypassed or altered) and by controlling the access to information proper of the TSFs so application data integrity is preserved.

•*T.INTEG-APPLI-DATA - O.CIPHER*
◦This objective protects the altering (part of) another application's data by ciphering sensitive data for applications.

•*T.INTEG-APPLI-DATA - O.KEY-MNGT*
◦Since the cryptographic Key objects are managed securely, the altering (part of) Key objects belonging to another application is protected.

•*T.INTEG-APPLI-DATA - O.PIN-MNGT*
◦Since the PIN objects are managed securely, the altering (part of) PIN objects belonging to another application is protected.

•*T.INTEG-APPLI-DATA - O.TRANSACTION*
◦Since atomic execution of a set of operations is provided, the altering (part of) another application's data is protected.

•*T.INTEG-APPLI-DATA – O.OS_OPERATE*
◦Since O.OS_OPERATE requires to ensure continue correct operation of the TOE regarding incorrect usage or power loss with recovery of a secure state, an attacker is not able to alter another application’s data.

•*T.SID.1 - O.INSTALL*
◦Since this objective prevents any malicious installation of an applet on the card which can result in usurpation of identity, it helps in preventing impersonation.

•*T.SID.1 - O.CARD-MANAGEMENT*
◦Contributes by preventing usurpation of identity resulting from a malicious installation of an applet on the card.

•*T.SID.1 - O.SHRD_VAR_INTEG*
◦Since writing access to a data memory area that is shared by all applications shall be ensured to only the currently selected application, this objective prevents from impersonation as the currently selected application.

•*T.SID.1 - O.SHRD_VAR_CONFID*
◦Since any data container that is shared by all applications is always cleaned after the execution of an application, this objective prevents disclosing the installation parameters of an applet.

•*T.SID.1 - O.FIREWALL*
◦Since the TOE shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and TSFs, this objective prevents disclosing and modifying some assets for impersonation.

•*T.SID.1 - O.SID*
◦Since each subject must be uniquely identified when operating, this objective helps in preventing impersonation.

•*T.SID.2 - O.INSTALL*
◦Since this objective contributes to counter this threat for what relates to the critical phase of applet installation which is in charge of the installer with special right to access objects on the card, a part of this threat is covered.

T.SID.2 - O.FIREWALL
◦Since the TOE shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and TSFs, a part of this threat is covered.

•*T.SID.2 - O.OPERATE*

◦Since the TOE must ensure continued correct operation of its security functions, the objective prevents to modify the TOE'a attribution of a privileged role.

•*T.SID.2 - O.SID*
◦Since each subject must be uniquely identified when operating, a part of this threat is covered.

•*T.SID.2 - O.SCP.RECOVERY*
◦Helps in covering this threat by allowing the TOE to return to a safe state after a cut of power so an attacker is not able to impersonate another role.

•*T.SID.2 - O.SCP.SUPPORT*
◦Helps by supporting the well-functioning of the TSFs of the TOE (avoiding they are bypassed or altered) and by controlling the access to information proper of the TSFs so an attacker is not able to impersonate another role.

•*T.SID.2 - O.OS_OPERATE*
Since O.OS_OPERATE requires to ensure continue correct operation of the TOE regarding incorrect usage or power loss with recovery of a secure state, an attacker is not able to impersonate another role.

•*T.EXE-CODE.1 - OE.VERIFICATION*
◦Since all bytecodes shall be verified at least once to ensure that each bytecode is valid at execution time as stated in the OE.VERIFICATION, unauthorized execution of a method is prevented.

•*T.EXE-CODE.1 - O.FIREWALL*
◦Since the TOE shall ensure controlled sharing of data containers owned by applets of different packages, and between applets and TSFs, this objective prevents the execution of non-sharable methods of a class instance by any subject apart from the class instance owner.

•*T.EXE-CODE.2 - OE.VERIFICATION*
◦Since all bytecodes shall be verified at least once to ensure that each bytecode is valid at execution time as stated in the OE.VERIFICATION, unauthorized execution of a method fragment or arbitrary data is prevented.

•*T.NATIVE - OE.VERIFICATION*
◦Since all bytecodes shall be verified at least once to ensure that each bytecode is valid at execution time as stated in the OE.VERIFICATION, the program counter of an applet is prevented to jump into a piece of native code by confining the control flow to the currently executed method.

•*T.NATIVE - OE.APPLET*
◦Since an application can not download its own native code on the card as stated in OE.APPLET, the application is prevented to execute a method fragment or arbitrary data.

•*T.RESOURCES - O.INSTALL*
◦Since this objective ensures correct consumption of resources during installation phase and other card management operations in case of failure, it helps in preventing malicious consumption of resources during installation.

•*T.RESOURCES - O.OPERATE*
◦Since the TOE must ensure continued correct operation of its security functions, the objective prevents to consume some resources by an attacker.

•*T.RESOURCES - O.RESOURCES*
◦Since the TOE shall control the availability of resources of applications, this objective protects some resources of the card from being consumed by an attacker.

•*T.RESOURCES - O.SCP.RECOVERY*
◦Helps in covering this threat by allowing the TOE to return to a safe state after a cut of power so an attacker is not able to consume more resources.

•*T.RESOURCES - O.SCP.SUPPORT*
◦Helps by supporting the well-functioning of the TSFs of the TOE (avoiding they are bypassed or altered) and by controlling the access to information proper of the TSFs so an attacker is not able to consume more resources.

•*T.RESOURCES – O.OS_OPERATE*
◦Since O.OS_OPERATE requires to ensure continue correct operation of the TOE regarding incorrect usage or power loss with recovery of a secure state, an attacker is not able to prevent correct operation of TOE through consumption of some resources of the card.

•*T.INSTALL - O.INSTALL*
◦Since the installer verifies each and every attempt to install an applet in terms of integrity and authenticity, a part of this threat is covered.

•*T.INSTALL - O.LOAD*
◦Since this objective ensures that any malicious attempt to load an unauthorized applet into the card will not be allowed in loading phase, it covers a part of this threat.

•*T.INSTALL - O.CARD-MANAGEMENT*
◦Contributes to preventing installation by unauthorized subjects.

•*T.INTEG-APPLI-CODE.2 - O.LOAD*
◦Since each application code should be verified when an application package is transmitted to the card for installation, this objective ensures the integrity of application code.

•*T.INTEG-APPLI-CODE.2 - O.CARD-MANAGEMENT*
◦Contributes to cover all the threats on confidentiality and integrity of code and data.

•*T.INTEG-APPLI-DATA.2 - O.LOAD*
◦Since each application data should be verified when an application package is transmitted to the card for installation, this objective ensures the integrity of application data.

•*T.INTEG-APPLI-DATA.2 - O.CARD-MANAGEMENT*
◦Contributes to cover all the threats on confidentiality and integrity of code and data.

•*T.DELETION - O.CARD-MANAGEMENT*
◦Contributes to preventing deletion by unauthorized subjects.

•*T.DELETION - O.DELETION*
◦Since each deletion of applet or package must be securely managed, this treat is covered.

•*T.EXE-CODE-REMOTE - O.REMOTE*
◦Since the TOE provides only restricted remote access from the CAD to the services implemented by the applets on the card, this objective contributes to prevent the invocation of a method that is not supposed to be accessible from outside the card.

•*T.OBJ-DELETION - O.OBJ-DELETION*
◦Since the object deletion process ensures secure deallocation of objects by making no references to deleted object to be accessible, this threat is covered.

•*T.RND – O.RND*
◦O.RND directly counters T.RND

•*P.VERIFICATION - OE.VERIFICATION*
◦Since all bytecodes shall be verified at least once to ensure that each bytecode is valid at execution time as stated in the OE.VERIFICATION, this OSP is directly covered.

•*P.CRYPTO – O.CIPHER*
P.CRYPTO details the list of cryptographic algorithms which have to be provided according to O.CIPHER.

•*A.APPLET- OE.APPLET*
◦Since an application can not download its own native code on the card as stated in OE.APPLET, this assumption is covered.

•*A.VERFICATION - OE.VERIFICATION*
◦Since all bytecodes shall be verified at least once to ensure that each bytecode is valid at execution time as stated in the OE.VERIFICATION, this assumption is covered.

•*A.VERFICATION - OE.APPLICATION*
◦A.VERIFICATION is covered by OE.APPLICATION that requires off-card verification and testing of applications loaded during personalization phases (after the TOE is in OP-READY life cycle state) or after post issuance before loading.

•*A.KEY_MANAGEMENT- OE.KEY_MANAGEMENT*
◦A.KEY_MANAGEMENT is covered by OE.KEY_MANAGEMENT which addresses protection of keys owned by the terminal or system interaction with the TOE during the TOE usage.

•*T.ACCESS - O.PROTECT_DATA*
◦T.ACCESS is covered by objective O.PROTECT_DATA which addresses the protection of data stored in the TOE from unauthorized disclosure and any corruption or unauthorized modification.

•*T.OS_OPERATE - O.OS_OPERATE*
◦T.OS_OPERATE is covered by objective O.OS_OPERATE which requires to ensure continue correct operation of the TOE regarding incorrect usage or power loss with recovery of a secure state.

•*T.LEAKAGE - O.SIDE_CHANNEL*
◦T.LEAKAGE is covered by O.SIDE_CHANNEL that requires protection against disclosure of confidential data by measurements and analysis of signals emitted by the TOE.

•*T.OBJ-DELETION - O.OBJ-DELETION*
◦Since the object deletion process ensures secure deallocation of objects by making no references to deleted object to be accessible, this threat is covered.

•*T.FAULT - O.FAULT_PROTECT*
◦T.FAULT is covered by O.FAULT_PROTECT that requires correct behavior of the TOE when operating outside the normal range.

•*P.ROLES-3- O.ROLES-3*
◦P.ROLES-3 is directly countered by O.ROLES-3.

•*P.INITIAL_LIFECYCLE_STATE-2 - OE.ACTORS-3*
◦P.INITIAL_LIFECYCLE_STATE-2 is coved by OE.ACTORS-3 as operation that is performed by card manufacturer in personalization phase.

•*P.CARD_ADMINISTRATOR_PRE-APPROVAL_2b - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3*
◦P.CARD_ADMINISTRATOR_PRE-APPROVAL_2b is directly countered by O.CARD_ADMINISTRATOR_PRE-APPROVAL_3

•*P.APPLICATION_PROVIDER_PRE-APPROVAL_2b - O.APPLICATION_PROVIDER_PRE-APPROVAL_2b*
◦P.APPLICATION_PROVIDER_PRE-APPROVAL_2b is directly countered by O.APPLICATION_PROVIDER_PRE-APPROVAL_2b

•*P.LOAD_FILE_VERIFICATION_3 - O.LOAD_FILE_VERIFICATION-3*
◦P.LOAD_FILE_VERIFICATION_3 is directly countered by O.LOAD_FILE_VERIFICATION-3

•*P.APPLICATION_CODE_VERIFICATION_3 - O.APPLICATION_CODE_VERIFICATION-3*
◦P.APPLICATION_CODE_VERIFICATION_3 is directly countered by O.APPLICATION_CODE_VERIFICATION-3

•*P.SECURE_COMMUNICATION_1 - O.SECURE_COMM_1*

◦P.SECURE_COMMUNICATION_1 is directly countered by O.SECURE_COMM_1

4. Security requirements

Operations performed on Security requirements are identified using the following conventions:

Assignment	*[assignment:* **assignment-done***]*
Selection	*[selection:* **selection-done***]*
Iteration	SFRid/Iteration name
Refinement	**Refinement***: refined content*

4.1. Security functional requirements

This section describes the security functional requirements using terms defined in the previous section. The grouping approach of the Javacard protection profile has been used.

4.1.1. *CoreG* Security functional requirements

This group is focused on the main security policy of the Java Card System, known as the *firewall*. This policy essentially concerns the security of installed *applet*s. The policy focuses on the execution of bytecodes.

4.1.1.1. 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, O.JAVAOBJECT* and all operations among subjects and objects covered by the SFP.

Subjects (prefixed with an "*S*") and objects (prefixed with an "*O*") covered by this policy are:

Subject/Object	Description
S.PACKAGE	Any *package*, which is the security unit of the firewall policy.
S.JCRE	The *Java Card RE*. This is the process that manages *applet* selection and deselection, along with the delivery of *APDU*s from and to the smart card device. *This subject is unique.*
O.JAVAOBJECT	Any object. Note that KEYS, PIN, arrays and *applet* instances are specific objects in the Java programming language.

Operations (prefixed with "*OP*") of this policy are described in the following table. Each operation has a specific number of parameters given between brackets, among which there is the "**accessed object**", the first one, when applicable. Parameters may be seen as security attributes that are under the control of the subject performing the operation. Operations stand for bytecodes supported by the Java Card platform.

Operation	Description
OP.ARRAY_ACCESS(O.JAVAOBJECT, field)	Read/Write an array component.
OP.INSTANCE_FIELD(O.JAVAOBJECT, field)	Read/Write a field of an instance of a class in the Java programming *language*
OP.INVK_VIRTUAL(O.JAVAOBJECT, method, arg1, *…)*	Invoke a virtual method (either on a class instance or an array object)
OP.INVK_INTERFACE(O.JAVAOBJECT, method, arg1,*…)*	Invoke an *interface* method.

OP.THROW(O.JAVAOBJECT)	Throwing **of** an object (**athrow**).
OP.TYPE_*ACCESS(O.JAVAOBJECT,* class*)*	Invoke **checkcast** or **instanceof** on an object**.**
OP.JAVA(…)	Any access in the sense of [JCRE21], §6.2.8. In our formalization, this is one of the preceding operations.
*OP.CREATE(*Sharing, LifeTime*)*	Creation of an object (**new** or **makeTransient** call).

Note that accessing array's components of a **static** array, and more generally fields and methods of **static** *objects,* is an access to the corresponding *O.JAVAOBJECT*.

FDP_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.

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: ***subjects, objects and their security attributes, described hereafter. The TSF shall enforce the FIREWALL access control SFP to objects*** based on the following: ***(1) the security attributes of the covered subjects and objects, (2) the currently active context, (3) the SELECTed applet Context, and (4) the attribute ActiveApplets, which is a list of the active applets' AIDs.***

The following table describes which security attributes are attached to which subject/object.

Subject/Object	**Attributes**
S.PACKAGE	Context, Selected, Active, Selection Status
S.JCRE	Active
O.JAVAOBJECT	Sharing, Context, LifeTime

The following table describes the possible values for each security attribute.

Name	**Description**
Context	*Package AID* , or "Java Card RE"
Sharing	Standard, SIO, Java Card RE entry point, or global array
LifeTime	**CLEAR_ON_DESELECT** or **PERSISTENT**.
Selected, Active	*Boolean value: true or false*
Selection Status	Multiselectable, Non-multiselectable or "None"
ActiveApplets	List of package's *AID*s

In the case of an array type, we state that 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,
•*Java Card RE entry points* (Temporary or Permanent), who have freely accessible methods but protected fields,
•Global arrays, having both unprotected fields (including components; refer to JavaCardClass discussion above) and methods.

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 ([JCRE21], §6.1.2). An object is owned by an *applet* instance, by the *Java Card RE* or by the *package* library where it has been defined (these latter objects can only be arrays that initialize static fields of packages).

Finally both "the currently active context" and "the SELECTed applet context" are security attributes internal to the Java Card VM.

> ([JCRE21], Glossary) *Currently selected applet***.** 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 currently selected applet. The Java Card RE sends all APDU commands to the currently selected applet.

While the expression "selected applet" 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 package whose *Selected* attribute holds *true*.

> ([JCRE21] §6.1.1) At any point in time, there is only **one active context** within the Java Card VM (this is called the *currently active context*).

In this policy, the "active context" is the package or Java Card RE whose *Active* security attribute holds *true*.

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

The Java Card platform, version 2.2, 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 applet*s. *Applet*s that belong to a same *package* are either all multiselectable or not ([JCVM22],§2.2.5). Therefore, the selection mode can be regarded as an attribute of *package*s. No selection mode is defined for a library *package*.

Support for multiple *logical channels* (with multiple selected applet instances) requires a change to the *selected applet* concept as stated in Java Card System, version 2.1.1.. Since more than one applet instance can be selected at the same time, and one *applet* instance can be selected on different *logical channels* simultaneously, it is necessary to differentiate the state of the applet instances in more detail. 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. ([JCRE22],§4).

The ActiveApplets security attribute is internal to the Java Card VM, that is, not attached to any specific object or subject of the SPM. The attribute is TSF data that plays a role in the SPM.

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 by the ***FIREWALL SFP***:

R.JAVA.1 ([JCRE21]§6.2.8) The currently active *S.PACKAGE* may freely perform any of *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 **"***Java Card RE entry point* **"** or **"global array".**

R.JAVA.2 ([JCRE21]§6.2.8) The currently active *S.PACKAGE* may freely perform any of *OP.ARRAY_ACCESS,OP.INSTANCE_FIELD, OP.TYPE_ACCESS, 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 ([JCRE21]§6.2.8.10) The currently active *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.5 The currently active *S.PACKAGE* may perform an *OP.CREATE* only if the value of the Sharing parameter is "Standard".

R.JAVA.20 ([JCRE22], §6.2.8.6,) An *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 one of the following applies:

1. The value of the attribute Selection Status of the package whose AID is **"***Package AID***"** is **"Multiselectable»,**
2. 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 ActiveApplets,

and in either of the cases above the invoked *interface* method extends the **Shareable** *interface*.

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

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

FDP_ACF.1.4/FIREWALL The TSF shall explicitly deny access of subjects to objects based on the 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 with OP.CREATE and a "CLEAR_ON_DESELECT" LifeTime parameter if the active context is not the same as the SELECTed applet

Context.

FDP_IFC.1/JCVM SUBSET INFORMATION FLOW CONTROL

FDP_IFC.1.1/JCVM The TSF shall enforce the ***JCVM information flow control SFP*** on ***the following subjects, information and operations***.

Subjects (prefixed with an "*S*") and information (prefixed with an "*I*") covered by this policy are:

Subject/Information	Description
S.LOCAL	Operand stack of a Java Card VM frame, or local variable of a Java Card VM frame containing an object or an array of references.
S.MEMBER	Any object's field, **static** field or array position.
I.DATA	Java Card VM Reference Data: *objectref addresses of temporary Java Card RE Entry Point objects and global arrays*.

There is a unique operation in this policy:

Operation	Description
OP.PUT(S1, S2, I)	Transfer a piece of information *I* from *S1* to *S2*.

