Security Target

TimeCOS Java Card Platform and EasyCard v1.0

Watchdata System Co., Ltd.

Executive Summary: This document is the Security Target of TimeCOS Java Card Platform and EasyCard, meeting the requirements of ASE class “Security Target evaluation” defined in Common Criteria v3.1 R3 Part 3 “Security Assurance Components”.

State : [ ] Draft [ ] Modified [√] Published	***Secret Rank***	High	***Current Version***	1.8
	Author	Shouqin Tang Qiuliang Cheng Wanli Sun Shaobo Li Yijun Wang	***Finished Date***	2013.05.14
	Auditor	N/A	***Date***	N/A
	Confirmer	N/A	***Date***	N/A

History

Version	Date	Author	Description
0.1	2011.06.01	Shouqin Tang	Create this document as initial version
0.2	2011.11.25	Qiuliang Cheng	Adjust the document structure, and do the comments
0.3	2011.12.06	Qiuliang Cheng	Fix the related content accordingly
0.4	2011.12.26	Shaobo Li	Fix the EasyCard related content
0.5	2012.02.10	Qiuliang Cheng Shaobo Li Wanli Sun	Fix the associated content accordingly
0.6	2012.02.15	Qiuliang Cheng Shaobo Li	Fix the Mifare Isolation mechanism description
1.0	2012.04.16	Qiuliang Cheng	Modify life-cycle definition and IC type
1.1	2012.05.16	Qiuliang Cheng	Add T.RND and O.RND
1.2	2012.07.18	Qiuliang Cheng	Change EEPROM word to NVM
1.3	2012.10.26	Qiuliang Cheng	Add section 1.2 “Product Architecture”, and add some example applets in figure of “TOE Architecture”
1.4	2012.10.30	Qiuliang Cheng	Add some example applets in section 1.5 “TOE Intended Usage”
1.5	2013.01.14	Qiuliang Cheng	Fixing observation detected by CB
1.6	2013.01.16	Qiuliang Cheng	Update TOE version
1.7	2013.04.27	Qiuliang Cheng	Add Java Card PP reference No.; Change FCS_COP.1/API to FCS_COP.1/API_DES and FCS_COP.1/API_RSA; Change FCS_CKM.1/API to FCS_CKM.1/API_DES and FCS_CKM.1/API_RSA
1.8	2013.05.14	Qiuliang Cheng	Fix the description of FAU_ARP.1

Table of Content

1. Purpose and Scope

This document is the Security Target of TimeCOS Java Card Platform and EasyCard, as required by the ASE class “Security Target evaluation” in the package of EAL4 defined in the Common Criteria version 3.1 Release 3, Part 3 “Security assurance components”, including ASE_CCL.1, ASE_ECD.1, ASE_INT.1, ASE_OBJ.2, ASE_REQ.2, ASE_SPD.1, and ASE_TSS.1.

1.1. Identification

1.1.1. Security Target Identification

The Security Target of TimeCOS Java Card Platform and EasyCard has following reference information:

- Title: TimeCOS Java Card Platform and EasyCard Security Target
- Version: 1.8
- Issue Date: 2013.05.14
- Reference: SEC_20110121_963_ASE

1.1.2. TOE Identification

The TOE of TimeCOS Java Card Platform and EasyCard has following reference information:

- Title: TimeCOS Java Card Platform and EasyCard
- Composed of:
  - TimeCOS Java Card Platform, compatible with [JCRE30], [JCAPI30], [JCVM30], [GPCS], and [VGPCIR]
  - EasyCard Application, compatible with [CPU_FS_ECC] and [KMS_ECC]
- Version: 1.1

1.2. TOE Overview

The TOE is the Smart Card Platform that is composed of:

- **SLE78CLFX**: The IC underlying platform [ICST], SLE78CLFX4000PM and SLE78CLFX2400PM, with the following libraries: RSA v1.02.013, EC v1.02.013, SHA-2 v1.01 and Toolbox v1.02.013
- **TimeCOS Java Card Platform**: The dedicated COS, composed of
  - **Chip Driver**: The driver for access to the security chip of SLE78CLFX4000PM and SLE78CLFX2400PM
  - **Watchdata OS**: The underlying OS, providing the access to the

functionalities of SLE78CLFX

- **JCS**: The Java Card System, fulfilling the specification of Java Card RTE (including VM, RE and API) version 3.0.1 Classic Edition
- **GP**: The GlobalPlatform, fulfilling the specification of GlobalPlatform (including OPEN, ISD, SD and API) version 2.1.1

• **EasyCard**: The native application, relaying on the Watchdata OS

The current Security Target claims for a level of assurance of evaluation EAL4+ augmented with AVA_VAN.5 and ALC_DVS.2 for the TOE. And it is conformant to the Java Card System Protection Profile Open Configuration base on Open 2.2.x and 3.0.1 Classic Edition configurations.

The TOE is designed and used as a multi-application platform, which provides the capabilities to the issuer for performing the installation, updating and deletion of various Java Card applets. But, the post-issued Java Card applets are outside of the scope of the current evaluation.

Additionally, the TOE includes a native application, named as EasyCard, which provides payment functionalities, such as the electronic purse used for transactions in public transportations and parking.

The TOE provides dual interfaces for a maximum flexibility in using different communication protocols: ISO 7816 and ISO 14443 (including Type A and Type B).

1.3. TOE Description

The TOE includes several components: IC underlying platform, OS, Java Card System, GlobalPlatform (compatible with Visa GlobalPlatform specification), and a native application EasyCard. The following picture describes the TOE architecture:

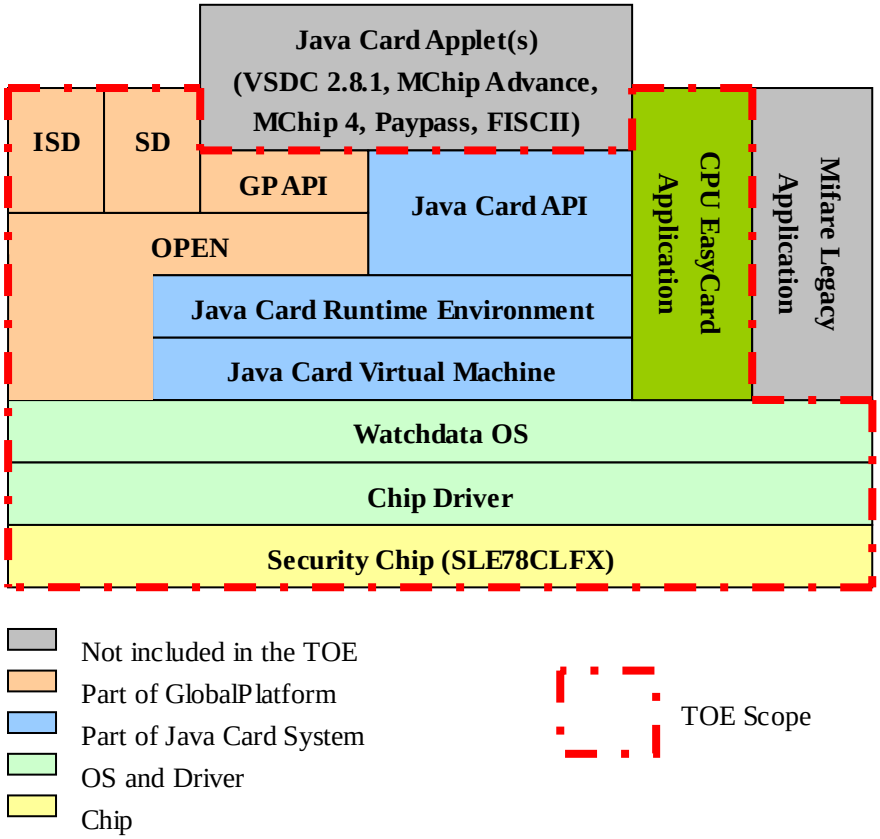


Fig. 1.1 TOE Architecture

The EasyCard is a native application included in the TOE, which fulfils the EasyCard Specification [CPU_FS_ECC] and [KMS_ECC] with the following functionalities:

- Supporting the following payment functions:
  - Read purse data
  - Debit transaction
  - Extended Debit transaction
  - Partial Refund transaction
  - Credit transaction
  - Auto-Load transaction
  - Cancel debit transaction
  - Cancel credit transaction
  - Read debit transaction Log
- Supporting the CPU functions.