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: ***(1) the 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 rule holds:

An operation OP.PUT(S1, S.MEMBER, I) is allowed if and only if the currently active context is "Java Card RE"; other OP.PUT operations are allowed regardless of the active context's value.

FDP_IFF.1.3/JCVM The TSF shall enforce ***[assignment:*** **none*]***.

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

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

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***.

FMT_MSA.1/JCRE MANAGEMENT OF SECURITY ATTRIBUTES

FMT_MSA.1.1/JCRE The TSF shall enforce the FIREWALL access control SFP and the JCVM information flow control SFP to restrict the ability to modify the active context, the SELECTed applet Context and the ActiveApplets security attributes to the Java Card RE (S.JCRE).

FMT_MSA.2/JCRE SECURE SECURITY ATTRIBUTES

FMT_MSA.2.1/JCRE The TSF shall ensure that only secure values are accepted for security attributes.

FMT_MSA.3/FIREWALL STATIC ATTRIBUTE INITIALIZATION

FMT_MSA.3.1/FIREWALL The TSF shall enforce the ***FIREWALL access control SFP*** *and 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/FIREWALL The TSF shall allow the following role(s) to specify alternative initial values to override the default values when an object or information is created: **none**.

FMT_SMR.1/JCRE SECURITY ROLES

FMT_SMR.1.1/JCRE The TSF shall maintain the roles: *the* ***Java Card RE***.

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

FMT_SMF.1/JCRE SPECIFICATION OF MANAGEMENT FUNCTIONS

FMT_SMF.1.1/JCRE The TSF shall be capable of performing the following management functions:
[assignment:
- ***Modify the active context and the SELECTed applet Context.***
- ***Modify the list of registered applets'*** *AID*.
]

4.1.1.2. Application Programming Interface

The following SFRs are related to the Java Card API.

FCS_CKM.1/**RSA** CRYPTOGRAPHIC KEY GENERATION

FCS_CKM.1.1/RSA The TSF shall generate cryptographic KEYS in accordance with a specified cryptographic KEY generation algorithm [assignment: **specified in [ESI]** and specified cryptographic KEY sizes [assignment: **multiple of 256 bits from 1024 bits to 2048 bits**] that meet the following: [assignment: **ISO/IEC 9796-1, Annex A, sections A.4 and A.5, and Annex C**].

FCS_CKM.1/**TRIPLE-DES** CRYPTOGRAPHIC KEY GENERATION

FCS_CKM.1.1/TRIPLE-DES The TSF shall generate cryptographic KEYS in accordance with a specified cryptographic KEY generation algorithm [assignment: **using the RNG**] and specified cryptographic KEY sizes [assignment: **128 bits and 192 bits**] that meet the following: [assignment: **none**].

FCS_CKM.2/**RSA** CRYPTOGRAPHIC KEY DISTRIBUTION

FCS_CKM.2.1/RSA The TSF shall distribute cryptographic KEYS in accordance with a specified cryptographic KEY distribution method [assignment: **setExponent and setModulus for RSAin javacard.security.RSAPrivateKey javacard.security.RSAPublicKey class**] that meets the following: [assignment: **[JCAPI221]**].

FCS_CKM.2/**TRIPLE-DES** CRYPTOGRAPHIC KEY DISTRIBUTION

FCS_CKM.2.1/TRIPLE-DES The TSF shall distribute cryptographic KEYS in accordance with a specified cryptographic KEY distribution method [assignment: **setKey() method in javacard.security.DESKey class**] that meets the following: [assignment: **[JCAPI221]**].

FCS_CKM.3/**TRIPLE-DES** CRYPTOGRAPHIC KEY ACCESS

FCS_CKM.3.1/TRIPLE-DES The TSF shall perform [assignment: **management of DES keys**] in accordance with a specified cryptographic KEY access method [assignment: **methods defined in packages javacard.security and javacardx.crypto of [JCAPI221]**] that meets the following: [assignment: **[JCAPI221]**].

FCS_CKM.3/**RSA** CRYPTOGRAPHIC KEY ACCESS

FCS_CKM.3.1/RSA The TSF shall perform [assignment: **management of RSA keys**] in accordance with a specified cryptographic KEY access method [assignment: **methods defined in packages javacard.security and javacardx.crypto of [JCAPI221]**] that meets the following: [assignment: **[JCAPI221]**].

FCS_CKM.4 CRYPTOGRAPHIC KEY DESTRUCTION

FCS_CKM.4.1 The TSF shall destroy cryptographic KEYS in accordance with a specified cryptographic KEY destruction method [assignment: **physically irreversible destruction of the stored key by method clearKey of [JCAPI221]**] that meets the following: [assignment: **none**].

FCS_COP.1/**TRIPLE-DES** CRYPTOGRAPHIC OPERATION

FCS_COP.1.1/TRIPLE-DES The TSF shall perform [assignment: **encryption and decryption**] in accordance with a specified cryptographic algorithm [assignment: **DES in**

CBC/ECB mode] and cryptographic KEY sizes [assignment: **128 bits and 192 bits**] that meet the following: [assignment: **[FIPS46-3]**].

FCS_COP.1/**RSA** CRYPTOGRAPHIC OPERATION

FCS_COP.1.1/RSA

The TSF shall perform [assignment: **encryption and decryption**] in accordance with a specified cryptographic algorithm [assignment: **RSA**] and cryptographic KEY sizes [assignment: **multiple of 32 bits from 1024 bits to 2048 bits**] that meet the following: [assignment: **[PKCS1v1.5]**].

FCS_COP.1/**DESMAC** CRYPTOGRAPHIC OPERATION

FCS_COP.1.1/DESMAC

The TSF shall perform [assignment: **8 byte MAC generation and verification**] in accordance with a specified cryptographic algorithm [assignment: **DES in CBC mode**] and cryptographic KEY sizes [assignment: **128 bits and 192 bits**] that meet the following: [assignment: **ISO/IEC 9797-1**].

FCS_COP.1/**RSA SIGNATURE** CRYPTOGRAPHIC OPERATION

FCS_COP.1.1/RSA SIGNATURE

The TSF shall perform [assignment: **digital signature generation and verification**] in accordance with a specified cryptographic algorithm [assignment: **RSA with SHA-1**] and cryptographic KEY sizes [assignment: ***1024, 1280, 1536, 1984 and 2048 bits***] that meet the following: [assignment: ***[PKCS1v1.5] and ISO9796***].

FCS_COP.1/**SHA-1** CRYPTOGRAPHIC OPERATION

FCS_COP.1.1/SHA-1

The TSF shall perform [assignment: **secure hash computation**] in accordance with a specified cryptographic algorithm [assignment: **SHA-1**] and cryptographic KEY sizes [assignment: **none**] that meet the following: [assignment: **[FIPS180-1]**].

FCS_COP.1/**MD5** CRYPTOGRAPHIC OPERATION

FCS_COP.1.1/MD5

The TSF shall perform [assignment: **secure hash computation**] in accordance with a specified cryptographic algorithm [assignment: **MD5**] and cryptographic KEY sizes [assignment: **none**] that meet the following: [assignment: **[RFC1321]**].

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 object:
the APDU buffer.

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 *de-allocation of the resource from* the following object*:* the bArray object.

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 *de-allocation of the resource* from the following objects*:* *any transient object.*

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 *de-allocation of the resource from* the following objects*: any reference to an object instance created during an aborted transaction.*

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 de-allocation of the resource from the following objects: the cryptographic buffer (**D.CRYPTO**).

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 ***OP.JAVA, OP.CREATE on O.JAVAOBJECTs.***
FDP_ROL.1.2/FIREWALL	The TSF shall permit operations to be rolled back within the [assignment: **scope of a select(), deselect(), process() or install() call, notwithstanding the restrictions given in [JCRE221], §7.7, within the bounds of the Commit Capacity ([JCRE221], §7.8), and those described in [JCAPI221]**].

4.1.1.3. Card Security Management

The following SFRs are related to the security requirements at the level of the whole card, in contrast to the previous ones, that are somewhat restricted to the TOE alone. For instance, a potential security violation detected by the Java Card virtual machine may require a reaction that does not only concern the virtual machine, such as blocking the card (or request the appropriate security module with the power to block the card to perform the operation).

FAU_ARP.1/JCS SECURITY ALARMS

FAU_ARP.1.1/JCS	The TSF shall ***throw an exception, lock the card session or reinitialize the Java Card System and its data [assignment:*** **none*]*** upon detection of a potential security violation.

REFINEMENT Potential security violation is refined to 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 or RF signal loss) and power failure
•Abortion of a transaction in an unexpected context (see (**abortTransaction()**, [JCAPI21] and ([JCRE21], §7.6.2)
•Violation of the Firewall or Java Card VM SFPs
•Unavailability of resources
•Array overflow
•Other runtime errors related to *applet*'s failure (like uncaught exceptions)

FDP_SDI.2 STORED DATA INTEGRITY MONITORING AND ACTION

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: integrity errors] on all objects, based on the following attributes: [assignment: **checksum attribute of D.APP_I_DATA, D.PIN and D.APP_KEYs**].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [assignment: **send a error message and reset itself**].

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 (shared between the card manager and the TOE), the bytecode and its data arguments (shared with *applet*s and API *package*s), when shared between the TSF and another trusted IT product.
Refinement:
The sharing between the TOE and the card manager refers to the sharing between the card manager and the Javacard System.

FPT_TDC.1.2 The TSF shall use ***the following rules*** when interpreting the TSF data from another trusted IT product:

•***The*** [JCVM21] ***specification;***
•***Reference export files;***
•***The ISO [7816-6] rules;***
•***The EMV specification***

FPT_FLS.1/JCS FAILURE WITH PRESERVATION OF SECURE STATE

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

FPR_UNO.1 UNOBSERVABILITY

FPR_UNO.1.1 The TSF shall ensure that [assignment: **subjects S.PACKAGE**] are unable to observe the operation [assignment: **all operations**] on [assignment: **all objects that contain secret data including keys and PIN**] by [assignment: **other subjects S.PACKAGE**].

FPT_TST.1 TSF TESTING

FPT_TST.1.1	The TSF shall run a suite of self-tests during initial start-up (at each power on) to demonstrate the correct operation of the TSF.
FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of TSF data.
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

4.1.1.4. AID Management

FMT_MTD.1/JCRE MANAGEMENT OF TSF DATA

*(See **FMT_SMR.1.1/JCRE** for the roles)*

FMT_MTD.1.1/JCRE	The TSF shall restrict the ability to **modify** the **list of registered applets' *AID*** to **the Java Card RE**.

FMT_MTD.3 SECURE TSF DATA

FMT_MTD.3.1	The TSF shall ensure that only secure values are accepted for TSF data.

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: ***the AID and version number of each package, the AID of*** each registered applet, and whether a registered applet is currently selected ***for execution ([JCVM21], §6.5).***

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.

FIA_USB.1 USER-SUBJECT BINDING

FIA_USB.1.1	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: **Context attribute of S.PACKAGE**].
FIA_USB.1.2	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: **The AID of a package being loaded shall not exist in GlobalPlatform Registry** **GlobalPlatform Registry: a container of information related to Card Content management**].
FIA_USB.1.3	The TSF shall enforce the following rules governing changes to the user

security attributes associated with subjects acting on the behalf of users: [assignment: **none**].

4.1.2. *InstG* Security functional requirements

This group bulks the SFRs related to the installation of the *applet*s, which addresses security aspects outside the runtime. The idea here is that installation of *applet*s is a critical phase, which lies partially out of the boundaries of the *firewall*, and therefore has to be deserved specific treatment. In the Common Criteria model, loading a *package* or installing an *applet* was considered as being an 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).

See also *FIA_ATD.1, **FIA_USB.1**, FMT_MTD.1, FMT_SMR.1* for various information about *applet* installation.

FDP_ITC.2/INSTALLER IMPORT OF USER DATA WITH SECURITY ATTRIBUTES

FDP_ITC.2.1/Installer — ***The TSF shall enforce the PACKAGE LOADING information flow control SFP*** when importing user data, controlled under the SFP, from outside of the TOE.

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

FDP_ITC.2.3/Installer — ***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/Installer — ***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/Installer — ***The TSF shall enforce the following rule when importing user data*** controlled under the SFP from outside the TOE:

A *package* may depend on (import or use data from) other *package*s already installed. This dependency is explicitly stated in the loaded *package* in the form of a list of *package AIDs*. The 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 ([JCVM21],§4.5.2).

FMT_SMR.1/INSTALLER SECURITY ROLES

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

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

FPT_FLS.1/INSTALLER FAILURE WITH PRESERVATION OF SECURE STATE

FPT_FLS.1.1/Installer — The TSF shall preserve a secure state when the following types of failures occur: the installer fails to load/install a package/applet as described in [JCRE21] §10.1.4.

FPT_RCV.3/INSTALLER AUTOMATED RECOVERY WITHOUT UNDUE LOSS

FPT_RCV.3.1/Installer When automated recovery from [assignment: none] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2/Installer For [assignment: **package loading, installation failure**], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

Other events such as the unexpected tearing of the card, power loss, and so on. are partially handled by the underlying hardware platform (see the SCPG group) and, from the TOE's side, by events "that clear transient objects" and transactional features. See ***FPT_FLS.1.1/JCS***, ***FDP_RIP.1.1/TRANSIENT***, ***FDP_RIP.1.1/ABORT*** and *FDP_ROL.1*.

FPT_RCV.3.3/Installer The functions provided by the TSF to recover from [**refinement**: **Package loading failure and Installation failure**] shall ensure that the secure initial state is restored without exceeding [assignment: **0%**] for loss of TSF data or objects under the control of the TSF.

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

FRU_RSA.1/INSTALLER MAXIMUM QUOTAS

FRU_RSA.1.1/Installer The TSF shall enforce maximum quotas of the following resources: ***imported packages*** and **declared classes, methods and fields** that **packages** can use **simultaneously.**

4.1.3. *ADELG Security functional requirements*

This group bulks the SFRs related to the deletion of *applet*s and/or *packages* , enforcing the *applet deletion manager* **(ADEL) policy** on security aspects outside the runtime. The idea here is that deletion is a critical phase and therefore requires specific treatment. This policy is better thought as a frame to be filled by ST implementers.

4.1.3.1. Applet Deletion Manager Policy

FDP_ACC.2/ADEL COMPLETE ACCESS CONTROL

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

Subjects (prefixed with an "*S*") and objects (prefixed with an "*O*") covered by this policy are:

S.ADEL The *applet deletion manager*. It may be an *applet* ([JCRE22], §11), but its role asks anyway for a specific treatment from the security viewpoint.
This subject is unique.

In Java Card version 2.2.1, S.ADEL calls the "uninstall" method of the applet instance to be deleted, if implemented by the applet, to inform it of the deletion request. The order in which these calls and the dependencies

checks are performed are out of the scope of this protection profile: O.CODE_PKG
The code of a *package* , including all linking information. On the Java Card platform, a package is the installation unit.

O.APPLET Any installed *applet* , its code and data.

O.JAVAOBJECT Java class instance or array.

Operations (prefixed with "*OP*") of this policy are described in the following table.

Operation	**Description**
OP.DELETE_APPLET(O.APPLET,…)	Delete an installed *applet* and its objects, either logically or physically.
OP.DELETE_PCKG(O.CODE_PKG,…)	Delete a *package* , either logically or physically
OP.DELETE_PCKG_APPLET(O.CODE_PKG, …)	Delete a *package* and its installed *applet*s, either logically or physically.

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: ***(1) the security attributes of the covered subjects and objects, (2)*** the list of AIDs of the applet instances registered on the card, (3) the attribute ResidentPackages, which journals the list of AIDs of the packages already loaded on the card, and (4) the attribute ActiveApplets, ***which is a list of the active applets' AIDs.***

The following table presents some of the security attributes associated to the subjects/objects under control of the policy. However, they are mostly implementation independent.

Subject/Object	**Attributes**
O.CODE_PKG	*package*'s *AID* , dependent packages' *AID*s, Static References
O.APPLET	Selection state
O.JAVAOBJECT	Owner, Remote

The package's *AID* identifies the package defined in the CAP file.

When an export file is used during preparation of a CAP file, the version numbers and *AIDs* indicated in the export file are recorded in the CAP files ([JCVM21], §4.5.2): the dependent packages AIDs attribute allows the retrieval of those identifications.

Static fields of a package may contain references to objects. The Static References attribute records those references.

An applet instance can be in two different selection states: selected or deselected. If the applet is selected (in some logical channel), then in turn it could either be *currently selected* or just *active*. At any time there could be

more than one active applet instances over each I/O interface, but only one currently selected (the maximum number of active applet instances depends on the Java Card platform version). This latter is the one that is processing the current command ([JCRE22], §4).

The Owner of an object is either the applet instance that created the object or the package (library) where it has been defined (these latter objects can only be arrays that initialize static fields of the package).

An object is said to be remote if it is an instance of a class that directly or indirectly implements the interface **java.rmi.Remote**.

Finally, there are needed security attributes that are not attached to any object or subject of the TSP: (1) the ResidentPackages Versions (or Resident Image,[JCVM21],§4.5) and *AID*s. They describe the packages that are already on the card, (2) the list of registered applet instances and (3) the ActiveApplets security attribute. They are all attributes internal to the Java Card VM, that is, not attached to any specific object or subject of the SPM.
These attributes are TSF data that play a role in the SPM.

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 by the ADEL SFP:***

The subject of this policy is *S.ADEL*.

Some basic common specifications are required in order to allow Java Card *applet*s and *package*s to be deleted without knowing the implementation details of a particular deletion manager. In particular, this policy introduces a notion of **reachability**, which provides a general means to describe objects that are referenced from a certain *applet* instance or *package*.

In the context of this policy, an object O is reachable if and only if either: (1) the owner of O is a registered *applet* instance A (O is reachable from A), (2) a static field of a loaded *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), and (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 ([JCRE22], §11.3.4.1, **Applet Instance Deletion**). The *S.ADEL* may perform *OP.DELETE_APPLET* upon an *O.APPLET* only if, (1) *S.ADEL* is currently selected, (2) *O.APPLET* is deselected 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 ([JCRE22], §8.5) *O.JAVAOBJECT* is remote reachable.

From Java Card platform, version 2.2.1, condition (2) becomes:

(2') There is no instance in the context of O.APPLET that is active in any logical channel.

R.JAVA.15 ([JCRE22],§11.3.4.1, **Multiple Applet Instance Deletion**). The *S.ADEL* may perform *OP.DELETE_APPLET* upon several *O.APPLET* only if, (1) *S.ADEL* is currently selected, (2) every

O.APPLET being deleted is deselected 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 ([JCRE22], §8.5) *O.JAVAOBJECT* is remote reachable.

From Java Card platform, version 2.2.1, condition (2) becomes:

(2') There is no instance of any of the O.APPLETs being deleted that is active in any logical channel.

R.JAVA.16 ([JCRE22], §11.3.4.2, **Applet/Library Package Deletion***).* The *S.ADEL* may perform *OP.DELETE_PCKG* upon an *O.CODE_PCKG* only if, (1) *S.ADEL* is currently selected, (2) no reachable *O.JAVAOBJECT*, from a package distinct from *O.CODE_PCKG* that is an instance of a class that belongs to *O.CODE_PCKG* exists on the card and (3) there is no package loaded on the card that depends on *O.CODE_PCKG.*

R.JAVA.17 ([JCRE22], §11.3.4.3, Applet **Package and Contained Instances Deletion***).* The *S.ADEL* may perform *OP.DELETE_PCKG_APPLET* upon an *O.CODE_PCKG* only if, (1) *S.ADEL* is currently selected, (2) no reachable *O.JAVAOBJECT*, from a package distinct from *O.CODE_PCKG,* which is an instance of a class that belongs to *O.CODE_PCKG* exists on the card, (3) there is no package loaded on the card that depends on *O.CODE_PCKG* and (4) for every *O.APPLET* of those being deleted it holds that: (i) *O.APPLET* is deselected 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 ([JCRE22],§8.5) *O.JAVAOBJECT* is remote reachable.

From Java Card platform, version 2.2.1, condition (4i) becomes:

(4i') There is no instance in the context of O.APPLET that is active in any logical channel.

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

FDP_ACF.1.4/ADEL The TSF shall explicitly deny access of ***any subject but the*** *S.ADEL* ***to*** *O.CODE_PKG* ***or*** *O.APPLET* ***for the purpose of deleting it from the card.***

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 ActiveApplets security attribute to the Java Card RE (S.JCRE).

FMT_MSA.3/ADEL STATIC ATTRIBUTE INITIALIZATION

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

FMT_MSA.3.2/ADEL The TSF shall allow the following role(s) to specify alternative initial values to override the default values when an object or information is created:

none.

FMT_SMR.1/ADEL SECURITY ROLES	
FMT_SMR.1.1/ADEL	The TSF shall maintain the roles: *the* ***applet deletion manager***.
FMT_SMR.1.2/ADEL	The TSF shall be able to associate users with roles.

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 ActiveApplets security attribute***.

4.1.3.2. Additional Deletion Requirements

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 de-allocation 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.

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* [JCRE22], §11.3.4.

4.1.4. *RMIG* Security functional requirements

This group is mainly devoted to specifying the policies that control the access to remote objects and the flow of information that takes place when the Java Card RMI service is used. There are specific control rules concerning the access to remote objects. The rules relate mainly to the lifetime of their corresponding remote references. Information concerning remote object references can be sent out of the card only if the corresponding remote object has been designated as exportable. Array parameters of remote method invocations are required to be allocated on the card as global arrays, the storage of references to those arrays must then be restricted as well.

4.1.4.1. Java Card RMI Policy

The *Java Card RMI* policy embodies both an access control and an information flow control policy.

FDP_ACC.2/JCRMI COMPLETE ACCESS CONTROL	
FDP_ACC.2.1/JCRMI	The TSF shall enforce the ***JCRMI access control SFP*** on *S.CAD, S.JCRE, O.APPLET, O.REMOTE_OBJ, O.REMOTE_MTHD, O.ROR, O.RMI_SERVICE* and all operations among subjects and objects covered by the SFP.

Subjects (prefixed with an "*S*") and objects (prefixed with an "*O*") covered by this policy are:

S.CAD	The *CAD*. In the scope of this policy it represents the actor that requests, by issuing commands to the card, for Java Card RMI services.
S.JCRE	The *Java Card RE* is responsible on behalf of the card issuer of the bytecode execution and runtime environment functionalities. In the context of this security policy, the *Java Card RE* is in charge of the execution of the commands provided to (1) obtain the initial remote reference of an applet instance and (2) perform Remote Method Invocation.
O.APPLET	Any installed *applet* , its code and data.
O.REMOTE_OBJ	A remote object is an instance of a class that implements one (or more) remote interfaces. A remote interface is one that extends, directly or indirectly, the interface **java.rmi.Remote** ([JCAPI22]).
[refinement:	
O.ROR	A remote object reference. It provides information concerning: (i) the identification of a remote object and (ii) the Implementation class of the object or the interfaces implemented by the class of the object. This is the object's information to which the CAD can access.
]	
O.REMOTE_MTHD	A method of a remote interface.
O.RMI_SERVICE	These are instances of the class **javacardx.rmi.RMIService**. They are the objects that actually process the Java Card RMI services.

Operations (prefixed with "*OP*") of this policy are described in the following table.

Operation	Description
OP.GET_ROR(O.APPLET,…)	Retrieves the initial remote object reference of a Java Card RMI based applet. This reference is the seed which the CAD client application needs to begin remote method invocations
OP.INVOKE(O.RMI_SERVICE,…)	Requests a remote method invocation on the remote object.

FDP_ACC.2.2/JCRMI 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/JCRMI SECURITY ATTRIBUTE BASED ACCESS CONTROL

FDP_ACF.1.1/JCRMI The TSF shall enforce the ***JCRMI access control SFP*** to objects based on the following: ***(1) the security attributes of the covered subjects and objects, (2)*** the list of AIDs of the applet instances registered on the card

and (3) the ***attribute ActiveApplets, which is a list of the active applets' AIDs.***

The following table presents the security attributes associated to the objects under control of the policy.

Object	**Attributes**
O.APPLET	*Package* 's *AID* or *none*
O.REMOTE_OBJ	Owner, class, Identifier, Exported
O.REMOTE_MTHD	Identifier
O.RMI_SERVICE	Owner, Returned References
[refinement:	
O.ROR	Valid
]	

The package's *AID* identifies the package defined in the CAP file.

An applet instance can be in two different selection states: selected or deselected. If the applet is selected (in some logical channel), then in turn it could either be *currently selected* or just *active*. At any time there could be more than one active applet instances over each I/O interface, but only one currently selected. This latter is the one that is processing the current command ([JCRE22], §4).

The **owner** of a remote object is the applet instance that created the object. The **class** attribute identifies the implementation class of the remote object. The **remote object Identifier** is a number that uniquely identifies a remote object in the card. The attribute **Exported** indicates whether the remote object is exportable or not.

A **remote method Identifier** is a number that uniquely identifies a remote method within a certain remote class.

The **owner** of an *O.RMI_SERVICE* is the applet instance that created the object. The attribute **Returned References** lists the remote object references that have been sent to the CAD during the applet selection session. This attribute is implementation dependent.

[refinement: The **validity** of an *O.ROR* is defined in [JCRE22], §8.5.**]**

Finally, there are some security attributes that are not attached to any object or subject of the TSP: (1) the list of registered applet instances and (2) the ActiveApplets security attribute. They are all attributes internal to the Java Card VM that is, not attached to any specific object or subject of the SPM. These attributes are TSF data that play a role in the SPM.

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

R.JAVA.18 The *S.CAD* may perform *OP.GET_ROR* upon an *O.APPLET* only if *O.APPLET* is the *currently selected applet*, and there exists an *O.RMI_SERVICE* with a registered initial reference to an *O.REMOTE_OBJ* that is owned by *O.APPLET*.

R.JAVA.19 The *S.JCRE* may perform *OP.INVOKE* upon *O.RMI_SERVICE*,

O.ROR and *O.REMOTE_MTHD,* only if, *O.ROR* is *valid* and belongs to the *Returned References* of *O.RMI_SERVICE*, and the *Identifier* of *O.REMOTE_MTHD* matches one of the remote methods in the class, indicated by the security attribute *class*, of the *O.REMOTE_OBJECT* to which *O.ROR* makes reference.

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

FDP_ACF.1.4/JCRMI The TSF shall explicitly deny access of ***any subject but*** *S.JCRE* to *O.REMOTE_OBJ* and *O.REMOTE_MTHD* for the purpose of performing a remote method invocation.

FDP_IFC.1/JCRMI SUBSET INFORMATION FLOW CONTROL

FDP_IFC.1.1/JCRMI The TSF shall enforce the ***JCRMI information flow control SFP*** on the following subjects, information and operations.

Subjects12 (prefixed with an "*S*") and information (prefixed with an "*I*") covered by this policy are:

Subject/Information	Description
S.JCRE	As in the Access control policy
S.CAD	As in the Access control policy
I.RORD	Remote object reference descriptors

A remote object reference descriptor provides information concerning: (I) the identification of the remote object and (ii) the implementation class of the object or the interfaces implemented by the class of the object. The descriptor is the only object's information to which the CAD can access.

There is a unique operation in this policy:

Operation	Description
OP.RET_RORD(S.JCRE,S.CAD,I.RORD)	Send a remote object reference descriptor to the CAD.

A remote object reference descriptor is sent from the card to the CAD either as the result of a successful applet selection command ([JCRE22], §8.4.1), and in this case it describes, if any, the initial remote object reference of the selected applet; or as the result of a remote method invocation ([JCRE22],§8.3.5.1) .

FDP_IFF.1/JCRMI SIMPLE SECURITY ATTRIBUTES

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

The following table summarizes which security attribute is attributed to which subject/information.

Subject/Information	Attributes
S.JCRE	None
S.CAD	None
I.RORD	ExportedInfo (Boolean value)

The ExportedInfo attribute of an I.RORD indicates whether the *O.REMOTE_OBJ* which *I.RORD* identifies is exported or not (as indicated by the security attribute Exported of the *O.REMOTE_OBJ*).

FDP_IFF.1.2/JCRMI The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rule holds:

An operation *OP.RET_RORD(S.JCRE*, *S.CAD*, *I.RORD)* is permitted only if the attribute ExportedInfo *I.RORD* has the value **"true"** ([JCRE22], §8.5).

FDP_IFF.1.3/JCRMI The TSF shall enforce [assignment: **none***]*.

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

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

FMT_MSA.1/JCRMI MANAGEMENT OF SECURITY ATTRIBUTES

FMT_MSA.1.1/JCRMI *The TSF shall enforce the FIREWALL access control SFP and the JCVM information flow control SFP to restrict the ability to modify the ActiveApplets security attribute to the Java Card RE (S.JCRE).*

FMT_MSA.1/EXPORT MANAGEMENT OF SECURITY ATTRIBUTES

FMT_MSA.1.1/EXPORT The TSF shall enforce the JCRMI access control SFP and the JCRMI information flow control SFP to restrict the ability to modify the security attribute Exported of an O.REMOTE_OBJ to its owner.

FMT_MSA.1/REM_REFS MANAGEMENT OF SECURITY ATTRIBUTES

FMT_MSA.1.1/REM_REFS The TSF shall enforce the ***JCRMI access control SFP and the JCRMI information flow control SFP*** to restrict the ability to **modify** the security attribute **Returned References of an O.RMI_SERVICE** to **its owner.**

FMT_MSA.3/JCRMI STATIC ATTRIBUTE INITIALIZATION

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

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

FMT_REV.1/JCRMI REVOCATION	
FMT_REV.1.1/JCRMI	The TSF shall restrict the ability to revoke the Returned References security attribute of an O.RMI_SERVICE to the Java Card RE [assignment: **none**].
FMT_REV.1.2/JCRMI	The TSF shall enforce the rules ***that determine the lifetime of remote object references***.

FMT_SMR.1/JCRMI SECURITY ROLES	
FMT_SMR.1.1/JCRMI	The TSF shall maintain the roles: applet.
FMT_SMR.1.2/JCRMI	The TSF shall be able to associate users with roles.

FMT_SMF.1/JCRMI SPECIFICATION OF MANAGEMENT FUNCTIONS	
FMT_SMF.1.1/JCRMI	The TSF shall be capable of performing the following management functions: - ***Modify the security attribute Exported of an O.REMOTE_OBJ.*** - ***Modify the security attribute Returned References of an O.RMI_SERVICE.***

4.1.5. *ODELG* Security functional requirements

The following requirements are concerned with the secure deletion of information provoked by the object deletion mechanism. This mechanism is triggered by the applet who 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 ***de-allocation 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().**

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 type of failure occurs: ***the object deletion functions fail to delete all the unreferenced objects owned by the applet that requested the execution of the method***

4.1.6. *CardG* Security functional requirements

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.

FCO_NRO.2.2/CM 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 the recipient given [assignment: immediate verification of the origin].

FIA_UID.1/CM TIMING OF IDENTIFICATION

FIA_UID.1.1/CM The TSF shall allow [assignment: **none**] on behalf of the user to be performed before the user is identified.

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

FDP_IFC.2/CM COMPLETE INFORMATION FLOW CONTROL

FDP_IFC.2.1/CM The TSF shall enforce the ***PACKAGE LOADING information flow control SFP*** on ***S.CRD, S.BCV, S.SPY*** and all operations that cause that information to flow to and from subjects covered by the SFP.

Subjects (prefixed with an "*S*") covered by this policy are those involved in the reception of an application package by the card through a potentially unsafe communication channel:

Subject	Description
S.BCV	The subject representing who is in charge of the bytecode verification of the packages (also known as the verification authority).
S.CRD	The on-card entity in charge of package downloading.
S.SPY	Any other subject that may potentially intercept, modify, or permute the messages exchanged between the former two subjects.

The operations (prefixed with "*OP*") that make information to flow between the subjects are those enabling to send a message through and to receive a message from the communication channel linking the card to the outside world. It is assumed that an attacker is able to read any message sent through the channel as clear text. Moreover, the attacker may capture any message sent through the communication channel and send its own messages to the other subjects.

Operation	Description
OP.SEND(M)	A subject sends a message M through the communication channel.
OP.RECEIVE(M)	A subject receives a message M from the communication channel.

The information (prefixed with an "**I**") controlled by the typing policy is the *APDUs* exchanged by the subjects through the communication channel linking the card and the CAD. Each of those messages contain part of an application package that is required to be loaded on the card, as well as any control information used by the subjects (either S.BCV or S.SPY) in the communication protocol.

Information	Description
I.APDU	Any *APDU* sent to or from the card through the communication channel.

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

FDP_IFF.1/CM SIMPLE SECURITY ATTRIBUTES

FDP_IFF.1.1/CM The TSF shall enforce the PACKAGE LOADING information flow control SFP based on the following types of subject and information security attributes:
[assignment:
The following table describes which security attributes are attached to which subject/information.

Subject/Information	Attributes
S.BCV	**none**
S.CRD	**SecureChannelEstabilshed, SecureLevel, SignatureVerified**
S.SPY	**none**
I.APDU	**SecureLevel**

The following table describes the possible values for each security attribute.

Name	Description
SecureChannelEstabilshed	**Boolean value: true or false**
SecureLevel	**None, MAC, ENC or [MAC and ENC]**
SignatureVerified	**Boolean value: true or false**

].

FDP_IFF.1.2/CM The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold:
[assignment:
An information flow is allowed only if all following rules hold:
a) The attribute SecureChannelEstabilshed of S.CRD is true.
b) The attribute SecureLevel of I.APDU meets the attribute SecureLevel of S.CRD.
c) The attribute SignatureVerified of S.CRD is true.
].

FDP_IFF.1.3/CM The TSF shall enforce the [assignment:**none**].

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

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

FDP_UIT.1/CM DATA EXCHANGE INTEGRITY

These requirements apply to those configurations where bytecode verification is not considered as being part of the TOE. If this is the case, then the bytecode verifier can be seen as an external IT product, and packages to be loaded on the card are user data in transit from that external product to the *Java Card System*.

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

FMT_MSA.1/CM MANAGEMENT OF SECURITY ATTRIBUTES

FMT_MSA.1.1/CM	*The TSF shall enforce the PACKAGE LOADING information flow control SFP* to restrict the ability to [selection: **query and modify**] the security attributes [assignment: **SecureChannelEstabilshed, SecureLevel, and SignatureVerified**] to [assignment: **Card Issuer , Application Provider and Controlling Authority**].

FMT_MSA.3/CM STATIC ATTRIBUTE INITIALIZATION

FMT_MSA.3.1/CM	The TSF shall enforce the PACKAGE LOADING information flow control SFP to provide restrictive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/CM	The TSF shall allow the [assignment: **none**] to specify alternative initial values to override the default values when an object or information is created.

FMT_SMR.1/CM SECURITY ROLES

FMT_SMR.1.1/CM	The TSF shall maintain the roles: [assignment: **Card Issuer ,Application Provider, Controlling Authority**].
FMT_SMR.1.2/CM	The TSF shall be able to associate users with roles.

FMT_SMF.1/CM SPECIFICATION OF MANAGEMENT FUNCTIONS

FMT_SMF.1.1/CM	The TSF shall be capable of performing the following management functions: [assignment: – Modify the security attribute **SecureChannelEstabilshed** of S.CRD. – Modify the security attribute **SecureLevel** of S.CRD. – Modify the security attribute **SignatureVerified** of S.CRD.].

FTP_ITC.1/CM INTER-TSF TRUSTED CHANNEL

FTP_ITC.1.1/CM	The TSF shall provide a communication channel between itself and another IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2/CM	The TSF shall permit the CAD placed in a secured environment to initiate communication through the trusted channel.
FTP_ITC.1.3/CM	The TSF shall initiate communication through the trusted channel for installing a new application package on the card.

4.1.7. *SCPG* Security functional requirements

This group contains the security requirements for the smart card platform, that is, operating system and chip that the Java Card System is implemented upon. The requirements are expressed in terms of security functional requirements from [CC2].

FPT_FLS.1/SCP FAILURE WITH PRESERVATION OF SECURE STATE

FPT_FLS.1.1/SCP The TSF shall preserve a secure state when the following types of failures occur: [assignment:
- **lack of EEPROM**
- **random generator and cryptographic co-processor failure**
- **RAM read/write failure**
- **Card Tearing**].

These components shall be used to specify the list of SCP capabilities supporting the Java Card System/CM that will still be operational at the occurrence of the mentioned failures (EEPROM worn out, lack of EEPROM, random generator failure).

FRU_FLT.2/SCP LIMITED FAULT TOLERANCE

FRU_FLT.2.1/SCP The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: [assignment: **exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1/SCP)**].