To support the above mentioned functionalities of the EasyCard, the Watchdata private OS “Watchdata OS” implements the following functionalities:

- File System: according to ISO 7816-4,
- Access control for the file system and the cryptographic services,

- secure messaging for external communication via a trusted channel (TC),
- Selection and management of security environments;
- User authentication with passwords,
- Component authentication with symmetric and asymmetric cryptographic keys.

The EasyCard application allows a configuration called legacy application that allows the debit and credit transaction using Mifare technology. This configuration is out of the scope of the current evaluation due to the weakness of the Mifare protocol.

The TimeCOS Java Card Platform fulfils the followings specifications:

- The Java Card System (including VM, RE and API) version 3.0.1
- The GlobalPlatform (including OPEN, ISD, SSD and API) version 2.1.1

The Java Card System fulfills the following specifications:

- Java Card Runtime Environment (JCRE), see [JCRE30]
- Java Card Virtual Machine (JCVM), see [JCVM30]
- Java Card Application Programming Interface (JCAPI), see [JCAPI30]

The GlobalPlatform fulfils the GlobalPlatform v2.1.1 specification [GPCS], but is compatible with the Visa GlobalPlatform v2.1.1 specification [VGPCIR].

The Card Manger is implemented by the GlobalPlatform, which provides the Issuer Security Domain (ISD), Supplementary Security Domains (SSD), GlobalPlatform Registry, Open GlobalPlatform Environment (OPEN), GlobalPlatform API, Cardholder Verification Method (CVM) and DAP.

The Secure Channel protocols, SCP01 and SCP02, provide the service of authentication for Card Content Management functionalities.

Java Card Applets are outside of the TOE scope, i.e. they are not considered as the part of the TOE.

The IC underlying platform is SLE78CLFX4000PM and SLE78CLFX2400PM with the library defined in [ICST].

The IC underlying platform provides a real 16-bit CPU-architecture and is compatible to the Intel 80251 architecture. The major components of the core system are the two CPUs (Central Processing Units), the MMU (Memory Management Unit) and MED (Memory Encryption/Decryption Unit). The two CPUs control each other in order to detect faults and serve by this for data integrity. The TOE implements a full 16 MByte linear addressable memory space for each privilege level, a simple scalable Memory Management concept and a scalable stack size. The flexible memory concept consists of ROM-and Flash-memory as part of the non volatile memory

(NVM), respectively NVM. For the NVM memory the Unified Channel Programming (UCP) memory technology is used.

The two cryptographic co-processors serve the need of modern cryptography: The symmetric co-processor (SCP) combines both AES and Triple-DES with dual-key or triple-key hardware acceleration. The Asymmetric Crypto Co-processor, called Crypto2304T in the following, is an optimized version of the Crypto@1408 used in the SLE88-family with performance improvements for RSA-2048 bit (4096-bit with CRT) and Elliptic Curve (EC) cryptography.

The SHA-library provides the calculation of a hash value of freely chosen data input in the CPU. The SHA-library is delivered as object code and is in this way available for the user software.

This secure hash-algorithm SHA-2 is intended to be used for signature generation, verification and generic data integrity checks. The use for keyed hash operations like HMAC or similar security critical operations involving keys, is not subject of this TOE and requires specific security improvements and DPA analysis including the operating system, which is not part of this TOE.

The toolbox library does not provide cryptographic support or additional security functionality as it provides only the following basic long integer arithmetic and modular functions in software, supported by the cryptographic coprocessor: Addition, subtraction, division, multiplication, comparison, reduction, modular addition, modular subtraction, modular multiplication, modular inversion and modular exponentiation. No security relevant policy, mechanism or function is supported.

This dual interface controller is able to communicate using either the contact based or the contactless interface. The implemented dual interface provides a maximum flexibility in using different communication protocols: ISO 7816, ISO 14443 Type A and Type B, FELICA® - ISO/IEC 18092 passive mode, Mifare compatible Interface or the Digital Contactless Bridge (DCLB) mode.

The following is a list of features provided by this IC underlying platform:

- Active shielding with intelligent shielding algorithm finishes the upper layers
- Memory bus supporting the AXITM protocol (Advanced eXtensible Interface) and an APBTM (Advanced Peripheral Bus) for high-speed communication with the peripherals.
- The TRNG fulfils the requirements from the functionality class P2 of the AIS31
- Error detection unit (EDU) automatically manages the error detection of the individual memories and detects incorrect transfer of data between the memories
- Memory Encryption and Decryption unit (MED)

1.3.1. TOE Logical Functionalities

The Java Card Runtime Environment Subsystem security functions contains core group such as Java Card Firewall policy and Card Security Management, Object deletion, and Logical channels.

The Java Card Runtime Environment Subsystem contains four modules as follow ones: Java Card Virtual Machine module, object management module, transaction management module, and logical channels management.

Java Card API module is a component of Java Card Runtime Environment Subsystem. It's interface between user Applet and card service. User Applet uses Java Card API to accesses card service.

Java Card API is a subset of Java API. This part is for the Java Card 3.0.1 Application Programming Interface about package, class, method and variable

The ISD represents the Card Issuer to provide the associated applications the security services.

The GlobalPlatform Registry holds the information managed by the OPEN to perform the GlobalPlatform functionality.

The GP Registry is used to store card management information, store application management information, support card resource management, store Application Life Cycle information, store card Life Cycle information, and track any counter associated with logs.

The OPEN is the GlobalPlatform Environment. It provides the API, dispatch the commands, select Applications, manage logical channels, and manage Card Content.

The OPEN owns and uses the GlobalPlatform Registry as an information resource for Card Content management.

The secure channel protocols, SCP01 and SCP02 provide the service of authentication on Card Content Management.

The Supplementary Security Domains are established as privileged applications on card to represent Application Providers. These Security Domains shall contain separated keys from the Card Issuer to provide related services to associated applications.

The CVM (Cardholder Verification Method) may be used by any privileged Application to access CVM verification service. The CVM handler shall reserve memory space for the CVM management, provide the CVM service to Applications, change the CVM state, and change the Retry Limit and Retry Counter.

The DAP verification allows an Application Provider to perform the authorization on load process with his Supplementary Security Domain.

The Mandated DAP verification allows a Controlling Authority to always perform the load process authentication with his Supplementary Security Domain. It allows the

Security Domain to have DAP Verification privilege, and thereby authorize or veto the loading of any application associated with that Security Domain. Multiple DAP blocks may be included in a single LOAD command.

The TOE allows the isolation between the Java Card System and the native application (EasyCard) by means of extended firewall that allows filtering by means of context.

1.3.2. TOE Guidance

The following guidance are provided by the TOE developer to the customer:

- [AGDOPE]
- [AGDPRE]

1.4. TOE Life-Cycle

The TOE life-cycle is consistent of the following phases:

- Phase 1: Platform Software Development
- Phase 2: IC Development
- Phase 3: Masking & Testing
- Phase 4: Packaging & Testing at IC Manufacturing Site
- Phase 5: Set to OP-READY
- Phase 6: Personalization & Issuance
- Phase 7: Post Issuance

The TOE is in usage phase starting from the end of IC manufacturing.

The purpose of the TOE security functions developed is to control and protect the TOE assets during the usage phase. And the security functions shall cover all usage phases.