FPT_PHP.3/SCP RESISTANCE TO PHYSICAL ATTACK

FPT_PHP.3.1/SCP The TSF shall resist [assignment: **all physical attacks**] to the [assignment: **IC**] by responding automatically such that the SFRs are always enforced.

FPT_RCV.3/SCP AUTOMATED RECOVERY WITHOUT UNDUE LOSS

FPT_RCV.3.1/SCP	When automated recovery from [assignment: **none**] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.
FPT_RCV.3.2/SCP	For [assignment: **all atomic operations on EEPROM**], the TSF shall ensure the return of the TOE to a secure state using automated procedures.
FPT_RCV.3.3/SCP	The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding [assignment: ***0%]*** for loss of TSF data or objects under the

control of the TSF.

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

FPT_RCV.4/SCP FUNCTION RECOVERY

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

FDP_ITT.1/SCP BASIC INTERNAL TRANSFER PROTECTION

FDP_ITT.1.1/SCP	The TSF shall enforce the [assignment: **Data Processing Policy**] to prevent the [selection: **disclosure**] of user data when it is transmitted between physically-separated parts of the TOE. **Refinement:** The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as physically-separated parts of the TOE.

FPT_ITT.1/SCP BASIC INTERNAL TSF DATA TRANSFER PROTECTION

FPT_ITT.1.1/SCP	The TSF shall protect TSF data from [selection: **disclosure**] when it is transmitted between separate parts of the TOE. **Refinement:** The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE. This requirement is equivalent to FDP_ITT.1 above but refers to TSF data instead of User Data. Therefore, it should be understood as to refer to the same Data Processing Policy defined under FDP_IFC.1 below.

FDP_IFC.1/SCP SUBSET INFORMATION FLOW CONTROL

FDP_IFC.1.1/SCP	The TSF shall enforce the [assignment: **Data Processing Policy**] on [assignment: **all confidential data**]. **Refinement:** This applies when data is processed or transferred by the TOE or by the underlying platform. Data Processing Policy: User Data and TSF data shall not be accessible from the TOE except when the Smartcard Embedded Software decides to communicate the User Data via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the OS.

FCS_RND.1/SCP QUALITY METRIC FOR RANDOM NUMBERS

FCS_RND.1.1/SCP	The TSF shall provide a mechanism to generate random numbers that

meet the [assignment: **AIS20 version 1, Functional Classes and Evaluation Methodology for Deterministic Random Number Generators, 2 December 1999, Class K3 Strength of Function High requirements**].

4.1.8. *CMGRG Security functional requirements*

This group contains the security requirements for the card manager. They are all expressed in terms of security functional requirements from [CC2].

The security requirements below helps defining a policy for controlling access to card content management operations and for expressing card issuer security concerns.

FDP_ACC.1/CMGR SUBSET ACCESS CONTROL

FDP_ACC.1.1/CMGR

The TSF shall enforce the ***CARD CONTENT MANAGEMENT access control SFP*** on [assignment: **subjects , objects and operations described hereafter.**

Subjects (prefixed with an "S") and objects (prefixed with an "O") covered by this policy are:

Subject/Object	Description
S.ISD	The Issuer Security Domain.
S.SD	The Application Provider Security Domain.
S.CONT_AUTH	Controlling Authority.
S.OPEN	GlobalPlatform environment.
O.CONTENT	Any content managed by Card Content Management Functions.
O.GP_REG	GlobalPlatform registry.

Operations (prefixed with "OP") of this policy are described in the following table.

Operation	Description
OP.VERIFY(O.CONTENT)	Verify Card Content.
OP.CCMF(O.GP_REG, O.CONTENT)	Perform Card Content Management Functions .

].

FDP_ACF.1/CMGR SECURITY ATTRIBUTE BASED ACCESS CONTROL

FDP_ACF.1.1/CMGR

The TSF shall enforce the ***CARD CONTENT MANAGEMENT*** access control SFP to objects based on the following: [assignment: **subjects, objects and their security attributes, described hereafter.**

The following table describes which security attributes are attached to which subject/object..

Subject/Object	Attributes

S.ISD	**None**
S.SD	**None**
S.CONT_AUTH	**None**
S.OPEN	**None**
O.CONTENT	**Verified, Authorized**
O.GP_REG	**None**

The following table describes the security attribute.

Name	Description
Verified	**Indicates if O.CONTENT has been verified by S.CONT_AUTH.**
Authorized	**Indicates if delegated CCMFs has been authorized by S.ISD.**

The following table describes the possible values for each security attribute.

Name	Values
Verified, Authorized	**Boolean value: true and false**

].

FDP_ACF.1.2/CMGR The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed [assignment: **by the CARD CONTENT MANAGEMENT SFP:**

R.GP.1 S.ISD and S.SD shall require only the minimum security requirements for GP commands as defined by GPCS.

R.GP.2 S.CONT_AUTH shall perform OP.VERIFY upon O.CONTENT for each OP.CCMF that is related to card content loading upon O.GP_REG.

R.GP.3 S.ISD and S.SD shall be allowed to request S.OPEN to perform OP.CCMF that is related to card content loading upon O.GP_REG only if the security attribute Verified of O.CONTENT is true.

R.GP.4 S.ISD shall always be allowed to request S.OPEN to perform any OP.CCMF (except card content loading) upon O.GP_REG for any O.CONTENT.

R.GP.5 S.ISD shall preauthorize every CCMF (except delete of S.SD's own O.CONTENT) requested by S.SD.

R.GP.6 S.SD shall be allowed to request S.OPEN to perform OP.CCMF (except delete of S.SD's own O.CONTENT) upon O.GP_REG only if the security attribute Authorized of O.CONTENT is true.

R.GP.7 S.ISD shall confirm for each delegated CCMF that has taken place.

R.GP.8 S.SD shall be allowed to request S.OPEN to preform OP.CCMF (extradition) upon O.GP_REG for its own O.CONTENT.

].

FDP_ACF.1.3/CMGR	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: **none**].
FDP_ACF.1.4/CMGR	The TSF shall explicitly deny access of subjects to objects based on the [assignment: **the rules described hereafter,** **R.GP.9 S.ISD and S.SD shall deny requesting S.OPEN to perform OP.CCMF that is related to card content loading upon O.GP_REG if the security attribute Verified of O.CONTENT is false.** **R.GP.10 S.SD shall deny requesting S.OPEN to perform OP.CCMF (except delete of S.SD's own O.CONTENT) upon O.GP_REG if the security attribute Authorized of O.CONTENT is false.**].

FMT_MSA.1/CMGR MANAGEMENT OF SECURITY ATTRIBUTES

FMT_MSA.1.1/CMGR	[**Editorially Refined**]The TSF shall enforce the ***CARD CONTENT MANAGEMENT access control SFP*** to restrict the ability to [selection: **modify**] **the security attribute Verified to Controlling Authority and the security attribute Authorized to Card Issuer**.

FMT_MSA.3/CMGR STATIC ATTRIBUTE INITIALIZATION

FMT_MSA.3.1/CMGR	The TSF shall enforce the **CARD CONTENT MANAGEMENT access control SFP** to provide **restrictive** default values for security attributes that are used to enforce the *SFP*.
FMT_MSA.3.2/CMGR	The TSF shall allow the [assignment: **none**] to specify alternative initial values to override the default values when an object or information is created.

FMT_SMR.1/CMGR SECURITY ROLES

FMT_SMR.1.1/CMGR	The TSF shall maintain the roles: [assignment: **Card Issuer, Application Provider and User**].
FMT_SMR.1.2/CMGR	The TSF shall be able to associate users with roles.

FMT_SMF.1/CMGR SPECIFICATION OF MANAGEMENT FUNCTIONS

FMT_SMF.1.1/CMGR	The TSF shall be capable of performing the following management functions: [assignment: **manage card content in GlobalPlatform registry.**].

FIA_UID.1/CMGR TIMING OF IDENTIFICATION

FIA_UID.1.1/CMGR	The TSF shall allow [assignment: **none**] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2/CMGR	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

4.1.9. General GP Security functional requirements

FDP_DAU.1/GP BASIC DATA AUTHENTICATION

FDP_DAU.1.1/GP The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [assignment: **every O.JAVAOBJECT**].

FDP_DAU.1.2/GP The TSF shall provide [assignment: **S.JCRE**] with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence

FIA_AFL.1/GP BASIC AUTHENTICATION FAILURE HANDLING

FIA_AFL.1.1/GP The TSF shall detect when [selection: *[assignment:* **10***]*] unsuccessful authentication attempts occur related to [assignment: **secure channel establishment**].

FIA_AFL.1.2/GP When the defined number of unsuccessful authentication attempts has been [selection: **surpassed**], the TSF shall [assignment: **block the TOE**].

FIA_ATD.1/GP USER ATTRIBUTE DEFINITION

FIA_ATD.1.1/GP The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: **Privilege**].

.

The following table describes the security attribute.

Name	Description
Privilege	**Application Privilege defined in [GP211]**

FIA_UAU.1/GP TIMING OF AUTHENTICATION

FIA_UAU.1.1/GP The TSF shall allow [assignment: **GET DATA and INITIALIZE UPDATE defined in GP**] on behalf of the user to be performed before the user is authenticated.

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

FIA_UAU.4/GP SINGLE-USE AUTHENTICATION MECHANISMS

FIA_UAU.4.1/GP The TSF shall prevent reuse of authentication data related to [**Secure Channel Establishment.**]

FIA_USB.1/GP USER-SUBJECT BINDING

FIA_USB.1.1/GP The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: **PRIVILEGE**].

FIA_USB.1.2/GP The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment:

none].

FIA_USB.1.3/GP The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: **none**].

FMT_MOF.1/GP MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOR

FMT_MOF.1.1/GP The TSF shall restrict the ability to [selection: **enable and modify the behaviour of**] the functions [assignment: **Card Content Management Functions (except delete of Application Provider's own application)**] to [assignment: **Card Issuer.**]

FMT_MSA.2/GP SECURE SECURITY ATTRIBUTES

FMT_MSA.2.1/GP The TSF shall ensure that only secure values are accepted for [assignment: **Verified and Authorized**].

FMT_MTD.1/GP MANAGEMENT OF TSF DATA

FMT_MTD.1.1/GP The TSF shall restrict the ability to [selection : **modify**] the [assignment : **GlobalPlatform Registry**] to [assignment : **Card Issuer and Application Provider**].

FMT_SMF.1/GP SPECIFICATION OF MANAGEMENT FUNCTION

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

- **enable and modify the behaviour of Card Content Management Functions.**

].

4.2. Security functional requirements rationale

Table 4. Security functional requirements (*CoreG*) - security objectives tracing

	O.ALARM	O.CIPHER	O.FIREWALL	O.KEY-MNGT	O.OPERATE	O.PIN-MNGT	O.RESOURCES	O.SID	O.TRANSACTION	O.SHRD_VAR_CONFID	O.SHRD_VAR_INTEG	O.REALLOCATION
FDP_ACC.2/FIREWALL			X		X	X						
FDP_ACF.1/FIREWALL			X		X	X						
FDP_IFC.1/JCVM			X							X	X	
FDP_IFF.1/JCVM			X								X	
FDP_RIP.1/OBJECTS						X						X

FMT_MSA.1/JCRE			X					X				
FMT_MSA.2/JCRE			X									
FMT_MSA.3/FIREWALL			X					X				
FMT_SMR.1/JCRE			X				X					
FMT_SMF.1/JCRE			X					X				
FCS_CKM.1/RSA		X		X								
FCS_CKM.2/TRIPLE-DES		X		X								
FCS_CKM.2/RSA		X		X								
FCS_CKM.3/TRIPLE-DES		X		X								
FCS_CKM.3/RSA		X		X								
FCS_CKM.4		X		X								
FCS_COP.1/TRIPLE-DES		X		X								
FCS_COP.1/RSA		X		X								
FCS_COP.1/DESMAC		X		X								
FCS_COP.1/RSA SIGNATURE		X		X								
FCS_COP.1/SHA-1		X		X								
FCS_COP.1/MD5		X		X								
FDP_RIP.1/APDU										X		
FDP_RIP.1/bArray										X		
FDP_RIP.1/TRANSIENT												X
FDP_RIP.1/ABORT									X			X
FDP_RIP.1/KEYS				X								X
FDP_ROL.1/FIREWALL					X	X	X		X			
FAU_ARP.1/JCS	X				X		X					
FDP_SDI.2				X		X						
FPT_TDC.1					X							
FPT_FLS.1/JCS	X				X		X					
FPR_UNO.1		X		X		X						
FPT_TST.1					X							
FMT_MTD.1/JCRE			X				X	X				
FMT_MTD.3			X				X	X				
FIA_ATD.1/AID					X			X				
FIA_UID.2/AID								X				
FIA_USB.1					X			X				

•FDP_ACC.2/FIREWALL - O.FIREWALL

◦O.FIREWALL is partially met by the FIREWALL access control policy .

•FDP_ACC.2/FIREWALL - O.OPERATE
◦The TOE is protected by FIREWALL access control policy with continued correct operation of its security functions.

•FDP_ACC.2/FIREWALL - O.PIN-MNGT
◦The firewall security functions protects the access to private and internal data of the objects that related to PIN management.

•FDP_ACF.1/FIREWALL - O.FIREWALL
◦O.FIREWALL is partially met by the FIREWALL access control policy .

•FDP_ACF.1/FIREWALL - O.OPERATE
◦The TOE is protected by FIREWALL access control policy with continued correct operation of its security functions.

•FDP_ACF.1/FIREWALL - O.PIN-MNGT
◦The firewall security functions protects the access to private and internal data of the objects that related to PIN management.

•FDP_IFC.1/JCVM - O.FIREWALL
◦O.FIREWALL is partially met by the Java Card VM information control policy.

•FDP_IFC.1/JCVM - O.SHRD_VAR_CONFID
◦The Java Card VM information control policy (FDP_IFC.1/JCVM) prevents an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.

•FDP_IFC.1/JCVM - O.SHRD_VAR_INTEG
◦The Java Card VM information control policy prevents an application from keeping a pointer to the input/output buffer of the card, or any other global array that is shared by all applications.

•FDP_IFF.1/JCVM - O.FIREWALL
◦O.FIREWALL is partially met by the Java Card VM information control policy.
•FDP_IFF.1/JCVM - O.SHRD_VAR_INTEG
◦The Java Card VM information control policy (FDP_IFF.1/JCVM) prevents an application from keeping a pointer to a shared buffer, which could be used to read its contents when the buffer is being used by another application.

•FDP_RIP.1/OBJECTS - O.PIN-MNGT
◦O.PIN-MNGT is partly met by FDP_RIP.1/OBJECTS that requires unavailability of any class instances and arrays upon the allocation of the resource.
•FDP_RIP.1/OBJECTS - O.REALLOCATION
◦O.REALLOCATION is satisfied by FDP_RIP.1/OBJECTS, which imposes that the contents of the re-allocated block shall always be cleared before delivering the block.

•FMT_MSA.1/JCRE - O.FIREWALL
◦FMT_MSA.1/JCRE indirectly contributes to meet O.FIREWALL.
•FMT_MSA.1/JCRE - O.SID
◦Subjects' identity is AID-based(applets, packages) and is met by FMT_MSA.1/JCRE.

•FMT_MSA.2/JCRE - O.FIREWALL
◦FMT_MSA.2/JCRE indirectly contribute to meet O.FIREWALL.

•*FMT_MSA.3/FIREWALL - O.FIREWALL*
◦FMT_MSA.3/FIREWALL indirectly contribute to meet O.FIREWALL.
•*FMT_MSA.3/FIREWALL - O.SID*

◦Subjects' identity is AID-based(applets, packages) and is met by FMT_MSA.1/FIREWALL.

•*FMT_SMR.1/JCRE – O.FIREWALL*
◦FMT_SMR.1/JCRE indirectly contribute to meet O.FIREWALL.
•*FMT_SMR.1/JCRE – O.RESOURCES*
◦Memory management is controlled by the TSF (FMT_SMR.1/JCRE).

•*FMT_SMF.1/JCRE – O.FIREWALL*
◦FMT_SMF.1/JCRE indirectly contribute to meet O.FIREWALL.
•*FMT_SMF.1/JCRE – O.SID*
◦Subjects' identity is AID-based(applets, packages) and is met by FMT_MSF.1/JCRE.

•*FCS_CKM.1/RSA - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.
•*FCS_CKM.1/RSA - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_CKM.1/TRIPLE-DES - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.
•*FCS_CKM.1/TRIPLE-DES - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_CKM.2/RSA - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.
•*FCS_CKM.2/RSA - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_CKM.2/TRIPLE-DES - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.
•*FCS_CKM.2/TRIPLE-DES - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_CKM.3/TRIPLE-DES - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.
•*FCS_CKM.3/ TRIPLE-DES - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_CKM.3/RSA - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.
•*FCS_CKM.3/RSA - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_CKM.4 - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.
•*FCS_CKM.4 - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_COP.1/TRIPLE-DES - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.

•*FCS_COP.1/TRIPLE-DES - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_COP.1/RSA - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.

•*FCS_COP.1/RSA - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_COP.1/DESMAC - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.

•*FCS_COP.1/DESMAC - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_COP.1/RSA SIGNATURE - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.

•*FCS_COP.1/RSA SIGNATURE - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_COP.1/SHA-1 - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.

•*FCS_COP.1/SHA-1 - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FCS_COP.1/MD5 - O.CIPHER*
◦O.CIPHER is partially covered by this SFR since it covers this iteration's cryptographic algorithm.

•*FCS_COP.1/MD5 - O.KEY-MNGT*
◦O.KEY-MNGT is partially covered by this SFR since it covers this iteration's cryptographic algorithm key management.

•*FDP_RIP.1/APDU – O.SHRD_VAR_CONFID*
◦The clearing requirement of APDU buffer required in Java Card API is met by FDP_RIP.1/APDU.

•*FDP_RIP.1/bArray - O.SHRD_VAR_CONFID*
◦The clearing requirement of bArray required in Java Card API is met by FDP_RIP.1/bArray .

•*FDP_RIP.1/ TRANSIENT - O.REALLOCATION*
◦O.REALLOCATION is satisfied by FDP_RIP.1/TRANSIENT, which imposes that the contents of the re-allocated block shall always be cleared before delivering the block.