All the necessary software for Security Functionalities exists in the ROM at end of IC manufacturing, and the way to use these functionalities is described in the TOE Administrator and User Guidance.

1.4.1. TOE Life Cycle Stages

Development Environment

1.4.2. In Phase 1

The OS Platform Developer develops the code of the basic card OS, RTE and GP software and the EasyCard application.

Production Environment

1.4.3. In Phase 2 & 3

The IC Manufacturer integrates the OS, RTE and GP software and the EasyCard application within the IC ROM memory.

Installation Environment

1.4.4. In Phase 4

The Card Manufacturer integrates the masked IC with the carrier, in accordance with the Card Issuer's requirements, to produce a complete card ready for delivery to the Card Enabler.

Personalization Environment

1.4.5. In Phase 5 & 6

The Card Issuer issues the cards to the Cardholders, and establishes the policies used by the Card Administrator.

The Card Enabler prepares the card for subsequent application loading by personalizing the Platform according to the instructions of the Card Issuer.

The Card Issuer loads the personalization data from EasyCard Applications in flash.

Usage Environment

1.4.6. In Phase 7

Within the policy constraints set by the Card Issuer, the Card Administrator has the ultimate control of the card with regards to card content management and card lifecycle management.

The Application Providers manage the Applications that reside on the cards and provide a card-based service to their customers (i.e. Cardholders).

The Card Administrator may delegate some card content management functions (CCMFs) to the Application Providers.

The Application Loader loads the Applications and/or personalization data on to the card according to the instructions of the Application Provider, complying with the security policies and procedures set by the Card Administrator.

The Controlling Authority has ultimate veto on the loading of Applications, e.g. make sure that it is compliant with national security or other legal/regulatory requirements.

The Verification Authority performs the off-card Application Code Verification and may also perform the on-card Application Code Verification.

The Cardholder is the person or entity that uses the card.

1.5. TOE Intended Usage

The TOE is used as a multi-application platform, which provides management capabilities to the card-issuer for performing the installation, updating and deletion of different applications (e.g. VSDC 2.8.1, MChip Advance, MChip 4, Paypass, and FISCII) that runs over the TimeCOS. The applets over the Java Card Platform are outside of the scope of the current security target.

The TOE uses the EasyCard application for electronic purse transactions in public transportations and parking lot. The EasyCard application is masked in the TOE and cannot be deleted. The EasyCard application is inside of the scope of the current evaluation.

2. CC Conformance Claim

This ST claims conformity with:

- Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 3, July 2009. Part 1
- Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 3, July 2009. Part 2 extended with FCS_RND.1 and FPT_EMSEC.1
- Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 3, July 2009. Part 3

2.1. Protection Profile Conformance Claim

This security target claims the following conformance with protection profiles:

- A demonstrable conformance with [JCS-OP-PP] “Java Card System Protection Profile Open Configuration version 2.6” (Reference No.: ANSSI-CC-PP-2010/03).

2.2. Security Target Package Conformity

This security target and the TOE claim conformity with EAL4+, augmented with AVA_VAN.5 (Advanced methodical vulnerability analysis) and ALC_DVS.2 (Sufficiency of security measures).

2.3. Security Target Conformance Claim Rationale

The TOE type of the current security target is “Java Card 3.0.1 conformant with Visa GlobalPlatform 2.1.1, implemented on Infineon SLE78CLFX4000PM and SLE78CLFX2400PM [ICST], and in the protection profile, the TOE type is “smart card platform enabled with Java Card technology”.

Therefore, the TOE types are compatible, since the security target’s TOE is a smart card that is enabled with Java Card technology.

3. Security Problem Definition

3.1. Assets

3.1.1. User Data

D.APP_CODE The code of the applets and libraries loaded on the card. To be protected from unauthorized modification.

D.APP_C_DATA The confidential sensitive data of the applications. To be protected from unauthorized disclosure.

D.APP_I_DATA The integrity sensitive data of the applications. To be protected from unauthorized modification.

D.PIN Any end user’s PIN. To be protected from unauthorized disclosure and modification.

D.APP_KEY The cryptographic keys owned by the applets. To be protected from unauthorized disclosure and modification.

D.APP_EASY_KEYS EasyCard application AES keys used for the debit or credit applications.

D.APP_DT_LOGs Debit transaction logs for debit transaction are present as records in the file EF with SFI=’1E’. It can be read freely through read record command and only updated based on the EasyCard operative.

D.APP_APPLICATION_DATA The EasyCard application data consists of the application EF files with SFI ranging from ‘01’ to ‘1D’. These application files exist in the card in the form of DGIs (Data Grouping Identifier) rather than in the form of files. Application data are defined by the issuers of the EasyCard application.

3.1.2. TSF Data

D.API_DATA	Private data of the API, like the contents of its private fields. To be protected from unauthorized disclosure and modification.
D.CRYPTO	Cryptographic data used in runtime cryptographic computations, like a seed used to generate a key. To be protected from unauthorized disclosure and modification.
D.JCS_CODE	The code of the Java Card System. To be protected from unauthorized disclosure and modification.
D.JCS_DATA	The internal runtime data areas necessary for the execution of the Java Card VM, such as, for instance, the frame stack, the program counter, the class of an object, the length allocated for an array, any pointer used to chain data-structures. To be protected from unauthorized disclosure or modification.
D.SEC_DATA	The runtime security data of the Java Card RE, like, for instance, the AIDs used to identify the installed applets, the currently selected applet, the current context of execution and the owner of each object. To be protected from unauthorized disclosure and modification.
D.APP_PURSE	EasyCard Purse Data contains the attributes of the purse, including the EM (Electronic Money) and the purse usage control. It can be read freely through read purse command. It is present in the card in the form of DGIs.
D.APP_FCI	FCI Data contains the access rights of user files and their protected key's key type and key version number and the record limit of this file. It can be read freely through get data command. It is present in the card in the form of DGIs (Data Grouping Identifier).
D.APP_ALC	The EasyCard application life cycle data is used to store the current status of the application. In each life cycle only a set of commands can be allowed, others are forbidden. It is an NVM data.

3.2. Users and Subjects

S.CARD_HOLDER	The card holder is the person that is in possession of the EasyCard application and uses it for EM payment transactions. Card holders need to protect their EasyCard application in the same way as cash.
S.TERMINAL_DEVICE	A terminal device is any technical system communicating with the TOE through the contactless interface.

S.ISSUER	The issuer guarantees the EM in an EM system, such as the bank. • creates and dispenses EM in exchange for funds received, • redeems collected EM and extinguishes it. It is also the responsible to management of the ISD with the card manager capabilities provided by Visa Global Platform specifications.
S.APPLET_DEV	The Java Card Applet developers.
S.VAL_VER	The third party or the issuer which validates the byte-code with off-card verifier and sign the applet with private key for being after validated by the DAP in the Card.
S.LOAD_AGENT	A load agent is a trusted agent of an issuer. He executes the load transactions with the EasyCard application on behalf of the issuer and operates a terminal device for this purpose. A load agent is responsible for the operational security of its part of the EM system, and must protect the load devices he controls against authorized use. He is also responsible for transferring payment received from the card holder to the issuer for settlement.
S. MERCHANT	A merchant sells goods or services or tickets for which he accepts payment by the EasyCard application. In order to handle the EM payment transactions, the merchant operates one or more terminal devices. The merchant is responsible for the operational security of the terminal device he controls.

3.3. Threats

This section introduces the threats to the assets against which specific protection is applied in the TOE.

3.3.1. Confidentiality

T.CONFID-APPLI-DATA	The attacker executes an application to disclose data belonging to another application. See #.CONFID-APPLI-DATA for details. Directly threatened asset(s): D.APP_C_DATA, D.PIN and D.APP_KEYs.

T.CONFID-JCS-CODE The attacker executes an application to disclose the Java Card System code. See #.CONFID-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.

T.CONFID-JCS-DATA The attacker executes an application to disclose data belonging to the Java Card System. See #.CONFID-JCS-DATA for details. Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.