•*FDP_RIP.1/ ABORT - O.TRANSACTION*
◦O.TRANSACTION is partially met by FDP_RIP.1/ABORT which ensures that no transaction data is accessible when a transaction is aborted.
•*FDP_RIP.1/ ABORT - O.REALLOCATION*
O.REALLOCATION is met by FDP_RIP.1/ABORT that requires unavailability of any reference to an object instance created during an aborted transaction upon deallocation of the resource.

•*FDP_RIP.1/KEYS - O.KEY-MNGT*
◦O.KEY-MNGT is met by FDP_RIP.1/KEYS that requires unavailability of the cryptographic buffer upon the deallocation of the resource.
•*FDP_RIP.1/ KEYS – O.REALLOCATION*
◦O.REALLOCATION is partly met by FDP_RIP.1/KEYS that requires unavailability of the cryptographic buffer upon the deallocation of the resource.

•*FDP_ROL.1/FIREWALL – O.OPERATE*
◦Applets' installation may be cleanly aborted (FDP_ROL.1/FIREWALL), TOE's security-critical parts are protected.
•*FDP_ROL.1/FIREWALL – O.PIN-MNGT*
◦O.PIN_MNGT is partly ensured by FDP_ROL.1/FIRWALL.
•*FDP_ROL.1/FIREWALL – O.RESOURCES*
◦ O.RESOURCES is met by FDP_ROL.1/FIREWALL, since failed installations are not to create memory leaks.
•*FDP_ROL.1/FIREWALL – O.TRANSACTION*
◦O.TRANSACTION is partially met by FDP_ROL.1/FIREWALL since it ensures that atomic operations are safe.

•*FAU_ARP.1/JCS – O.ALARM*
◦O.ALARM is partially met by FAU_ARP.1/JCS since an alarm is reported when a non fatal error occurs.
•*FAU_ARP.1/JCS – O.OPERATE*
◦O.OPERATE is met by FAU_ARP.1/JCS, since the TOE is able to detect and block various failures and security violations during usual working.
•*FAU_ARP.1/JCS – O.RESOURCES*
◦O.RESOURCES is met by FAU_ARP.1/JCS, since the TSFs detects stack/memory overflows during execution of application.

•*FDP_SDI.2 – O.KEY-MNGT*
◦The integrity of key data is protected by security functional requirements defined in FDP_SDI.2.
•*FDP_SDI.2 – O.PIN-MNGT*
◦The integrity of PIN data is protected by security functional requirements defined in FDP_SDI.2.

•*FPT_TDC.1 – O.OPERATE*
◦O.OPERATE is met by FTP_TDC.1 that requires the capability to consistently interpret the CAP files (shared between the card manager and the TOE), the bytecode and its data arguments (shared with applet s and API packages), when shared between the TSF and another trusted IT product.

•*FPT_FLS.1/JCS – O.ALARM*
◦O.ALARM is met by FPT_FLS.1/JCS that requires feedback information upon detection of a potential security violation.
•*FPT_FLS.1/JCS – O.OPERATE*
◦O.OPERATE is met by FPT_FLS.1/JCS by preserving a secure status against potential security violations.
•*FPT_FLS.1/JCS – O.RESOURCES*
◦O.RESOURCES is met by FPT_FLS.1/JCS by preserving a secure status against unavailability of resources.

•*FPR_UNO.1 – O.CIPHER*
◦O.CIPHER is met by FPR_UNO.1 that ensures non-observability of operations on sensitive data.
•*FPR_UNO.1 – O.KEY-MNGT*
◦O.KEY-MNGT is met by FPR_UNO.1 that ensures non-observability of operations on sensitive data.
•*FPR_UNO.1 – O.PIN-MNGT*
◦O.PIN-MNGT is met by FPR_UNO.1 that ensures non-observability of operations on sensitive data.

•*FPT_TST.1 – O.OPERATE*
◦O.OPERATE is met by FPT_TST.1, since startup of the TOE is covered by FPT_TST.1.

•*FMT_MTD.1/JCRE – O.FIREWALL*
◦O.FIREWALL is indirectly met by the functional requirements of FMT_MTD.1/JCRE.
•*FMT_MTD.1/JCRE – O.RESOURCES*
◦O.RESOURCES is met by FMT_MTD.1/JCRE, since memory management is controlled by the TSF.
•*FMT_MTD.1/JCRE – O.SID*
◦Subjects' identity is AID-based (applets, packages) and is met by FMT_MTD.1/JCRE.

•*FMT_MTD.3 – O.FIREWALL*
◦O.FIREWALL is indirectly met by the functional requirements of FMT_MTD.3.
•*FMT_MTD.3 – O.RESOURCES*

◦O.RESOURCES is met by FMT_MTD.3, since memory management is controlled by the TSF.
•*FMT_MTD.3 – O.SID*
◦Subjects' identity is AID-based (applets, packages) and is met by FMT_MTD.3.

•*FIA_ATD.1/AID – O.OPERATE*
◦Since communication with external users and their internal subjects is well-controlled to prevent alteration of TSF data, O.OPERATE is covered by FIA_ATD.1/AID.
•*FIA_ATD.1/AID – O.SID*
◦Subjects' identity is AID-based (applets, packages) and is met by FIA_ATD.1/AID.

•*FIA_UID.2/AID – O.SID*
◦Since installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities, O.SID is covered by FIA_UID.2/AID.

•*FIA_USB.1 – O.OPERATE*
◦Communication with external users and internal subjects of security-critical parts of TOE is well-controlled by security functions defined in FIA_USB.1.
•*FIA_USB.1 – O.SID*
◦Since installation procedures ensure protection against forgery (the AID of an applet is under the control of the TSFs) or re-use of identities, O.SID is covered by FIA_USB.1.

Table 5. Security functional requirements (*InstG*) - security objectives tracing

	O.ALARM	*O.FIREWALL*	*O.OPERATE*	*O.RESOURCES*	*O.SID*	*O.INSTALL*	*O.DELETION*	*O.REMOTE*	*O.SCP.RECOVERY*	*O.SCP.SUPPORT*
FDP_ITC.2/INSTALLER		X	X		X	X				
FMT_SMR.1/INSTALLER		X		X			X	X		
FPT_FLS.1/INSTALLER	X		X	X		X	X		X	
FPT_RCV.3/INSTALLER			X	X		X	X		X	X
FRU_RSA.1/INSTALLER				X						

•FDP_ITC.2/INSTALLER - O.FIREWALL
◦Import of user data should be done in a controlled operations with security attributes to meet O.FIREWALL.
•FDP_ITC.2/INSTALLER – O.OPERATE
◦Since communication with external user and their internal subjects is well-controlled to prevent alteration of TSF data, O.OPERATE is met.
•FDP_ITC.2/INSTALLER – O.SID
◦O.SID is met by FDP_ITC.2/INSTALLER.
•FDP_ITC.2/INSTALLER – O.INSTALL
◦Security attributes of installed data are under the control of FIREWALL access control policy to meet O.INSTALL.
•FMT_SMR.1/INSTALLER – O.FIREWALL
◦FMT_SMR.1/INSTALLER indirectly contribute to meet O.FIREWALL.
•FMT_SMR.1/INSTALLER – O.RESOURCES
◦The memory management is controlled by TSF including the Installer.
•FMT_SMR.1/INSTALLER – O.DELETION
◦FMT_SMR.1/INSTALLER indirectly contribute to meet O.DELETION
•FMT_SMR.1/INSTALLER – O.REMOTE
◦FMT_SMR.1/INSTALLER indirectly contribute to meet O.REMOTE

•FPT_FLS.1/INSTALLER – O.ALARM
◦O.ALARM is partially met by FPT_FLS.1/INSTALLER, since an alarm is raised when installation fails.
•FPT_FLS.1/INSTALLER – O.OPERATE
◦The TSF is able to detect and block various failures or security violations during installation to meet O.OPERATE.
•FPT_FLS.1/INSTALLER – O.RESOURCES
◦The TSF detects stack/memory overflows and manage the failure during installation of applications to meet O.RESOURCES.
•FPT_FLS.1/INSTALLER – O.INSTALL
◦The TSFs are protected against possible failures of the Installer to meet O.INSTALL.
•FPT_FLS.1/INSTALLER – O.DELETION
◦The TSFs are protected against possible failures of the deletion procedures to meet O.DELETION.
•FPT_FLS.1/INSTALLER – O.SCP.RECOVERY
◦O.SCP.RECOVERY is met by FPT_FLS.1/INSTALLER.
•FPT_RCV.3/INSTALLER – O.OPERATE
◦Safe recovery from failure is ensured to meet O.OPERATE.
•FPT_RCV.3/INSTALLER – O.RESOURCES
◦Failed installations are not to create memory leak to meet O.RESOURCES.
•FPT_RCV.3/INSTALLER - O.INSTALL
◦The TSFs are protected against possible failures of the Installer to meet O.INSTALL.
•FPT_RCV.3/INSTALLER - O.DELETION
◦The TSFs are protected against possible failures of the deletion procedures to meet O.DELETION.
•FPT_RCV.3/INSTALLER - O.SCP.RECOVERY
◦FPT_RCV.3/INSTALLER is used to support O.SCP.RECOVERY to assist the TOE to recover in the event of a power failure or signal loss during installation.
•FPT_RCV.3/INSTALLER - O.SCP.SUPPORT
◦FPT_RCV.3/INSTALLER is used to support O.SCP.SUPPORT to assist the TOE to recover in the event of a power failure or signal loss during installation.
•FRU_RSA.1/INSTALLER – O.RESOURCES
◦The TSF detects stack/memory overflows during execution of application to meet O.RESOURCES.

Table 6. Security functional requirements (*AdelG*) - security objectives tracing

	O.ALARM	*O.FIREWALL*	*O.KEY-MNGT*	*O.OPERATE*	*O.PIN-MNGT*	*O.RESOURCES*	*O.SID*	*O.REALLOCATION*	*O.DELETION*	*O.SCP.RECOVERY*
FDP_ACC.2/ADEL									X	
FDP_ACF.1/ADEL									X	
FMT_MSA.1/ADEL		X					X		X	
FMT_MSA.3/ADEL		X					X		X	
FMT_SMR.1/ADEL		X				X			X	
FMT_SMF.1/ADEL		X					X			
FDP_RIP.1/ADEL			X		X			X	X	
FPT_FLS.1/ADEL	X			X		X			X	X

•FDP_ACC.2/ADEL - O.DELETION
◦The non-introduction of security holes is ensured by the ADEL access control policy to meet O.DELETION.

•FDP_ACF1/.ADEL - O. DELETION
◦The non-introduction of security holes is ensured by the ADEL access control policy to meet O.DELETION.
•FMT_MSA.1/ADEL - O.FIREWALL
◦FMT_MSA.1/ADEL indirectly contributes to meet O.FIREWALL.
•FMT_MSA.1/ADEL - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.1/ADEL.
•FMT_MSA.1/ADEL - O.DELETION
◦The functional requirements of FMT_MSA.1/ADEL contributes to meet O.DELETION.
•FMT_MSA.3/ADEL - O.FIREWALL
◦FMT_MSA.3/ADEL indirectly contributes to meet O.FIREWALL.
•FMT_MSA.3/ADEL - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.3/ADEL.
•FMT_MSA.3/ADEL - O.DELETION
◦The functional requirements of FMT_MSA.3/ADEL contributes to meet O.DELETION.
•FMT_SMR.1/ADEL - O.FIREWALL
◦FMT_SMR.1/ADEL indirectly contributes to meet O.FIREWALL.
•FMT_SMR.1/ADEL - O.RESOURCES
◦The memory management is controlled by TSF including the Applet Deletion Manager.
•FMT_SMR.1/ADEL - O.DELETION
◦The functional requirements of FMT_SMR.1/ADEL contributes to meet O.DELETION.
•FMT_SMF.1/ADEL -O.FIREWALL
◦FMT_SMR.1/ADEL indirectly contributes to meet O.FIREWALL.
•FMT_SMF.1/ADEL - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_SMF.1/ADEL.
•FDP_RIP.1/ADEL - O.KEY-MNGT
◦The TSF ensure any previous information is made unavailable to meet O.KEY-MNGT.
•FDP_RIP.1/ADEL - O.PIN-MNGT
◦The TSF ensure any previous information is made unavailable to meet O.PIN-MNGT.
•FDP_RIP.1/ADEL - O.REALLOCATION
◦The TSF ensure that the contents of the re-allocated block shall always be cleared to meet O.REALLOCATION.
•FDP_RIP.1/ADEL - O.DELETION
◦The non-accessibility of deleted data is met by FDP_RIP.1/ADEL.
•FPT_FLS.1/ADEL - O.ALARM
◦O.ALARM is partially met by FPT_FLS.1/ADEL since an alarm is raised when applet deletion fails.
•FPT_FLS.1/ADEL - O.OPERATE
◦The TSF is able to detect and block various failures or security violations during deletion to meet O.OPERATE.
•FPT_FLS.1/ADEL - O.RESOURCES
◦The TSF detects stack/memory overflows and manage the failure during deletion of applications to meet O.RESOURCES.
•FPT_FLS.1/ADEL - O.DELETION
◦The TSFs are protected against possible failures of the deletion procedures to meet O.DELETION.
•FPT_FLS.1/ADEL - O.SCP.RECOVERY
◦O.SCP.RECOVERY is met by FPT_FLS.1/ADEL.

Table 7. Security functional requirements (*RMIG*) - security objectives tracing

	O.FIREWALL	*O.SID*	*O.REMOTE*
FDP_ACC.2/JCRMI	X		X
FDP_ACF.1/JCRMI	X		X
FDP_IFC.1/JCRMI			X
FDP_IFF.1/JCRMI			X
FMT_MSA.1/JCRMI	X	X	X

FMT_MSA.1/EXPORT	X	X	X
FMT_MSA.1/REM_REFS	X	X	X
FMT_MSA.3/JCRMI	X	X	X
FMT_REV.1/JCRMI	X		X
FMT_SMR.1/JCRMI	X		X
FMT_SMF.1/JCRMI	X	X	

•FDP_ACC.2/JCRMI – O.FIREWALL
◦O.FIREWALL is met by the Java Card RMI access control policy.
•FDP_ACC.2/JCRMI - O.REMOTE
◦The access to the TOE's internal data and the flow of information from the card to the CAD required by the Java Card RMI service is under control of the Java Card RMI access control policy.
•FDP_ACF.1/JCRMI - O. FIREWALL
◦O.FIREWALL is met by the Java Card RMI access control policy.
•FDP_ACF.1/JCRMI - O.REMOTE
◦The access to the TOE's internal data and the flow of information from the card to the CAD required by the Java Card RMI service is under control of the Java Card RMI access control policy.
•FDP_IFC.1/JCRMI - O.REMOTE
◦The access to the TOE's internal data and the flow of information from the card to the CAD required by the Java Card RMI service is under control of the JCRMI information flow control policy.
•FDP_IFF.1/JCRMI - O.REMOTE
◦The access to the TOE's internal data and the flow of information from the card to the CAD required by the Java Card RMI service is under control of the JCRMI information flow control policy.
•FMT_MSA.1/JCRMI - O.FIREWALL
◦FMT_MSA.1/JCRMI indirectly contributes to meet O.FIREWALL.
•FMT_MSA.1/JCRMI - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.1/JCRMI.
•FMT_MSA.1/JCRMI - O.REMOTE
◦FMT_MSA.1/JCRMI indirectly contributes to meet O.REMOTE.
•FMT_MSA.1/EXPORT - O.FIREWALL
◦FMT_MSA.1/EXPORT indirectly contributes to meet O.FIREWALL.
•FMT_MSA.1/EXPORT - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.1/EXPORT.
•FMT_MSA.1/EXPORT - O.REMOTE
◦FMT_MSA.1/EXPORT indirectly contributes to meet O.REMOTE.
•FMT_MSA.1/REM_REFS - O.FIREWALL
◦FMT_MSA.1/REM_REFS indirectly contributes to meet O.FIREWALL.
•FMT_MSA.1/REM_REFS - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.1/REM_REFS.
•FMT_MSA.1/REM_REFS - O.REMOTE
◦FMT_MSA.1/REM_REFS indirectly contributes to meet O.REMOTE.
•FMT_MSA.3/JCRMI - O.FIREWALL
◦FMT_MSA.3/JCRMI indirectly contributes to meet O.FIREWALL.
•FMT_MSA.3/JCRMI - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.3/JCRMI.
•FMT_MSA.3/JCRMI - O.REMOTE
◦FMT_MSA.3/JCRMI indirectly contributes to meet O.REMOTE.
•FMT_REV.1/JCRMI - O.FIREWALL
◦FMT_REV.1/JCRMI indirectly contributes to meet O.FIREWALL.
•FMT_REV.1/JCRMI - O.REMOTE
◦FMT_REV.1/JCRMI indirectly contributes to meet O.REMOTE.
•FMT_SMR.1/JCRMI - O.FIREWALL
◦FMT_SMR.1/JCRMI indirectly contributes to meet O.FIREWALL.
•FMT_SMR.1/JCRMI - O.REMOTE
◦FMT_SMR.1/JCRMI indirectly contributes to meet O.REMOTE.

•FMT_SMF.1/JCRMI - O.FIREWALL
◦FMT_SMF.1/JCRMI indirectly contributes to meet O.FIREWALL.
•FMT_SMF.1/JCRMI - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_SMF.1/JCRMI.

Table 8. Security functional requirements (*ODELG*) - security objectives tracing

	O.ALARM	*O.KEY-MNGT*	*O.OPERATE*	*O.PIN-MNGT*	*O.RESOURCES*	*O.REALLOCATION*	*O.OBJ-DELETION*	*O.SCP.RECOVERY*
FDP_RIP.1/ODEL		X		X		X	X	
FPT_FLS.1/ODEL	X		X		X		X	X

•FDP_RIP.1/ODEL - O.KEY-MNGT
◦The TSF ensure any previous information is made unavailable to meet O.KEY-MNGT.
•FDP_RIP.1/ODEL - O.PIN-MNGT
◦The TSF ensure any previous information is made unavailable to meet O.KEY-MNGT.
•FDP_RIP.1/ODEL - O.REALLOCATION
◦The TSF ensure that the contents of the re-allocated block shall always be cleared to meet O.REALLOCATION.
•FDP_RIP.1/ODEL - O.OBJ-DELETION
◦The non-accessibility of deleted data is met by FDP_RIP.1/ODEL.
•FPT_FLS.1/ODEL - O.ALARM
◦O.ALARM is partially met by FPT_FLS.1/ODEL since an alarm is raised when there is an error deleting an object.
•FPT_FLS.1/ODEL - O.OPERATE
◦The TSF is able to detect and block various failures or security violations during deletion to meet O.OPERATE.
•FPT_FLS.1/ODEL - O.RESOURCES
◦The TSF detects stack/memory overflows and manage the failure during deletion of applications to meet O.RESOURCES.
•FPT_FLS.1/ODEL - O.OBJ-DELETION
◦The TSFs are protected against possible failures of the deletion procedures to meet O.DELETION.
•FPT_FLS.1/ODEL - O.SCP.RECOVERY
◦O.SCP.RECOVERY is met by FPT_FLS.1/ODEL.