T.CONFID-EASY-DATA The attacker tries to get the confidential data of D.APP_EASY_KEYS, and D.APP_APPLICATION_DATA through the communication interface of the TOE by sending commands or by listening to the communication between a terminal and the TOE

3.3.2. Integrity

T.INTEG-APPLI-CODE The attacker executes an application to alter (part of) its own code or another application's code. See #.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

T.INTEG-APPLI-CODE.LOAD The attacker modifies (part of) its own or another application code when an application package is transmitted to the card for installation. See #.INTEG-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

T.INTEG-APPLI-DATA The attacker executes an application to alter (part of) another application's data. See #.INTEG-APPLI-DATA for details. Directly threatened asset(s): D.APP_I_DATA, D.PIN and D.APP_KEYs.

T.INTEG-APPLI-DATA.LOAD The attacker modifies (part of) the initialization data contained in an application package when the package is transmitted to the card for installation. See #.INTEG-APPLI-DATA for details. Directly threatened asset(s): D.APP_I_DATA and D_APP_KEY.

T.INTEG-JCS-CODE The attacker executes an application to alter (part of) the Java Card System code. See #.INTEG-JCS-CODE for details. Directly threatened asset(s): D.JCS_CODE.

T.INTEG-JCS-DATA The attacker executes an application to alter (part of) Java Card System or API data. See #.INTEG-JCS-DATA for details. Directly threatened asset(s): D.API_DATA, D.SEC_DATA, D.JCS_DATA and D.CRYPTO.

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

T.INTEG-EASY-DATA The attacker tries to compromise the D.APP_PURSE, D.APP_FCI, D.APP_EASY_KEYS, D.APP_ALC, D.APP_DT_LOGs, and D.APP_APPLICATION_DATA through the communication interface of the TOE by sending commands or by listening to the communication between a terminal and the TOE.

3.3.3. Identity Usurpation

T.SID.1 An applet impersonates another application, or even the Java Card RE, in order to gain illegal access to some resources of the card or with respect to the end user or the terminal. See #.SID for details. Directly threatened asset(s): D.SEC_DATA (other assets may be jeopardized should this attack succeed, for instance, if the identity of the JCRE is usurped), D.PIN and D.APP_KEYs.

T.SID.2 The attacker modifies the TOE's attribution of a privileged role (e.g. default applet and currently selected applet), which allows illegal impersonation of this role. See #.SID for further details. Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on whose identity was forged).

3.3.4. Unauthorized Execution

T.EXE-CODE.1 An applet performs an unauthorized execution of a method. See #.EXE-JCS-CODE and #.EXE-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

T.EXE-CODE.2 An applet performs an execution of a method fragment or arbitrary data. See #.EXE-JCSCODE and #.EXE-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

T.EXE-CODE-REMOTE The attacker performs an unauthorized remote execution of a method from the CAD. See #.EXE-APPLI-CODE for details. Directly threatened asset(s): D.APP_CODE.

Application note: This threat concerns version 2.2.x of the Java Card RMI, which allow external users (that is, other than on-card applets) to trigger the execution of code belonging to an on-card applet. On the contrary, T.EXE-CODE.1 is restricted to the applets under the TSF.

T.NATIVE	An applet executes a native method to bypass a TOE Security Function such as the firewall. See #.NATIVE for details. Directly threatened asset(s): D.JCS_DATA.
T.FORGE_TRANS	An attacker tries to force the EasyCard application into a non stable state by retrying a previous transaction, bypassing some code, stopping or disrupting the execution of the application instance in order to succeed an unauthorized transaction.

3.3.5. Denial of Service

T.RESOURCES	An attacker prevents correct operation of the Java Card System through consumption of some resources of the card: RAM or NVRAM. See #.RESOURCES for details. Directly threatened asset(s): D.JCS_DATA.

3.3.6. Card Management

T.DELETION	The attacker deletes an applet or a package already in use on the card, or uses the deletion functions to pave the way for further attacks (putting the TOE in an insecure state). See #.DELETION for details). Directly threatened asset(s): D.SEC_DATA and D.APP_CODE.
T.INSTALL	The attacker fraudulently installs post-issuance of an applet on the card. This concerns either the installation of an unverified applet or an attempt to induce a malfunction in the TOE through the installation process. See #.INSTALL for details. Directly threatened asset(s): D.SEC_DATA (any other asset may be jeopardized should this attack succeed, depending on the virulence of the installed application).
T.DELETION.2	The attacker uses a bug in the card manager implementation to delete applets without authorization.

3.3.7. Services

T.OBJ-DELETION	The attacker keeps a reference to a garbage collected object in order to force the TOE to execute an unavailable method, to make it to crash, or to gain access to a memory containing data that is now being used by another application. See #.OBJ-DELETION for further details. Directly threatened asset(s): D.APP_C_DATA, D.APP_I_DATA and D.APP_KEYs.

3.3.8. Miscellaneous

T.PHYSICAL	The attacker discloses or modifies the design of the TOE, its sensitive data or application code by physical (opposed to logical) tampering means. This threat includes IC failure analysis, electrical probing and unexpected tearing. That also includes the modification of the runtime execution of Java Card System or SCP software through alteration of the intended execution order of (set of) instructions through physical tampering techniques.

This threatens all the identified assets.

This threat refers to the point (7) of the security aspect #.SCP, and all aspects related to confidentiality and integrity of code and data.

High level threats are classified as:

The manipulation of on-card information, including the modification of data, and the malfunction of security mechanism

The disclosure of on-card information

The disclosure of off-card information such as Design and construction data

T.ISOLATION	The attacker uses problems or bugs identified in the native application to access/modify/erase actives from the Java Card System implementation or vice versa. And attacker uses the Mifare legacy application to access/modify/erase without authorization the EasyCard assets.
T.MISUSE	An attacker tries to use the TOE functions to gain access to the Easycard D.APP_PURSE, D.APP_FCI, D.APP_EASY_KEYS, D.APP_ALC, D.APP_DT_LOGs, D.APP_APPLICATION_DATA assets without knowledge of user authentication data or any implicit authorization.
T.LEAKAGE	An attacker may exploit information which is leaked from the TOE during its usage in order to disclose confidential data (User Data or TSF data). The information leakage may be inherent in the normal operation or caused by the attacker. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements.
T.RND	An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.

3.4. Organizational Security Policies

This section describes the organizational security policies to be enforced with respect to the TOE environment.

OSP.VERIFICATION This policy shall ensure the adequacy between the export files used in the verification and those used for installing the verified file. The policy must also ensure that no modification of the file is performed between its verification and the signing by the verification authority.

OSP.MNG_SECRETS Management of secret performed outside the product on behalf of the S.ISSUER shall comply with security organizational policies that enforce integrity and confidentiality of these data.

3.5. Assumptions

This section introduces the assumptions made on the environment of the TOE.

A.APPLET Applets loaded post-issuance do not contain native methods. The Java Card specification explicitly “does not include support for native methods” ([JCVM30], §3.3) outside the API.

A.VERIFICATION All the bytecodes are verified at least once, before the loading, before the installation or before the execution, in order to ensure each bytecode is valid at execution time.

4. Security Objectives

4.1. Security Objectives for the TOE

This section introduces the objectives for TOE.

4.1.1. Identification

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

4.1.2. Execution

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

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

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

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

O.NATIVE The only means that the Java Card VM shall provide for an application to execute native code is the invocation of a method of the Java Card API, or any additional API. See #.NATIVE for details.

O.OPERATE The TOE must ensure continued correct operation of its security functions. See #.OPERATE for details.

O.REALLOCATION The TOE shall ensure that the re-allocation of a memory block for the runtime areas of the Java Card VM does not disclose any information that was previously stored in that block.

O.RESOURCES The TOE shall control the availability of resources for the applications. See #.RESOURCES for details.

4.1.3. Services

O.ALARM The TOE shall provide appropriate feedback information upon detection of a potential security violation. See #.ALARM for details.

O.CIPHER The TOE shall provide a means to cipher sensitive data for applications in a secure way. In particular, the TOE must support cryptographic algorithms consistent with cryptographic usage policies and standards. See #.CIPHER for details.

O.KEY-MNGT The TOE shall provide a means to securely manage cryptographic keys. This concerns the correct generation, distribution, access and destruction of cryptographic keys. See #.KEYMNGT.

O.PIN-MNGT The TOE shall provide a means to securely manage PIN objects. See #.PIN-MNGT for details.

Application note: PIN objects may play key roles in the security architecture of client applications. The way they are stored and managed in the memory of the smart card must be carefully considered, and this applies to the whole object rather than the sole value of the PIN. For instance, the try counter's value is as sensitive as that of the PIN.

O.REMOTE	The TOE shall provide restricted remote access from the CAD to the services implemented by the applets on the card. This particularly concerns the Java Card RMI services introduced in version 2.2.x of the Java Card platform.
O.TRANSACTION	The TOE must provide a means to execute a set of operations atomically. See #.TRANSACTION for details.

4.1.4. Object Deletion

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

4.1.5. Applet Management

O.DELETION	The TOE shall ensure that both applet and package deletion perform as expected. See #.DELETION for details.
O.LOAD	The TOE shall ensure that the loading of a package into the card is safe.

Application note: Usurpation of identity resulting from a malicious installation of an applet on the card may also be the result of perturbing the communication channel linking the CAD and the card. Even if the CAD is placed in a secure environment, the attacker may try to capture, duplicate, permute or modify the packages sent to the card. He may also try to send one of its own applications as if it came from the card issuer. Thus, this objective is intended to ensure the integrity and authenticity of loaded CAP files.

O.INSTALL	The TOE shall ensure that the installation of an applet performs as expected (See #.INSTALL for details).

4.1.6. Card Manager

O.CARD-MANAGEMENT	The card manager shall control the access to card management functions such as the installation, update or deletion of applets. It shall also implement the card issuer's policy on the card. The card manager is an application with specific rights, which is responsible for the administration of the smart card. This component will in practice be tightly connected with the TOE, which in turn shall very likely rely on the card manager for the effective enforcing of some of its security functions. Typically the card manager shall be in charge of the life cycle of the whole card, as well as that of the installed applications (applets). The card manager should prevent that card content management (loading, installation, deletion) is carried out, for instance, at invalid

states of the card or by non-authorized actors. It shall also enforce security policies established by the card issuer.

4.1.7. Security Platform

O.SCP.IC The SCP shall provide all IC security features against physical attacks. This security objective for the environment refers to the point (7) of the security aspect #.SCP: o It is required that the IC is designed in accordance with a well-defined set of policies and Standards (likely specified in another protection profile), and will be tamper resistant to actually prevent an attacker from extracting or altering security data (like cryptographic keys) by using commonly employed techniques (physical probing and sophisticated analysis of the chip). This especially matters to the management (storage and operation) of cryptographic keys.

O.SCP.RECOVERY If there is a loss of power, or if the smart card is withdrawn from the CAD while an operation is in progress, the SCP must allow the TOE to eventually complete the interrupted operation successfully, or recover to a consistent and secure state. This security objective for the environment refers to the security aspect #.SCP(1): The smart card platform must be secure with respect to the SFRs. Then after a power loss or sudden card removal prior to completion of some communication protocol, the SCP will allow the TOE on the next power up to either complete the interrupted operation or revert to a secure state.

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

1) It does not allow the TSFs to be bypassed or altered and does not allow access to other low-level functions than those made available by the packages of the API. That includes the protection of its private data and code (against disclosure or modification) from the Java Card System and native application.
2) It provides secure low-level cryptographic processing to the Java Card System and the native application.
3) It supports the needs for any update to a single persistent object or class field to be atomic, and possibly a low-level transaction mechanism.
4) And testing of the TSF.

It allows the Java Card System to store data in “persistent technology memory” or in volatile memory, depending on its needs (for instance, transient objects must not be stored in non-volatile memory). The memory model is structured and allows for low-level control accesses (segmentation fault detection).

O.RND	The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have a sufficient entropy. The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

4.1.8. Additional

O.ISOLATION	The TOE shall provide mechanisms based on access control that isolate Java Card System from other native applications, being impossible the interaction between them. Also, the TOE shall provide mechanisms based on access control, other than avoid the usage of CPU EasyCard assets through Mifare interface.
O.CRYPTO	The TOE shall provide the cryptographic computation services to native applications in usage phase for several modes of AES algorithm.
O.APP_INTEG	The TOE shall provide mechanisms for detecting integrity errors in stored user data D.APP_PURSE, D.APP_FCI, D.APP_EASY_KEYS, D.APP_ALC, D.APP_DT_LOGs and D.APP_APPLICATION_DATA.
O.LEAKAGE	The TOE must provide protection against disclosure of confidential data (User Data or TSF data) stored and/or processed in the TOE.
O.APP_ACL	In the usage phase, the TOE shall provide configurable access control system to prevent unauthorized access to the EasyCard user data and TSF data, and the TOE shall provide its users with the means of controlling and limiting access to the objects and resources they own or are responsible for in a configurable and deterministic manner.
O.APP_MNG	The TOE shall provide the following EasyCard application management services:

- The TOE shall provide the secure mechanism for the management of the application life cycle,
- The TOE shall provide the secure mechanism for the management of the secret keys.

O.TRANS During the transaction, the TOE shall provide the following services for interacting with EasyCard application:

- The TOE shall enforce mutual authentication with external devices during any transaction,
- The TOE shall count and limit the transactions,
- The TOE shall record the last transactions to support effective security management,
- The TOE shall sign the transaction and give the signature to the external devices to proof the success of the transaction,
- The TOE shall detect and reject replayed transactions,
- The TOE shall enforce each type transaction uses the suited secret key,
- The TOE shall ensure the continued correct operation of abnormal process of transactions such as interruption during transactions

4.2. Security Objectives for the Environment

This section introduces the security objectives to be achieved by the environment.

OE.APPLET No applet loaded post-issuance shall contain native methods.

OE.VERIFICATION All the bytecodes shall be verified at least once, before the loading, before the installation or before the execution, depending on the card capabilities, in order to ensure that each bytecode is valid at execution time. See #.VERIFICATION for details.

OE.MNG_SECRETS The secret User or TSF data managed outside the TOE shall be protected against unauthorized disclosure and modification.

4.3. Security Objectives Mapping and Coverage Rationale

4.3.1. Threats

Threats	***Security Objectives***
T.CONFID-APPLI-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARDMANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.ALARM, O.TRANSACTION, O.CIPHER, O.PIN-MNGT, O.KEYMNGT, O.REALLOCATION
T.CONFID-JCS-CODE	OE.VERIFICATION, O.CARD-MANAGEMENT, O.NATIVE
T.CONFID-JCS-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARDMANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.ALARM
T.INTEG-APPLI-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE
T.INTEG-APPLI-CODE.LOAD	O.LOAD, O.CARD-MANAGEMENT
T.INTEG-APPLI-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARDMANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.GLOBAL_ARRAYS_INTEG, O.ALARM, O.TRANSACTION, O.CIPHER, O.PIN-MNGT, O.KEYMNGT, O.REALLOCATION
T.INTEG-APPLI-DATA.LOAD	O.LOAD, O.CARD-MANAGEMENT
T.INTEG-JCS-CODE	O.CARD-MANAGEMENT, OE.VERIFICATION, O.NATIVE
T.INTEG-JCS-DATA	O.SCP.RECOVERY, O.SCP.SUPPORT, O.CARDMANAGEMENT, OE.VERIFICATION, O.SID, O.OPERATE, O.FIREWALL, O.ALARM
T.SID.1	O.CARD-MANAGEMENT, O.FIREWALL, O.GLOBAL_ARRAYS_CONFID, O.GLOBAL_ARRAYS_INTEG, O.INSTALL, O.SID
T.SID.2	O.SCP.RECOVERY, O.SCP.SUPPORT, O.SID, O.OPERATE, O.FIREWALL, O.INSTALL
T.EXE-CODE.1	OE.VERIFICATION, O.FIREWALL
T.EXE-CODE.2	OE.VERIFICATION
T.EXE-CODE-REMOTE	O.REMOTE