Table 9. Security functional requirements (*CarG*) - security objectives tracing

	O.FIREWALL	*O.RESOURCES*	*O.SID*	*O.LOAD*
FCO_NRO.2/CM				X
FIA_UID.1/CM				X
FDP_IFC.2/CM				X
FDP_IFF.1/CM				X
FDP_UIT.1/CM				X
FMT_MSA.1/C	X		X	

M				
FMT_MSA.3/CM	X		X	
FMT_SMR.1/CM	X	X		
FMT_SMF.1/CM	X		X	
FTP_ITC.1/CM				X

•FCO_NRO.2/CM - O.LOAD
◦Evidence of the origin of the package during loading procedure is enforced to meet O.LOAD.
•FIA_UID.1/CM - O.LOAD
◦Appropriate identification of each user before allowing any TSF-mediated action is enforced to meet O.LOAD.
•FDP_IFC.2/CM - O.LOAD
◦The integrity of the corresponding data is under the control of the PACKAGE LOADING information flow policy to meet O.LOAD.
•FDP_IFF.1/CM - O.LOAD
◦The integrity of the corresponding data is under the control of the PACKAGE LOADING information flow policy to meet O.LOAD.
•FDP_UIT.1/CM – O.LOAD
◦The integrity of the corresponding data is under the control of the PACKAGE LOADING information flow policy to meet O.LOAD.
•FMT_MSA.1/CM - O.FIREWALL
◦FMT_MSA.1/CM indirectly contributes to meet O.FIREWALL.
•FMT_MSA.1/CM - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.1/CM.
•FMT_MSA.3/CM - O.FIREWALL
◦FMT_MSA.3/CM indirectly contributes to meet O.FIREWALL.
•FMT_MSA.3/CM - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.3/CM.
•FMT_SMR.1/CM - O.FIREWALL
◦FMT_SMR.1/CM indirectly contributes to meet O.FIREWALL.
•FMT_SMR.1/CM - O.RESOURCES
◦FMT_SMR.1/CM indirectly contributes to meet O.RESOURCES.
•FMT_SMF.1/CM - O.FIREWALL
◦FMT_SMF.1/CM indirectly contributes to meet O.FIREWALL.
•FMT_SMF.1/CM - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_SMF.1/CM.

Table 10. Security functional requirements (*SCPG*) - security objectives tracing

	O.ALARM	*O.OPERATE*	*O.PIN-MNGT*	*O.RESOURCES*	*O.DELETION*	*O.SCP.RECOVERY*	*O.SCP.SUPPORT*	*O.SCP.IC*	*O.SIDE_CHANNEL*	*O.RND*
FPT_FLS.1/SCP	X	X		X		X				
FRU_FLT.2/SCP						X				
FPT_PHP.3/SCP								X		
FPT_RCV.3/SCP		X		X	X	X	X			
FPT_RCV.4/SCP							X			

FDP_ITT.1/ SCP							X	X	X	
FPT_ITT.1/ SCP							X	X	X	
FDP_IFC.1 /SCP							X	X	X	
FCS_RND. 1/SCP										X

•FPT_FLS.1/SCP – O.ALARM
◦O.ALARM is partially met by FPT_FLS.1/SCP since the IC raises an alarm when it detects an error.
•FPT_FLS.1/SCP - O.OPERATE
◦The TSF is able to detect and block various failures or security violations during deletion to meet O.OPERATE.
•FPT_FLS.1/SCP - O.RESOURCES
◦The TSF preserves a secure state when stack/memory overflows to meet O.RESOURCES.
•FPT_FLS.1/SCP - O.SCP.RECOVERY
◦O.SCP.RECOVERY is met by FPT_FLS.1/ADEL.
•FRU_FLT.2/SCP - O.SCP.RECOVERY
◦O.SCP.RECOVERY is met by FRU_FLT.2/SCP
•FPT_PHP.3/SCP - O.SCP.IC
◦O.SCP.IC is met by FPT_PHP.3/SCP
•FPT_RCV.3/SCP – O.OPERATE
◦Security-critical parts and procedures of the TOE are protected by a automated safe recovery from failure to meet O.OPERATE.
•FPT_RCV.3/SCP - O.RESOURCES
◦The TSF ensure any failure is not to create memory leaks to meet O.RESOURCES.
•FPT_RCV.3/SCP - O.DELETION
◦The TSFs are protected against possible failures of the deletion process to meet O.DELETION.
•FPT_RCV.3/SCP - O.SCP.RECOVERY
◦OE.SC.RECOVERY is met by FPT_RCV.3/SCP
•FPT_RCV.3/SCP - O.SCP.SUPPORT
◦OE.SC.SUPPORT is met by FPT_RCV.3/SCP
•FPT_RCV.4/SCP - O.SCP.SUPPORT
◦OE.SC.SUPPORT is met by FPT_RCV.4/SCP
•FDP_ITT.1/SCP - O.SCP.SUPPORT
◦O.SCP.SUPPORT includes protecting internal user data transfers as stated in FDP_ITT.1/SCP.
•FDP_ITT.1/SCP - O.SCP.IC
◦O.SCP.IC includes protecting user data from physical probing and chip analisys which includes user data transfers as stated in FDP_ITT.1/SCP.
•FDP_ITT.1/SCP - O.SIDE_CHANNEL
◦O.SIDE_CHANNEL protects user data from side channel leakage when being transferred as stated in FDP_ITT.1/SCP.
•FPT_ITT.1/SCP - O.SCP.SUPPORT
◦O.SCP.SUPPORT includes protecting internal data transfers as stated in FPT_ITT.1/SCP.
•FPT_ITT.1/SCP - O.SCP.IC
◦O.SCP.IC includes protecting data from physical probing and chip analisys which includes user data transfers as stated in FPT_ITT.1/SCP.
•FPT_ITT.1/SCP - O.SIDE_CHANNEL
◦O.SIDE_CHANNEL protects data from side channel leakage when being transferred as stated in FPT_ITT.1/SCP.
•FDP_IFC.1/SCP - O.SCP.SUPPORT
◦O.SCP.SUPPORT includes protecting internal data transfers as stated in FDP_ITT.1/SCP.
•FDP_IFC.1/SCP - O.SCP.IC
◦O.SCP.IC includes protecting data from physical probing and chip analisys which includes user data transfers as stated in FDP_ITT.1/SCP.
•FDP_IFC.1/SCP - O.SIDE_CHANNEL
◦O.SIDE_CHANNEL protects data from side channel leakage when being transferred as stated in FDP_ITT.1/SCP.

•FCS_RND.1/SCP – O.RND
◦O.RND requires a random number generator that is provided as stated in FCS_RND.1/SCP.

Table 12. Security functional requirements - security objectives tracing (Card Manager)

	O.FIREWALL	O.RESOURCES	O.SID	O.PROTECT_DATA	O.OS_OPERATE	O.SIDE_CHANNEL	O.FAULT_PROTECT	O.SECURE_COMM_1	O.LOAD_FILE_VERIFICATION-3	O.ROLES-3	O.APPLICATION_PROVIDER_PRE-APPROVAL_2B	O.CARD_ADMINISTRATOR_PRE-APPROVAL_3	O.APPLICATION_CODE_VERIFICATION-3
FMT_MSA.1/CMGR	X		X										
FMT_MSA.3/CMGR	X		X										
FMT_SMR.1/CMGR	X	X											
FMT_SMF.1/CMGR	X		X										
FCS_CKM.1/RSA				X									
FCS_CKM.1/TRIPLE-DES				X									
FCS_CKM.2/RSA				X									
FCS_CKM.2/TRIPLE-DES				X									
FCS_CKM.3/TRIPLE-DES				X		X		X					
FCS_CKM.3/RSA				X		X							
FCS_CKM.4				X		X							
FCS_COP.1/TRIPLE-DES				X		X		X					
FCS_COP.1/RSA				X		X							
FCS_COP.1/DESMAC				X		X							
FCS_COP.1/RSA SIGNATURE				X		X							
FCS_COP.1/SHA-1				X		X							
FCS_COP.1/MD5				X		X							
FDP_ACC.1/CMGR				X									
FDP_ACF.1/CMGR				X									
FDP_DAU.1/GP					X								
FDP_ITC.2/INSTALLER													X

FDP_RIP.1/OBJECTS				X	X								
FDP_RIP.1/APDU				X	X								
FDP_RIP.1/bArray				X	X								
FDP_RIP.1/TRANSIENT				X	X								
FDP_RIP.1/ABORT				X	X								
FDP_RIP.1/KEYS				X	X								
FDP_RIP.1/ADEL				X	X								
FDP_RIP.1/ODEL				X	X								
FDP_SDI.2				X	X								
FIA_AFL.1/GP				X	X					X			
FIA_ATD.1/GP				X	X					X	X	X	
FIA_UAU.1/GP				X									
FIA_UAU.4/GP								X					
FIA_UID.1/CMGR				X									
FIA_USB.1/GP				X						X		X	
FMT_MOF.1/GP											X	X	
FMT_MSA.1/CMGR									X		X	X	X
FMT_MSA.2/GP					X				X				X
FMT_MSA.3/CMGR									X			X	X
FMT_MTD.1/GP												X	
FMT_SMF.1/GP											X	X	
FMT_SMR.1/CMGR									X	X			X
FPR_UNO.1				X		X							
FPT_FLS.1/SCP					X		X						
FPT_PHP.3/SCP							X						
FPT_RCV.3/SCP					X								
FPT_RCV.4/SCP					X								
FPT_TST.1					X								
FRU_FLT.2/SCP					X								

•FMT_MSA.1/CMGR - O.FIREWALL
◦FMT_MSA.1/CMGR indirectly contributes to meet O.FIREWALL.

•FMT_MSA.1/CMGR - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.1/CMGR.

•FMT_MSA.3/CMGR - O.FIREWALL
◦FMT_MSA.3/CMGR indirectly contributes to meet O.FIREWALL.

•FMT_MSA.3/CMGR - O.SID

◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_MSA.3/CMGR.

•FMT_SMR.1/CMGR - O.FIREWALL
◦FMT_SMR.1/CMGR indirectly contributes to meet O.FIREWALL.

•FMT_SMR.1/CMGR - O.RESOURCES
◦FMT_SMR.1/CMGR indirectly contributes to meet O.RESOURCES.

•FMT_SMF.1/CMGR - O.FIREWALL
◦FMT_SMF.1/CMGR indirectly contributes to meet O.FIREWALL.

•FMT_SMF.1/CMGR - O.SID
◦Subjects' identity is AID-based(applets, packages) and is also met by FMT_SMF.1/CMGR.

•FCS_CKM.1/RSA - O.PROTECT_DATA
◦FCS_CKM.1/RSA covers O.PROTECT_DATA by providing secure keypair generation mechanism for keys used for RSA data encryption.

•FCS_CKM.1/TRIPLE-DES - O.PROTECT_DATA
◦FCS_CKM.1/RSA covers O.PROTECT_DATA by providing secure key generation mechanism for keys used for TRIPLE-DES data encryption.

•FCS_CKM.2/RSA - O.PROTECT_DATA
◦FCS_CKM.2/RSA covers O.PROTECT_DATA by providing secure keypair distribution mechanism for keys used for RSA data encryption.

•FCS_CKM.2/TRIPLE-DES - O.PROTECT_DATA
◦FCS_CKM.2/TRIPLE-DES covers O.PROTECT_DATA by providing secure key distribution mechanism for keys used for TRIPLE-DES data encryption.

•FCS_CKM.3/TRIPLE-DES - O.PROTECT_DATA
◦FCS_CKM.3/TRIPLE-DES covers O.PROTECT_DATA by providing secure key management mechanism.

•FCS_CKM.3/TRIPLE-DES - O.SIDE_CHANNEL
◦FCS_CKM.3 covers O.SIDE_CHANNEL by providing secure key management mechanism.

•FCS_CKM.3/TRIPLE-DES - O.SECURE_COMM_1
◦The TOE provides secure communication protocol using secure channel keys. The keys are managed by secure key management mechanism.

•FCS_CKM.3/RSA - O.PROTECT_DATA
◦FCS_CKM.3/TRIPLE-DES covers O.PROTECT_DATA by providing secure key management mechanism.

•FCS_CKM.3/RSA - O.SIDE_CHANNEL
◦FCS_CKM.3 covers O.SIDE_CHANNEL by providing secure key management mechanism.

•FCS_CKM.4 - O.PROTECT_DATA
◦FCS_CKM.4 covers O.PROTECT_DATA by providing secure key destruction mechanism.

•FCS_CKM.4 - O.SIDE_CHANNEL
◦FCS_CKM.4 covers O.SIDE_CHANNEL by providing secure key destruction mechanism.

•FCS_COP.1/TRIPLE-DES - O.PROTECT_DATA
◦FCS_COP.1/TRIPLE-DES covers O.PROTECT_DATA by providing secure cryptographic operation mechanism that can encrypt data using TRIPLE-DES.

◦FCS_COP.1/TRIPLE-DES - O.SIDE_CHANNEL
◦FCS_COP.1/TRIPLE-DES covers O.PROTECT_DATA by providing secure cryptographic operation mechanism for TRIPLE-DES using .

•FCS_COP.1/TRIPLE-DES - O.SECURE_COMM_1
The TOE provides secure key operation mechanism to establish a secure communication channel.

•FCS_COP.1/RSA SIGNATURE - O.PROTECT_DATA
◦FCS_COP.1 covers O.PROTECT_DATA by providing secure key operation mechanism.

•FCS_COP.1/RSA SIGNATURE - O.SIDE_CHANNEL
◦FCS_COP.1 covers O.SIDE_CHANNEL by providing secure key operation mechanism.

•FCS_COP.1/RSA - O.PROTECT_DATA
◦FCS_COP.1 covers O.PROTECT_DATA by providing secure key operation mechanism.

•FCS_COP.1/RSA - O.SIDE_CHANNEL
◦FCS_COP.1 covers O.SIDE_CHANNEL by providing secure key operation mechanism.

•FCS_COP.1/DESMAC - O.PROTECT_DATA
◦FCS_COP.1 covers O.PROTECT_DATA by providing secure key operation mechanism.

•FCS_COP.1/DESMAC - O.SIDE_CHANNEL
◦FCS_COP.1 covers O.SIDE_CHANNEL by providing secure key operation mechanism.

•FCS_COP.1/SHA-1 - O.PROTECT_DATA
◦FCS_COP.1 covers O.PROTECT_DATA by providing secure operation mechanism.

•FCS_COP.1/SHA-1 - O.SIDE_CHANNEL
◦FCS_COP.1 covers O.SIDE_CHANNEL by providing secure operation mechanism.

•FCS_COP.1/MD5 - O.PROTECT_DATA
◦FCS_COP.1 covers O.PROTECT_DATA by providing secure operation mechanism.

•FCS_COP.1/MD5 - O.SIDE_CHANNEL
◦FCS_COP.1 covers O.SIDE_CHANNEL by providing secure operation mechanism.

•FDP_ACC.1/CMGR - O.PROTECT_DATA
◦O.PROTECT_DATA is partly covered by CARD CONTENT MANAGEMENT access control policy.

•FDP_ACF.1/CMGR - O.PROTECT_DATA
◦O.PROTECT_DATA is partly covered by CARD CONTENT MANAGEMENT access control policy.

•FDP_DAU.1/GP – O.OS_OPERATE
◦FDP_DAU.1/GP covers O.OS_OPERATE by providing functionality which generates and verifies the evidence of information validity.

•FDP_ITC.2/INSTALLER - O.APPLICATION_CODE_VERIFICATION-3
◦O.APPLICATION_CODE_VERIFICAION is partly covered by FDP_ITC.2/Installer which ensure binary compatibility and the interpretation of CAP files.

•FDP_RIP.1/OBJECTS - O.PROTECT_DATA
◦FDP_RIP.1/OBJECTS covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/OBJECTS - O.OS_OPERATE
◦FDP_RIP.1/OBJECTS partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_RIP.1/APDU - O.PROTECT_DATA
◦FDP_RIP.1/APDU covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/APDU - O.OS_OPERATE
◦FDP_RIP.1/APDU partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_RIP.1/bArray - O.PROTECT_DATA
◦FDP_RIP.1/bArray covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/bArray - O.OS_OPERATE
◦FDP_RIP.1/bArray partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_RIP.1/TRANSIENT - O.PROTECT_DATA
◦FDP_RIP.1/TRANSIENT covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/TRANSIENT - O.OS_OPERATE
◦FDP_RIP.1/TRANSIENT partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_RIP.1/ABORT - O.PROTECT_DATA
◦FDP_RIP.1/ABORT covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/ABORT - O.OS_OPERATE
◦FDP_RIP.1/ABORT partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_RIP.1/KEYS - O.PROTECT_DATA
◦FDP_RIP.1/KEYS covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/KEYS - O.OS_OPERATE
◦FDP_RIP.1/KEYS partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_RIP.1/ADEL - O.PROTECT_DATA
◦FDP_RIP.1/ADEL covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/ADEL - O.OS_OPERATE
◦FDP_RIP.1/ADEL partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_RIP.1/ODEL - O.PROTECT_DATA
◦FDP_RIP.1/ODEL covers O.PROTECT_DATA by providing unavailability of previous information.

•FDP_RIP.1/ODEL - O.OS_OPERATE
◦FDP_RIP.1/ODEL partly covers O.OS_OPERATE, since unavailability of previous information prevents unauthorized use of the TOE.

•FDP_SDI.2 - O.PROTECT_DATA
◦FDP_SDI.2 covers O.PROTECT_DATA by monitoring integrity of user data stored in TSF.

•FDP_SDI.2 - O.OS_OPERATE
◦Integrity checking of sensitive user data ensures continued correct operation of TOE’s security functions.

•FIA_AFL.1/GP - O.PROTECT_DATA
◦Sensitive data is protected by limiting the number of unsuccessful authentications.