T.NATIVE	OE.VERIFICATION, OE.APPLET, O.NATIVE
T.RESOURCES	O.INSTALL, O.OPERATE, O.RESOURCES, O.SCP.RECOVERY, O.SCP.SUPPORT
T.DELETION, T.DELETION.2	O.DELETION, O.CARD-MANAGEMENT
T.INSTALL	O.INSTALL, O.LOAD, O.CARD-MANAGEMENT
T.OBJ-DELETION	O.OBJ-DELETION
T.PHYSICAL	O.SCP.IC
T.ISOLATION	O.ISOLATION
T.CONFID-EASY-DATA	O.CRYPTO, O.APP_ACL, OE.MNG_SECRETS, O.APP_MNG
T.INTEG-EASY-DATA	O.APP_INTEG
T. FORGE_TRANS	O.TRANS
T.MISUSE	O.CRYPTO, O.APP_ACL, O.TRANS, O.APP_MNG
T.LEAKAGE	O.LEAKAGE
T.RND	O.RND

Confidentiality

T.CONFID-APPLI-DATA

Coverage rationale provided in the section 6.3.1.1 of the [JCS-OP-PP].

T.CONFID-JCS-CODE

Coverage rationale provided in the section 6.3.1.1 of the [JCS-OP-PP].

T.CONFID-JCS-DATA

Coverage rationale provided in the section 6.3.1.1 of the [JCS-OP-PP].

Remark: the PP marked the O.CARD-MANAGEMENT, O.SCP.IC, OE.SCP.RECOVERY, OE.SCP.SUPPORT as Objective for the environment, but the current ST these objectives have moved to objectives for the TOE because the Card manager and the SCP are inside the scope of the TOE. Also the A.DELETION has moved to T.DELETION.2. Then the rationale should be understood taking into consideration these changes.

T.CONFID-EASY-DATA

O.APP_ACL (supported by O.TRANS, O.APP_MNG, O.CRYPTO) implies that data in the TOE can only be read, written or modified according to the access rules as defined in the access control policy EasyCard Access Control Policy SFP, which was

defined in O.APP_ACL. The support by O.TRANS and O.APP_MNG is needed since several rules of EasyCard Access Control Policy SFP access rules restrict the access to Purse data, the authenticity of which is made sure by services required by O.TRANS and O.APP_MNG. The support by O.CRYPTO is needed since several services required by O.TRANS and O.APP_MNG rely on cryptographic mechanisms required by O.CRYPTO.

OE.MNG_SECRETS imply that authorized persons, who are allowed to read, write or modify data in the card, use these rights only in an environment, where unauthorized access to these data is prevented by the environment.

Integrity

T.INTEG-APPLI-CODE

Coverage rationale provided in the section 6.3.1.2 of the [JCS-OP-PP].

T.INTEG-APPLI-CODE.LOAD

Coverage rationale provided in the section 6.3.1.2 of the [JCS-OP-PP].

T.INTEG-APPLI-DATA

Coverage rationale provided in the section 6.3.1.2 of the [JCS-OP-PP].

T.INTEG-APPLI-DATA.LOAD

Coverage rationale provided in the section 6.3.1.2 of the [JCS-OP-PP].

T.INTEG-JCS-CODE

Coverage rationale provided in the section 6.3.1.2 of the [JCS-OP-PP].

T.INTEG-JCS-DATA

Coverage rationale provided in the section 6.3.1.2 of the [JCS-OP-PP].

Remark: the PP marked the O.CARD-MANAGEMENT, O.SCP.IC, OE.SCP.RECOVERY, OE.SCP.SUPPORT as Objective for the environment, but the current ST these objectives have moved to objectives for the TOE because the Card manager and the SCP are inside the scope of the TOE. Also the A.DELETION has moved to T.DELETION.2. Then the rationale should be understood taking into consideration these changes.

T.INTEG-EASY-DATA

Covered by O.APP_INTEG, which requires that requires that the sensitive data must be integrated stored in the TOE.

Identity Usurpation

T.SID.1

Coverage rationale provided in the section 6.3.1.3 of the [JCS-OP-PP].

T.SID.2

Coverage rationale provided in the section 6.3.1.3 of the [JCS-OP-PP].

Remark: the PP marked the O.CARD-MANAGEMENT, O.SCP.IC, OE.SCP.RECOVERY, OE.SCP.SUPPORT as Objective for the environment, but the current ST these objectives have moved to objectives for the TOE because the Card manager and the SCP are inside the scope of the TOE. Also the A.DELETION has moved to T.DELETION.2. Then the rationale should be understood taking into consideration these changes.

Unauthorized Execution

T.EXE-CODE.1

Coverage rationale provided in the section 6.3.1.4 of the [JCS-OP-PP].

T.EXE-CODE.2

Coverage rationale provided in the section 6.3.1.4 of the [JCS-OP-PP].

T.EXE-CODE-REMOTE

Coverage rationale provided in the section 6.3.1.4 of the [JCS-OP-PP].

T. FORGE_TRANS

Transactions are adverted directly by the security objective O.TRANS preventing the unauthorized transactions.

Denial of Service

T.RESOURCES

Coverage rationale provided in the section 6.3.1.5 of the [JCS-OP-PP].

Remark: the PP marked the O.CARD-MANAGEMENT, O.SCP.IC, OE.SCP.RECOVERY, OE.SCP.SUPPORT as Objective for the environment, but the current ST these objectives have moved to objectives for the TOE because the Card manager and the SCP are inside the scope of the TOE. Also the A.DELETION has moved to T.DELETION.2. Then the rationale should be understood taking into consideration these changes.

Card Management

T.DELETION

Coverage rationale provided in the section 6.3.1.6 of the [JCS-OP-PP].

T.INSTALL

Coverage rationale provided in the section 6.3.1.6 of the [JCS-OP-PP].

T.DELETION.2

The threat T.DELETION.2 is covered by the O.CARD-MANAGEMENT which controls the access to card management functions such as deletion of applets.

Remark: the PP marked the O.CARD-MANAGEMENT, O.SCP.IC, OE.SCP.RECOVERY, OE.SCP.SUPPORT as Objective for the environment, but the current ST these objectives have moved to objectives for the TOE because the Card manager and the SCP are inside the scope of the TOE. Also the A.DELETION has moved to T.DELETION.2. Then the rationale should be understood taking into consideration these changes.

Services

T.OBJ-DELETION

Coverage rationale provided in the section 6.3.1.7 of the [JCS-OP-PP].

Miscellaneous

T.PHYSICAL

Coverage rationale provided in the section 6.3.1.8 of the [JCS-OP-PP].

T.ISOLATION

The TOE provides internal mechanisms that avoid the interaction between the Java Card System and the native applications residents in the TOE. And also avoid the interaction of CPU EasyCard assets when the Mifare legacy application is used. The coverage is directly between the objective and the threats.

T,RND

Directly coverage with the O.RND

T.MISUSE

O.APP_ACL (supported by O.TRANS, O.APP_MNG, O.CRYPTO) implies that data in the TOE can only be read, written or modified according to the access rules as defined in the access control policy EasyCard Access Control Policy SFP, which was defined in O.APP_ACL. The support by O.TRANS and O.APP_MNG is needed since several rules of EasyCard Access Control Policy SFP access rules restrict the access to Purse data, the authenticity of which is made sure by services required by O.TRANS and O.APP_MNG. The support by O.CRYPTO is needed since several services required by O.TRANS and O.APP_MNG rely on cryptographic mechanisms required by O.CRYPTO.

T.LEAKAGE

The threat T.LEAKAGE is adverted directly by the security objective O.LEAKAGE addressing the protection against disclosure of confidential data (User Data or TSF data) stored and/or processed in the TOE by attacks including but not limited to use of side channels, fault injection or physical manipulation.

4.3.2. Organizational Security Policies

OSP	*Objectives*
OSP.VERIFICATION	OE.VERIFICATION
OSP.MNG_SECRETS	OE.MNG_SECRETS

OSP.VERIFICATION

Coverage rationale provided in the section 6.3.2 of the [JCS-OP-PP].

OSP.MNG_SECRET

Covered directly by the OE.MNG_SECRETS

4.3.3. Assumptions

Assumptions	*Objectives*
A.APPLET	OE.APPLET
A.VERIFICATION	OE.VERIFICATION

A.APPLET

Coverage rationale provided in the section 6.3.3 of the [JCS-OP-PP].

A.VERIFICATION

Coverage rationale provided in the section 6.3.1.3 of the [JCS-OP-PP].

5. Extended Components Definition

5.1. Definition of the Family FCS_RND

To define the IT security functional requirements of the TOE a sensitive family (FCS_RND) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes. The component FCS_RND is not limited to generation of cryptographic keys unlike the component FCS_CKM.1.

The similar component FIA_SOS.2 is intended for non-cryptographic use.

The family “Generation of random numbers (FCS_RND)” is specified as follows.

5.1.1. Generation of Random Numbers (FCS_RND)

Family behaviour:

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

Component levelling:

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

Management: FCS_RND.1

There are no management activities foreseen.

Audit: FCS_RND.1

There are no actions defined to be auditable.

FCS_RND.1 Quality metric for random numbers

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RND.1.1 **The TSF shall provide a mechanism to generate random numbers that meet [assignment: *a defined quality metric*].**

5.2. Definition of the Family FPT_EMSEC

The sensitive family FPT_EMSEC (TOE Emanation) of the Class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE. The TOE shall prevent attacks against the TOE and other secret data where the attack is based on external observable physical phenomena of the TOE. Examples of such attacks are evaluation of TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc. This family describes the functional requirements for the limitation of intelligible emanations which are not directly addressed by any other component of CC part 2.

The family "TOE Emanation (FPT_EMSEC)" is specified as follows

5.2.1. TOE Emanation (FPT_EMSEC)

Family behaviour:

This family defines requirements to mitigate intelligible emanations.

Component levelling:

FPT_EMSEC.1 TOE emanation has two constituents:

1) FPT_EMSEC.1.1 Limit of Emissions requires to not emit intelligible emissions enabling access to TSF data or user data.
2) FPT_EMSEC.1.2 Interface Emanation requires to not emit interface emanation enabling access to TSF data or user data.

Management: FPT_EMSEC.1

There are no management activities foreseen.

Audit: FPT_EMSEC.1

There are no actions defined to be auditable

FPT_EMSEC.1 TOE Emanation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMSEC.1.1 The TOE shall not emit [assignment: types of emissions] in excess of [assignment: specified limits] enabling access to

	[assignment: *list of types of TSF data*] and [assignment: list of types of user data].
FPT_EMSEC.1.2	**The TSF shall ensure [assignment: *type of users*] are unable to use the following interface [assignment: *type of connection*] to gain access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].**

6. Security Functional Requirements

The operations over the security requirements are marked as:

- Assignment, Between brackets and bold e.g: **[Bold]**
- Selection, between brackets and with the word selection at the beginning e.g: [selection: selection-done]
- Iteration, SFRid/Iteration name
- Refinement with the word refinement at the beginning and italics: e.g: *refined content*

Remark: The Application notes defined in the SFR of the [JCS-OP-PP] apply to the current security target.

6.1. Java Card Platform

This section states the security functional requirements for the Java Card System. All the groups defined in the table below.

Group	***Description***
Core with Logical Channels(CoreG_LC)	The CoreG_LC contains the requirements concerning the runtime environment of the Java Card System implementing logical channels. This includes the firewall policy and the requirements related to the Java Card API. Logical channels are a Java Card specification version 2.2 feature. This group is the union of requirements from the Core (CoreG) and the Logical channels (LCG) groups defined in [PP/0305] (cf. Java Card System Protection Profile Collection [PP JCS]).
Remote Method Invocation (RMI)	The RMIG contains the security requirements for the remote method invocation feature, which provides a new protocol of communication between the terminal and the applets. This was introduced in Java Card specification version 2.2.
Installation	The InstG contains the security requirements concerning the

(InstG)	installation of post-issuance applications. It does not address card management issues in the broad sense, but only those security aspects of the installation procedure that are related to applet execution.
Applet deletion (ADELG)	The ADELG contains the security requirements for erasing installed applets from the card, a feature introduced in Java Card specification version 2.2.
Object deletion (ODELG)	The ODELG contains the security requirements for the object deletion capability. This provides a safe memory recovering mechanism. This is a Java Card specification version 2.2 feature.
Secure carrier (CarG)	The CarG group contains minimal requirements for secure downloading of applications on the card. This group contains the security requirements for preventing, in those configurations that do not support on-card static or dynamic bytecode verification, the installation of a package that has not been bytecode verified, or that has been modified after bytecode verification.

Subjects are active components of the TOE that (essentially) act on the behalf of users. The users of the TOE include people or institutions (like the applet developer, the card issuer, the verification authority), hardware (like the CAD where the card is inserted or the PCD) and software components (like the application packages installed on the card). Some of the users may just be aliases for other users. Subjects (prefixed with an “S”) are described in the following table:

Subject	***Description***
S.ADEL	The applet deletion manager is unique and involved in the ADEL security policy.
S.APPLET	Any applet instance.
S.CAD	The CAD represents the actor that requests, by issuing commands to the card, for RMI services. It also plays the role of the off-card entity that communicates with the S.INSTALLER.
S.INSTALLER	The installer is the on-card entity which acts on behalf of the card issuer. This subject is involved in the loading of packages and installation of applets.
S.JCRE	The runtime environment under which Java programs in a smart card are executed.
S.JCVM	The bytecode interpreter that enforces the firewall at runtime.
S.LOCAL	Operand stack of a JCVM frame, or local variable of a JCVM frame containing an object or an array of references.
S.MEMBER	Any object’s field, static field or array position.

S.PACKAGE	A package is a namespace within the Java programming language that may contain classes and interfaces, and in the context of Java Card technology, it defines either a user library, or one or several applets.

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

Object	***Description***
O.APPLET	Any installed applet, its code and data.
O.CODE_PKG	The code of a package, including all linking information. On the Java Card platform, a package is the installation unit.
O.JAVAOBJECT	Java class instance or array. It should be noticed that KEYS, PIN, arrays and applet instances are specific objects in the Java programming language.
O.REMOTE_MTHD	A method of a remote interface.
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]).
O.RMI_SERVICE	These are instances of the class javacardx.rmi.RMIService. They are the objects that actually process the RMI services.
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.

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

Information	***Description***
I.APDU	Any APDU sent to or from the card through the communication channel.
I.DATA	JCVM Reference Data: object ref addresses of APDU buffer, JCRE-owned instances of APDU class and byte array for install method.
I.RORD	Remote object reference descriptors which provide 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.