•FIA_AFL.1/GP - O.OS_OPERATE
◦Continued correct operation of TOE’s security functions is ensured by limiting the number of unsuccessful authentications.

•FIA_AFL.1/GP - O.ROLES-3
◦The TOE is able to recognize the roles defined in O.ROLES-3 using secure channel establishment mechanism.

•FIA_ATD.1/GP - O.PROTECT_DATA
◦O.PROTECT_DATA is partly covered by CARD CONTEMENT MANAGEMENT access control policy and Privilege security attribute represents privilege of each subject in the policy.

•FIA_ATD.1/GP - O.OS_OPERATE
◦O.OS_OPERATE is partly covered by CARD CONTEMENT MANAGEMENT access control policy and Privilege security attribute represents privilege of each subject in the policy.

•FIA_ATD.1/GP - O.ROLES-3
◦Roles defined in O.ROLES-3 are recognized by Privilege security attribute.

•FIA_ATD.1/GP - O.APPLICATION_PROVIDER_PRE-APPROVAL_2b
◦O.APPLICATION_PROVIDER_PRE-APPROVAL_2b is covered by CARD CONTEMENT MANAGEMENT access control policy and Privilege security attribute represents privilege of each subject in the policy.

•FIA_ATD.1/GP - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦O.CARD_ADMINISTRATOR_PRE-APPROVAL_3 is covered by CARD CONTEMENT MANAGEMENT access control policy and Privilege security attribute represents privilege of each subject in the policy.

•FIA_UAU.1/GP – O.PROTECT_DATA
◦O.PROTECT_DATA is covered by FIA_UAU.1/GP, since Card Content management Functions are only allowed after successful authentication.

•FIA_UAU.4/GP - O.SECURE_COMM_1
◦O.SECURE_COMM_1 is partly covered by FIA_UAU.4/GP by providing secure destruction mechanism of authentication data.

•FIA_UID.1/CMGR - O.PROTECT_DATA
◦O.PROTECT_DATA is covered by FIA_UID.1/CMGR, since Card Content Management Functions are not allowed before user identification.

•FIA_USB.1/GP - O.PROTECT_DATA
◦O.PROTECT_DATA is partly covered by CARD CONTEMENT MANAGEMENT access control policy and Privilege security attribute represents privilege of each subject in the policy.

•FIA_USB.1/GP - O.ROLES-3
◦Roles defined in O.ROLES-3 are recognized by Privilege security attribute.

•FIA_USB.1/GP - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦O.CARD_ADMINISTRATOR_PRE-APPROVAL_3 is covered by CARD CONTEMENT MANAGEMENT access control policy and Privilege security attribute represents privilege of each subject in the policy.

•FMT_MOF.1/GP - O.APPLICATION_PROVIDER_PRE-APPROVAL_2b
◦O.APPLICATION_PROVIDER_PRE-APPROVAL_2b is covered by FMT_MOF.1/GP.

•FMT_MOF.1/GP - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦O.CARD_ADMINISTRATOR_PRE-APPROVAL_3 is covered by FMT_MOF.1/GP.

•FMT_MSA.1/CMGR - O.LOAD_FILE_VERIFICATION-3
◦O.LOAD_FILE_VERIFICATION-3 is covered by CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.1/CMGR - O.APPLICATION_PROVIDER_PRE-APPROVAL_2b
◦O.APPLICATION_PROVIDER_PRE-APPROVAL_2b is covered by CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.1/CMGR - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦O.CARD_ADMINISTRATOR_PRE-APPROVAL_3 is covered by CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.1/CMGR - O.APPLICATION_CODE_VERIFICATION-3

◦ O.APPLICATION_CODE_VERIFICATION-3 is covered by CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.2/GP - O.OS_OPERATE
◦Since only secure values are accepted for security attributes of CARD CONTENT MANAGEMENT access control policy, continued correct operation of TOE's security funcions are ensured.

•FMT_MSA.2/GP – O.LOAD_FILE_VERIFICATION-3
◦O.LOAD_FILE_VERIFICATION-3 is covered by FMT-MSA.2/GP, since only secure values are accepted for security attributes of CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.2/GP - O.APPLICATION_CODE_VERIFICATION-3
◦O.APPLICATION_CODE_VERIFICATION-3 is covered by FMT-MSA.2/GP, since only secure values are accepted for security attributes of CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.3/CMGR - O.LOAD_FILE_VERIFICATION-3
◦O.LOAD_FILE_VERIFICATION-3 is covered by FMT_MSA.3/CMGR, since restrictive default values are enforced to security attributes of CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.3/CMGR - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦O.CARD_ADMINISTRATOR_PRE-APPROVAL_3 is covered by FMT_MSA.3/CMGR, since restrictive default values are enforced to security attributes of CARD CONTENT MANAGEMENT access control policy.

•FMT_MSA.3/CMGR - O.APPLICATION_CODE_VERIFICATION-3
◦O.APPLICATION_CODE_VERIFICATION-3 is covered by FMT_MSA.3/CMGR, since restrictive default values are enforced to security attributes of CARD CONTENT MANAGEMENT access control policy.

•FMT_MTD.1/GP - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦O.CARD_ADMINISTRATOR_PRE-APPROVAL_3 is covered by FMT_MTD.1/GP.

•FMT_SMF.1/GP - O.APPLICATION_PROVIDER_PRE-APPROVAL_2b
◦Since TSF is capable to manage card contents in GlobalPlatform registry, O.APPLICATION_PROVIDER_PRE-APPROVAL_2b is covered.

•FMT_SMF.1/GP - O.CARD_ADMINISTRATOR_PRE-APPROVAL_3
◦Since TSF is capable to manage card contents in GlobalPlatform registry, O.CARD_ADMINISTRATOR_PRE-APPROVAL_3 is covered.

•FMT_SMR.1/CMGR - O.LOAD_FILE_VERIFICATION-3
◦O.LOAD_FILE_VERIFICATION-3 is partly covered by FMT_SMR.1/CMGR.

•FMT_SMR.1/CMGR - O.ROLES-3
◦O.ROLES-3 is covered by FMT_SMR.1/CMGR.

•FMT_SMR.1/CMGR - O.APPLICATION_CODE_VERIFICATION-3
◦O.APPLICATION_CODE_VERIFICATION-3 is partly covered by FMT_SMR.1/CMGR.

•FPR_UNO.1 – O.PROTECT_DATA
◦O.PROTECT_DATA is covered by FPR_UNO.1, since non-observability of operations on sensitive information such as keys is ensured.

•FPR_UNO.1 - O.SIDE_CHANNEL
◦O.SIDE_CHANNEL is covered by FPR_UNO.1, since non-observability of operations on sensitive information such as keys is ensured.

•FPT_FLS.1/SCP - O.OS_OPERATE
◦By preserving secure status when failures occur, continued correct operation of security functions of the TOE is ensured.

•FPT_FLS.1/SCP - O.FAULT_PROTECT

◦By preserving secure status when the TOE is used in unsafe environment, correct operation of security functions of the TOE is ensured.

•FPT_PHP.3/SCP - O.FAULT_PROTECT
◦Since TOE is protected from physical attacks, correct operation of security functions of the TOE is ensured.

•FPT_RCV.3/SCP - O.OS_OPERATE
◦Security-critical parts and procedures of the TOE are protected by a automated safe recovery from failure to meet O.OPERATE.

•FPT_RCV.4/SCP - O.OS_OPERATE
◦O.OS_OPERATE is covered by FPT_RCV.4/SCP, since atomic operations to static and object's fields are ensured.

•FPT_TST.1 - O.OS_OPERATE
◦O.OS_OPERATE is covered by FPT_TST.1, since self-testing at initial start-up demonstrate correct operations of the TSF.

•FRU_FLT.2/SCP - O.OS_OPERATE
◦Since the TSF ensures all operations except operations that consume memory when lack of memory occurs, O.OS_OPERATE is partly covered.

Security functional requirements Dependency Table (a KO note means that a rationale for the non inclusion of the dependence is provided):

Table 12.Security functional requirements Dependency Table

SFR	Dependency	Rationale
FDP_ACC.2/FIREWALL	FDP_ACF.1	OK:FDP_ACF.1/FIREWALL
FDP_ACF.1 /FIREWALL	FDP_ACC.1 FMT_MSA.3	OK:FDP_ACC.2/FIREWALL FMT_MSA.3/FIREWALL
FDP_IFC.1 /JCVM	FDP_IFF.1	OK:FDP_IFF.1/JCVM
FDP_IFF.1 /JCVM	FDP_IFC.1 FMT_MSA.3	OK:FDP_IFC.1/JCVM FMT_MSA.3/FIREWALL
FDP_RIP.1 /OBJECTS	N/A	OK
FMT_MSA.1 /JCRE	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	OK:FDP_ACC.2/FIREWALL FDP_IFC.1/JCVM FMT_SMR.1/JCRE FMT_SMF.1.1/JCRE
FMT_MSA.2 /JCRE	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	OK:FDP_ACC.2/FIREWALL FDP_IFC.1/JCVM FMT_MSA.1/JCRE FMT_SMR.1/JCRE
FMT_MSA.3 /FIREWALL	FMT_MSA.1 FMT_SMR.1	OK:FMT_MSA.1/JCRE FMT_SMR.1/JCRE
FMT_SMR.1 /JCRE	FIA_UID.1	OK:FIA_UID.2.1/AID
FMT_SMF.1 /JCRE	N/A	OK
FCS_CKM.1/RSA	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	OK:FCS_CKM.2/RSA FCS_CKM.4
FCS_CKM.1/TRIPLE-DES	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	OK:FCS_COP.1/TRIPLE-DES FCS_CKM.4
FCS_CKM.2/RSA	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1/RSA FCS_CKM.4
FCS_CKM.2/TRIPLE-DES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1/TRIPLE-DES FCS_CKM.4
FCS_CKM.3/TRIPLE-DES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1/TRIPLE-DES FCS_CKM.4
FCS_CKM.3/RSA	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1/RSA FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	OK:FCS_CKM.1/RSA FCS_CKM.1/TRIPLE-DES
FCS_COP.1/TRIPLE-DES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1 FCS_CKM.4

FCS_COP.1/RSA	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1/RSA FCS_CKM.4
FCS_COP.1/DESMAC	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1/TRIPLE-DES FCS_CKM.4
FCS_COP.1/RSA SIGNATURE	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	OK:FCS_CKM.1/RSA FCS_CKM.4
FCS_COP.1/SHA-1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	KO: not satisfied
FCS_COP.1/MD5	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	KO: not satisfied
FDP_RIP.1 /APDU	N/A	OK
FDP_RIP.1 /bArray	N/A	OK
FDP_RIP.1 /TRANSIENT	N/A	OK
FDP_RIP.1 /ABORT	N/A	OK
FDP_RIP.1 /KEYS	N/A	OK
FDP_ROL.1 /FIREWALL	[FDP_ACC.1 or FDP_IFC.1]	OK:FDP_ACC.2/FIREWALL FDP_IFC.1/JCVM
FAU_ARP.1 /JCS	FAU_SAA.1	KO:FAU_SAA.1 is not satisfied.
FDP_SDI.2	N/A	OK
FPT_TDC.1	N/A	OK
FPT_FLS.1 /JCS	N/A	OK
FPR_UNO.1	N/A	OK
FPT_TST.1	N/A	OK
FMT_MTD.1 /JCRE	FMT_SMR.1 FMT_SMF.1	OK:FMT_SMR.1/JCRE FMT_SMF.1.1/JCRE
FMT_MTD.3	FMT_MTD.1	OK:FMT_MTD.1/JCRE
FIA_ATD.1 /AID	N/A	OK
FIA_UID.2 /AID	N/A	OK
FIA_USB.1	FIA_ATD.1	OK:FIA_ATD.1/AID
FPT_FLS.1 /SCP	N/A	OK
FRU_FLT.2 /SCP	FPT_FLS.1	OK:FPT_FLS.1/SCP
FPT_PHP.3 /SCP	N/A	OK
FPT_RCV.3 /SCP	AGD_OPE.1	OK:AGD_OPE.1
FPT_RCV.4 /SCP	N/A	OK
FDP_ITT.1/SCP	[FDP_ACC.1, or FDP_IFC.1]	OK: FDP_IFC.1/SCP

FPT_ITT.1/SCP	N/A	OK
FDP_IFC.1/SCP	FDP_IFF.1	KO: FDP_IFF.1
FCS_RND.1/SCP	N/A	OK
FDP_ITC.2 /Installer	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1] FPT_TDC.1	OK:FPT_TDC.1 FDP_IFC.2/CM FTP_ITC.1/CM
FMT_SMR.1 /Installer	FIA_UID.1	KO:FIA_UID.1
FPT_FLS.1 /Installer	N/A	OK
FPT_RCV.3 /Installer	AGD_OPE.1	OK:AGD_OPE.1
FRU_RSA.1 /Installer	N/A	OK
FDP_ACC.2 /JCRMI	FDP_ACF.1	OK:FDP_ACF.1/JCRMI
FDP_ACF.1 /JCRMI	FDP_ACC.1 FMT_MSA.3	OK:FDP_ACC.2/JCRMI FMT_MSA.3/JCRMI
FDP_IFC.1 /JCRMI	FDP_IFF.1	OK:FDP_IFF.1/JCRMI
FDP_IFF.1 /JCRMI	FDP_IFC.1 FMT_MSA.3	OK:FDP_IFC.1/JCRMI FMT_MSA.3/JCRMI
FMT_MSA.1 /JCRMI	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	OK:FDP_IFC.1/JCRMI FMT_SMR.1/JCRMI FMT_SMF.1/ADEL
FMT_MSA.1 /EXPORT	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	OK:FDP_IFC.1/JCRMI FMT_SMR.1/JCRMI FMT_SMF.1/JCRMI
FMT_MSA.1 /REM_REFS	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	OK:FDP_IFC.1/JCRMI FMT_SMR.1/JCRMI FMT_SMF.1/JCRMI
FMT_MSA.3 /JCRMI	FMT_MSA.1 FMT_SMR.1	OK:FMT_MSA.1/JCRMI FMT_SMR.1/JCRMI
FMT_REV.1 /JCRMI	FMT_SMR.1	OK:FMT_SMR.1/JCRMI
FMT_SMR.1 /JCRMI	FIA_UID.1	OK:FIA_UID.2.1/AID
FMT_SMF.1 /JCRMI	N/A	OK
FDP_RIP.1 /ODEL	N/A	OK
FPT_FLS.1 /ODEL	N/A	OK
FDP_ACC.2 /ADEL	FDP_ACF.1	OK:FDP_ACF.1/ADEL
FDP_ACF.1 /ADEL	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2/ADEL FMT_MSA.3/ADEL
FMT_MSA.1 /ADEL	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	OK:FDP_ACC.2/ADEL FMT_SMR.1/ADEL FMT_SMF.1.1/ADEL
FMT_MSA.3 /ADEL	FMT_MSA.1 FMT_SMR.1	OK:FMT_MSA.1/ADEL FMT_SMR.1/ADEL
FMT_SMR.1 /ADEL	FIA_UID.1	KO:FIA_UID.1

FMT_SMF.1 /ADEL	N/A	OK
FDP_RIP.1 /ADEL	N/A	OK
FPT_FLS.1 /ADEL	N/A	OK
FCO_NRO.2 /CM	FIA_UID.1	OK:FIA_UID.1/CM
FIA_UID.1 /CM	N/A	OK
FDP_IFC.2 /CM	FDP_IFF.1	OK:FDP_IFF.1/CM
FDP_IFF.1 /CM	FDP_IFC.1 FMT_MSA.3	OK:FDP_IFC.2/CM FMT_MSA.3/CM
FDP_UIT.1 /CM	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	OK:FDP_IFC.2/CM FTP_ITC.1/CM
FMT_MSA.1 /CM	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	OK:FDP_IFC.2/CM FMT_SMR.1/CM FMT_SMF.1.1/CM
FMT_MSA.3 /CM	FMT_MSA.1 FMT_SMR.1	OK:FMT_MSA.1/CM FMT_SMR.1/CM
FMT_SMR.1 /CM	FIA_UID.1	OK:FIA_UID.1/CM
FMT_SMF.1 /CM	N/A	OK
FTP_ITC.1 /CM	N/A	OK
FDP_ACC.1 /CMGR	FDP_ACF.1	OK:FDP_ACF.1/CMGR
FDP_ACF.1 /CMGR	FDP_ACC.1 FMT_MSA.3	OK:FDP_ACC.1/CMGR FMT_MSA.3/CMGR
FMT_MSA.1 /CMGR	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	OK:FDP_ACC.1/CMGR FMT_SMR.1/CMGR FMT_SMF.1.1/CMGR
FMT_MSA.3 /CMGR	FMT_MSA.1 FMT_SMR.1	OK:FMT_MSA.1/CMGR FMT_SMR.1/CMGR
FMT_SMR.1 /CMGR	FIA_UID.1	OK:FIA_UID.1/CMGR
FMT_SMF.1 /CMGR	N/A	OK
FIA_UID.1 /CMGR	N/A	OK
FDP_DAU.1 /GP	N/A	OK
FIA_AFL.1 /GP	FIA_UAU.1	OK:FIA_UAU.1/GP
FIA_ATD.1 /GP	N/A	OK
FIA_UAU.1 /GP	FIA_UID.1	OK:FIA_UID.1/CM FIA_UID.1/CMGR
FIA_UAU.4 /GP	N/A	OK
FIA_USB.1 /GP	FIA_ATD.1	OK:FIA_ATD.1/GP
FMT_MOF.1/GP	FMT_SMR.1 FMT_SMF.1	OK:FMT_SMR.1/CMGR FMT_SMF.1.1/GP
FMT_MSA.2/GP	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	OK:FDP_ACC.1/CMGR FMT_MSA.1.1/CMGR FMT_SMR.1/CMGR

FMT_MTD.1/GP	FMT_SMR.1 FMT_SMF.1	OK:FMT_SMR.1/CMGR FMT_SMF.1.1/GP
FMT_SMF.1/GP	N/A	OK

FAU_SAA.1 Potential violation analysis is used to specify the set of auditable events whose occurrence or accumulated occurrence held to indicate a potential violation of the TSP, and any rules to be used to perform the violation analysis. The dependency of FAU_ARP.1/JCS on this functional requirement assumes that a “potential security violation” is an audit event indicated by the FAU_SAA.1 component. The events listed in FAU_ARP.1/JCS are, on the contrary, merely self-contained ones (arithmetic exception, ill-formed bytecodes, access failure) and ask for a straightforward reaction of the TSFs on their occurrence at runtime. The Java Card VM or other components of the TOE detect these events during their usual working order. Thus, in principle there would be no applicable audit recording in this framework. Moreover, no specification of one such recording is provided elsewhere. Therefore no set of auditable events could possibly be defined.

FIA_UID.1 This is required by the component ***FMT_SMR.1*** of the group *InstG*.
However, the role installer defined in this component is attached to an IT security function rather than to a “user” of the CC terminology. The installer does not “identify” itself with respect to the TOE, but is a part of
it. Thus, here it is claimed that this dependency can be left out. The reader may notice that the role is required because of the SFRs on management of TSF data and security attributes, essentially those of the firewall policy.

This is also required by the component ***FMT_SMR.1*** in group *ADELG*. See the explanation in the paragraph above (the role in this case is applet deletion manager).

Moreover, the requirements for FCS_COP.1/SHA-1 and FCS_COP.1/MD5 are not satisfied because no kind of key is used for hash operations.
As stated in the Data Processing Policy referred to in FDP_IFC.1 there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1 and its Data Processing Policy (FDP_IFC.1). Therefore the dependency is considered satisfied.

4.3. Security Assurance Requirements

The selected Assurance Requirements are the ones that form the EAL4 package augmented with:

- AVA_VAN.5
- ALC_DVS.2

4.4. Security assurance requirements rationale

This security assurance requirements where chosen because EAL4+ provides a good security assurance level for the TOE during development and in its operational provides a good security assurance level for the TOE in its environment.

5. TOE Summary Specification

This sections provides a description of how the TOE satisfies all the security functional requirements.

CoreG

FDP_ACC.2/FIREWALL
All subjects and objects defined in this SFR are protected by firewall functionally.

FDP_ACF.1/FIREWALL
Each object, JCRE representation and all the rules defined in this SFR are implemented according to Java VM Card specification 2.2.1.

FDP_IFC.1/JCVM
All subjects and operations defined in this SFR are stored and are implemented according to Java Card VM specification 2.2.1.

FDP_IFF.1/JCVM
All subjects defined in this SFR are stored in object representation according to Java Card VM specification 2.2.1.
All operations defined in this SFR are executed according to Java Card VM specification 2.2.1.

FDP_RIP.1/OBJECTS
The Java Card RE clears all the bytes in any object representation when it is allocated.

FMT_MSA.1/JCRE
Any access to all the security attributes defined in this SFR can be only done by Java Card RE.

FMT_MSA.2/JCRE
The Java Card RE maintains copies of security attribute values and checks with it whether the security attributes are securely managed or in a secure state.

FMT_MSA.3/FIREWALL
All security attributes defined in this SFR are set to default values according to Java Card RE specification 2.2.1. Once it is set to the default value, modification on these security attributes is not allowed at running time.

FMT_SMR.1/JCRE
The TSF obviously specifies the role of Java Card RE according to Java Card VM specification 2.2.1. Also, it distinguishes each user by its security attributes and associates them with appropriate permissions of operations.

FMT_SMF.1/JCRE
All the security management functions defined in this SFR can be performed if and only if the currently active

context is Java Card RE.

FCS_CKM.1
The TSF is capable of cryptographic key generation for each iteration by using functionalities provided by the IC.

FCS_CKM.2
The distributions of cryptographic key is done by the distribution methods defined in this SFR.

FCS_CKM.3
The access of cryptographic key is done by the access methods defined in each iteration of this SFR.

FCS_CKM.4
The destruction of cryptographic key is done by the destruction methods defined in this SFR.

FCS_COP.1
The TOE is capable of performing all cryptographic operations defined in all iterations of this SFR. It also meets each standard as specified.

FDP_RIP.1/APDU
The APDU buffer is implemented as to be static so that allocation occurs only once in the initialization process. Also it is cleared at the end of each and every exchange of APDU.

FDP_RIP.1/bArray
the bArray is not available after allocation.

FDP_RIP.1/TRANSIENT
All transient objects are manged by rules specified in Java Card RE specification 2.2.1.

FDP_RIP.1/ABORT
The TSF makes any references to an object instance created during an aborted transaction unavailable.

FDP_RIP.1/KEYS
All objects defined in this SFR are not available upon invocation of clear method.

FDP_ROL.1/FIREWALL
During an atomic transaction any changes on persistent memory are managed by TSF.

FAU_ARP.1/JCS
At detection of a potential security violations specified in the SFR, the TSF takes security actions to protect the TOE.

FDP_SDI.2
The TSF maintains checksums for application sensitive data and integrity error is detected.

FPT_TDC.1
The TOE is implemented to fully conform to all the rules specified in this SFR by following the Javacard specifications in terms of interpretation of CAP files.

FPT_FLS.1/JCS
Upon any detection of potential security violation, all the operations in process are stopped and appropriate actions are taken.

FPR_UNO.1
All objects defined in this SFR are implemented as an object representation in Java Card RE and they are protected by firewall based on their contexts. The TSF also implement the following physical countermeasures not to allow observability.

FPT_TST.1
The TSF runs a suite of self-tests prior to usage of each part of TOE and implements a maintenance mode

which provides the capability to verify integrity of TSF data to authorized users.

FMT_MTD.1/JCRE
Since any methods of the Applet manager is only accessible with Java Card RE privilege, this SFR is covered.

FMT_MTD.3
The TSF accepts the load file AID only if it dose not exist within the GlobalPlatform registry.

FIA_ATD.1/AID
All security attributes defined in this SFR are implemented. The TSF maintains all the lists of security attributes defined in this SFR as specified in Java Card VM.

FIA_UID.2/AID
During the loading procedure of the package and the registration of applet instances, each user is identified and registered with its AID.

FIA_USB.1
The TSF maintains all the user security attributes defined in this SFR and associate it with subjects to identify each user and to enforce security rules based on it.

InstG

FDP_ITC.2/INSTALLER
The TSF implements an installer which fully conforms with [JCRE221] and it imports user data with security attributes as specified in [JCVM221].

FMT_SMR.1/Installer
The role of the Installer is implemented and specified TSF implements and obviously specifies the role of the Installer according to Java Card RE specification 2.2.1.

FPT_FLS.1/Installer
If the installer fails to load a package or an applet, the TSF will deallocate all the partial data by using Garbage Collector and roll back to its previous state.
If the installer fails to install a package or an applet, the TSF recognizes the failure as an abortion of atomic transaction so it rolls back to the previous status as specified in FDP_ROL.1/FIREWALL.

FPT_RCV.3/Installer
If the installer fails to load a package or an applet, the TSF will deallocate all the partial data by using Garbage Collector and roll back to its previous state.
If the installer fails to install a package or an applet, the TSF recognizes the failure as an abortion of atomic transaction so it rolls back to the previous status as specified in FDP_ROL.1/FIREWALL.

FRU_RSA.1/Installer
The TSF implements maximum quotas for packages, declared classes, methods and fields that packages can use simultaneously as specified in [JCVM221].

ADELG

FDP_ACC.2/ADEL
The TOESF implements the Package Manager/Applet Manager which is in charge of dealing with all the subjects and objects defined in this SFR at applet/package deletion process conforming to Java Card VM 221 specification.

FDP_ACF.1/ADEL
The TSF provides no way to access those deletion methods bypassing the Package Manager/Applet Manager.

FMT_MSA.1/ADEL
The TSF implements the Java Card RE to handle selection of applet and provides no way to the selection

process bypassing Java Card RE.

FMT_MSA.3/ADEL
Any default value for all security attributes defined in this SFR is considered to be restrictive.

FMT_SMR.1/ADEL
The TSF specifies the role of the Applet Deletion Manager according to Java Card RE specification 2.2.1. Also, it identifies each user by its security attributes and associates them with the Applet Deletion Manager enforcing the ADEL access control SFP.

FMT_SMF.1/ADEL
The TSF implements the Java Card RE to handle selection of applet and this functionality covers all the management functions defined in this SFR.

FDP_RIP.1/ADEL
When any of deletion operations defined in this SFR is performed on applet instances and/or packages, the TSF deallocates and clears all of related resources.

FPT_FLS.1/ADEL
The TSF treat the deletion process as an atomic transaction.

RMIG

FDP_ACC.2/JCRMI
All subjects and objects except S.CAD and O.APPLET are implemented to conform all requirements defined in Java Card RMI part of Java Card Runtime specification 2.2.1. O.APPLET is any installed applet. All operations defined in this SFR are also implemented according to the requirements defined in Java Card RMI part of Java Card Runtime specification 2.2.1 and the TOE does not provide any way to access Java Card RMI service without using the operations defined in this SFR.

FDP_ACF.1/JCRMI
Each object and JCRE representation contain all security attributes defined in this SFR as internal attributes of itself and all the rules defined in this SFR are implemented according to Java VM Card specification 2.2.1.

FDP_IFC.1/JCRMI
Java Card Runtime Environment is implemented to conform all requirements defined in Java Card Runtime specification 2.2.1.

FDP_IFF.1/JCRMI
The TSF allows to send a remote object reference descriptor which provide information of the remote object only if the remote object has been exported.

FMT_MSA.1/JCRMI
Selecting an applet on a logical channel is always processed by the Java Card RE.

FMT_MSA.1/EXPORT
The Exported status of a remote object can be modified by invoking internal methods.

FMT_MSA.1/REM_REFS
Returned References lists the remote object references and the SELECT FILE command for Java Card RMI is processed by a registered applet.

FMT_MSA.3/JCRMI
All security attribute of remote object except the Exported status are created and initialized at the creation of the object and they are no longer modifiable.

FMT_REV.1/JCRMI
The lifetime of CLEAR_ON_DESELECT array is managed by Java Card RE and rules that determine the lifetime of remote object references are always enforced.

FMT_SMR.1/JCRMI
The TSF implements internal structure and functionalities to manage package loading and applet installation defined in Java Card specification 2.2.1.

FMT_SMF.1/JCRMI
The Exported status of a remote object can be modified by invoking internal methods.
The applet which owns RMIService object can modify Returned Reference.

ODELG

FDP_RIP.1/ODEL
The TOESF schedules object deletion service and collect the resource as specified in Java Card API 221. TSF ensures that those are completely unavailable upon the deallocation.

FPT_FLS.1/ODEL
The deletion process is threatedTSF treat the deletion process as an atomic transaction.

CarG

FCO_NRO.2/CM
The TSF identifies originator of the information by verifying DAP signature.

FIA_UID.1/CM
The TSF does not allow anything before user identification.

FDP_IFC.2/CM
Information flow between S.BCV and S.CRD is always covered by PACKAGE LOADING information flow control SFP.

FDP_IFF.1/CM
The TSF maintains security attributes related to Seucre Channel Establishment and a security attribute for signature verification.

FDP_UIT.1/CM
The TSF mandates DAP verification upon an application package being loaded to determine whether some operation has occurred.

FMT_MSA.1/CM
The ISD and SSD can establish a secure channel between a terminal and a card by mandated DAP verification.

FMT_MSA.3/CM
The establishment of a secure channel and DAP verification shall always be performed prior to the package loading process.

FMT_SMR.1/CM
The TOE implements the ISD and SSD and provide SSD with mandated DAP verification.

FMT_SMF.1/CM
The TSF modifies the security attribute SecureChannelEstabilshed, SecureLevel and SignatureVerified according to the result of Secure Channel Establishment or DAP verification.

FTP_ITC.1/CM
The TSF supports establishment of a secure communication channel between itself and CAD.

SCPG

FPT_FLS.1/SCP
If the TSF detects an abnormal condition, it throws an exception, restores the previous data or reset the card.

FRU_FLT.2/SCP
The TSF preserves a secure state when exposed to all conditions not menctioned in FPT_FLS.1/SCP.

FPT_PHP.3/SCP
[SAMSUNG IC Samsung S3CC91C 16-bit RISC Microcontroller], physical part of TOE, provides security functionality which is capable of resisting physical attack against the TSF.

FPT_RCV.3/SCP
The TSF ensures automated recovery of all atomic operations on EEPROM.

FPT_RCV.4/SCP
The TSF maintains all the changes on persistent memory that can affect to its secure state in a dedicated buffer.

FDP_ITT.1/SCP
Transfer between parts of the TOE is used for normal operation of the OS and it is enforced by the underlying platform.

FPT_ITT.1/SCP
Transfer between parts of the TOE is used for normal operation of the OS and it is enforced by the underlying platform.

FDP_IFC.1/SCP
Transfer between parts of the TOE is used for normal operation of the OS and it is enforced by the underlying platform.

FCS_RND.1/SCP
A well functioning RNG is provided by the underlying platform.

CMGRG
FDP_ACC.1/CMGR
The TSF implements all objects, subjects and operations defined in this SFR.

FDP_ACF.1/CMGR
The TSF implements all objects and its security attributes defined in this SFR.

FMT_MSA.1/CMGR
The modifications of the security attribute named Verified and Authorized are restricted to the entity who knows the cryptographic key: controlling authority and card issuer.

FMT_MSA.3/CMGR
DAP verification by the controlling authority and authorization by the delegated CCMFs always should be prior to any card content management operation.

FMT_SMR.1/CMGR
The TSF defines roles of Controlling authority, Application Provider and Card issuer as specified in [GP-SPEC] and associates users with each of those roles.

FMT_SMF.1/CMGR
The TSF implements the Card Manager.

FIA_UID.1/CMGR
The TSF do not allow any card content management operation to unidentified users. Also, each user needs to modify the security attributes.

General GP Security functional requirements

FDP_DAU.1/GP
The TSF verifies checksum for every Java Card object and updates object checksum.

FIA_AFL.1/GP
The TSF blocks the TOE if the defined numbers of times unsuccessful authentication attempts has been occurred.

FIA_ATD.1/GP
The TSF maintains Application Privilege for each installed applet instance.

FIA_UAU.1/GP
The TSF only allows GET DATA and INITIALIZE UPDATE commands if secure communication channel has not been established.

FIA_UAU.4/GP
The TSF ensures that a session key used to establish a secure channel is distinct from any other previous key.

FIA_USB.1/GP
The TSF associate Application Privilege with an applet instance when the applet instance is installed.

FMT_MOF.1/GP
The ISD is the only subject able and modify Card Content management Functions. Although SSD has Delegated Management privilege, it shall get authorization from the ISD.

FMT_MSA.2/GP
The TSF always enforce the requirements for DAP verification and Delegated Management.

FMT_MTD.1/GP
The ISD can freely modify card content in GlobalPlatform registry and SSD is able to modify it after getting authorization from the ISD.

FMT_SMF.1/GP
The TSF provides Delegate Management and authorized users can perform Card Content Management Functions.

6. Definitions and Abbreviation

APDU	Application Protocol Data Unit, an ISO [7816-4] defined communication format between the card and the off-card applications
Applet	Any Java Card technology-based application
AS	Application software, it is the part of ES in charge of the Application of the Smart Card IC.
BS	Basic Software Is the part of ES in charge of the generic functions of the Smart Card IC such as Operating System, general routines and Interpreters.
CAD	Card acceptance device, a physical device used to communicate with the card.
CAP	Converted applet format, a CAP file contains a binary representation of a package of classes that can be installed on a device and used to execute the package's classes on a Java Card virtual machine
Cardholder	The end user of the card
CM	Card Manager, generic term for the card management entities of a GlobalPlatform card i.e. the OPEN, Issuer Security Domain and a Cardholder Verification Method services provide
DAC	Discretionary Access Control
DAP Verification	A mechanism used by a Security Domain to verify that a Load File Data Block is authentic
DS	Dedicated Software, it is defined as the part of ES provided to test the component and/or to manage specific functions of the component.
ES	Embedded Software, it is defined as the software embedded in the Smart Card Integrated Circuit. The ES may be in any part of the non-volatile memories of the

	Smart Card IC.
Firewall	The mechanism in the Java Card technology for ensuring applet isolation and object sharing.
IC	Integrated Circuit , Electronic component(s) designed to perform processing and/ or memory functions.
Initialization	The process to write specific information in the NVM during IC manufacturing and testing (phase 3) as well as to execute security protection procedures by the IC manufacturer. The information could contain protection codes or cryptographic keys.
Initialization Data	Specific information written during manufacturing or testing of the TOE
ISD	Issuer Security Domain, the primary on-card entity providing support for the control, security, and communication requirements of the card administrator (typically the Card Issuer)
Personalizer	Institution (or its agent) responsible for the Smart Card personalization and final testing.
Personalization data	Specific information in the NVM during personalization phase
SCP	Smart Card Platform. It is comprised of the integrated circuit, the operating system and the dedicated software of the smart card.
Subject	An active entity within the TOE that causes information to flow among objects or change the system's status. It usually acts on the behalf of a user. Objects can be active and thus are also subjects of the TOE.

7. Versioning

Version	Description
1.0	Whole document

References

7816-3	Organization, International Standarization, Identification cards - Integrated circuit(s) - Electrical interface and transmission protocols, ISO/IEC FCD 7816-3, vesrion 2.1c2, 2004-10-1
7816-4	Organization, International Standarization, Identification cards - Integrated circuit(s) - Interindustry commands for interchange, ISO/IEC FCD 7816-4, version 2.1, 2003-01-17
BSI-PP	Smartcard IC Platform Protection Profile BSI-PP-0002 version 1.0, July 2001
JC-PP	Java CardTM System Protection Profile Collection, version 1.0b
JVM	The Java Virtual Machine Specification. Lindholm, Yellin. ISBN 0-201-43294-3
ESI	ETSI, Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms, V1.2.1 (2005-07)
JCAPI221	Application Programming Interface - Java Card Platform, Version 2.2.1
FIPS46-3	Laboratory, Information Technology, Data Encryption Standard (FIPS PUB 46-3), 1999 October 25
PKCS1v1.5	PKCS \#1 v2.1: RSA Cryptography Standard
FIPS180-1	Laboratory, Information Technology, Secure Hash Standard (FIPS PUB 180-1)
RFC1321	Rivest, R., RFC-1321 The MD5 Message-Digest Algorithm

JCRE221	Runtime Environment Specification - Java Card Platform, Version 2.2.1
7816-6	Organization, International Standarization, Identification cards - Integrated circuit(s) cards with contacts- Part 6: Interindustry data elements,
JCVM221	Virtual Machine Specification - Java Card Platform, Version 2.2.1
GP-SPEC	GlobalPlatform Card Specification Version 2.1.1 March 2003