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

Security attribute	***attribute Description/Value***
Active Applets	The set of the active applets' AIDs. An active applet is an applet that is selected on at least one of the logical channels.
Applet Selection Status	"Selected" or "Deselected".
Applet's version number	The version number of an applet (package) indicated in the export file.
Class	Identifies the implementation class of the remote object.
Context	Package AID or "Java Card RE".
Currently Active Context	Allows the retrieval of the Package AID and Applet's version number ([JCVM22], §4.5.2).
Dependent package AID	Allows the retrieval of the Package AID and Applet's version number ([JCVM22], §4.5.2).
ExportedInfo	ExportedInfo Boolean (indicates whether the remote object is exportable or not).
Identifier	The Identifier of a remote object or method is a number that uniquely identifies the remote object or method, respectively.
LC Selection Status	Multiselectable, Non-multiselectable or "None".
Life Time	CLEAR_ON_DESELECT or PERSISTENT (*).
Owner	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). The owner of a remote object is the applet instance that created the object.
Package AID	The AID of each package indicated in the export file.
Registered Applets	The set of AID of the applet instances registered on the card.
Remote	An object is Remote if it is an instance of a class that directly or indirectly implements the interface java.rmi.Remote.
Resident Packages	The set of AIDs of the packages already loaded on the card.

Returned References	The set of remote object references that have been sent to the CAD during the applet selection session. This attribute is implementation dependent.
Selected Applet Context	Package AID or “None”.
Sharing	Standards, SIO, Java Card RE entry point or global array.
Static References	Static fields of a package may contain references to objects. The Static References attribute records those references.

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

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

Operation	***Description***
OP.ARRAY_ACCESS(O.JAVAOBJECT field)	Read/Write an array component.
OP.CREATE(Sharing, LifeTime) (*)	Creation of an object (new or makeTransient call).
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 applets, either logically or physically.
OP.GET_ROR(O.APPLET,...)	Retrieves the initial remote object reference of a RMI based applet. This reference is the seed which the CAD client application needs to begin remote method invocations.
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.INVOKE(O.RMI_SERVICE,...)	Requests a remote method invocation on the remote object.
OP.JAVA(...)	Any access in the sense of [JCRE22], §6.2.8. It stands for one of the operations OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW, OP.TYPE_ACCESS.
OP.PUT(S1,S2,I)	Transfer a piece of information I from S1 to S2.
OP.RET_RORD(S.JCRE,S.CAD,I.RORD)	Send a remote object reference descriptor to the CAD.
OP.THROW(O.JAVAOBJECT)	Throwing of an object (athrow, see [JCRE22], §6.2.8.7).
OP.TYPE_ACCESS(O.JAVAOBJECT, class)	Invoke checkcast or instanceof on an object in order to access to classes (standard or shareable interfaces objects).

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

6.2. COREG_LC Security Functional Requirements

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

6.2.1. FIREWALL Policy

FDP_ACC.2/FIREWALL Complete Access Control

FDP_ACC.2.1/FIREWALL

The TSF shall enforce the **[FIREWALL access control SFP]** on **[S.PACKAGE, S.JCRE, S.JCVM, O.JAVAOBJECT]** and all operations among subjects and objects covered by the SFP.

Refinement:

The operations involved in the policy are:

- *OP.CREATE,*
- *OP.INVK_INTERFACE,*
- *OP.INVK_VIRTUAL,*
- *OP.JAVA,*

- *OP.THROW,*
- *OT.TYPE_ACCESS.*

FDP_ACC.2.2/FIREWALL

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

FDP_ACF.1/FIREWALL Security Attribute based Access Control

FDP_ACF.1.1/FIREWALL

The TSF shall enforce the **[FIREWALL access control SFP]** to objects based on the following: **[**

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

]

FDP_ACF.1.2/FIREWALL

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

- **R.JAVA.1 ([JCRE22], §6.2.8): S.PACKAGE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW or OP.TYPE_ACCESS upon any O.JAVAOBJECT whose Sharing attribute has value "JCRE entry point" or "global array".**
- **R.JAVA.2 ([JCRE22], §6.2.8): S.PACKAGE may freely perform OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE or OP.THROW upon any O.JAVAOBJECT whose Sharing attribute has value "Standard" and whose Lifetime attribute has value "PERSISTENT" only if O.JAVAOBJECT's Context attribute has the same value as the active context.**
- **R.JAVA.3 ([JCRE22], §6.2.8.10): S.PACKAGE may perform OP.TYPE_ACCESS upon an O.JAVAOBJECT whose Sharing attribute has value "SIO" only if O.JAVAOBJECT is being cast into (checkcast) or is being verified as being an instance of (instanceof) an interface that extends the Shareable interface.**

- **R.JAVA.4 ([JCRE22], §6.2.8.6): S.PACKAGE may perform OP.INVK_INTERFACE upon an O.JAVAOBJECT whose Sharing attribute has the value "SIO", and whose Context attribute has the value "Package AID", only if the invoked interface method extends the Shareable interface and one of the following conditions applies:**
  - **The value of the attribute Selection Status of the package whose AID is "Package AID" is "Multiselectable",**
  - **The value of the attribute Selection Status of the package whose AID is "Package AID" is "Non-multiselectable", and either "Package AID" is the value of the currently selected applet or otherwise "Package AID" does not occur in the attribute Active Applets.**
- **R.JAVA.5: S.PACKAGE may perform OP.CREATE only if the value of the Sharing parameter is "Standard".]**

FDP_ACF.1.3/FIREWALL

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[**

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

]

FDP_ACF.1.4/FIREWALL

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[**

- **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.**
- **Any subject attempting to create an object by the means of OP.CREATE and a "CLEAR_ON_DESELECT" LifeTime parameter if the active context is not the same as the Selected Applet Context.**

]

FDP_IFC.1/JCVM Subset Information Flow Control

FDP_IFC.1.1/JCVM

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

FDP_IFF.1/JCVM Simple Security Attributes

FDP_IFF.1.1/JCVM

The TSF shall enforce the **[JCVM information flow control SFP]** based on the following types of subject and information security attributes: **[**

Subjects	*Security attributes*
S.JCVM	**Currently Active Context**

]

FDP_IFF.1.2/JCVM

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[**

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

]

FDP_IFF.1.3/JCVM

The TSF shall enforce the **[no additional information flow control SFP rules]**.

FDP_IFF.1.4/JCVM

The TSF shall explicitly authorise an information flow based on the following rules: **[no rules, based on security attributes, that explicitly authorise information flows]**.

FDP_IFF.1.5/JCVM

The TSF shall explicitly deny an information flow based on the following rules: **[no rules, based on security attributes that explicitly deny information flows]**.

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]** to restrict the ability to [selection: modify] the security attributes **[Selected Applet Context to the Java Card RE]**.

FMT_MSA.1/JCVM Management of Security Attributes

FMT_MSA.1.1/JCVM

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

FMT_MSA.2/FIREWALL_JCVM Secure Security Attributes

FMT_MSA.2.1/FIREWALL_JCVM

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

FMT_MSA.3/FIREWALL Static Attribute Initialization

FMT_MSA.3.1/FIREWALL

The TSF shall enforce the **[FIREWALL access control SFP]** to provide [selection: restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/FIREWALL

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

FMT_MSA.3/JCVM Static Attribute Initialization

FMT_MSA.3.1/JCVM

The TSF shall enforce the **[JCVM information flow control SFP]** to provide [selection: restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/JCVM [Editorially Refined]

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

FMT_SMF.1/JC Specification of Management Functions

FMT_SMF.1.1/JC

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

FMT_SMR.1/JC Security Roles

FMT_SMR.1.1/JC

The TSF shall maintain the roles: **[**

- **Java Card RE (JCRE),**
- **Java Card VM (JCVM).**

]

FMT_SMR.1.2/JC

The TSF shall be able to associate users with roles.

6.2.2. Application Programming Interface

The following SFRs are related to the Java Card API. The whole set of cryptographic algorithms is generally not implemented because of limited memory resources and/or limitations due to exportation. Therefore, the following requirements only apply to the implemented subset. It should be noticed that the execution of the additional native code is not within the TSF. Nevertheless, access to API native methods from the Java Card System is controlled by TSF because there is no difference between native and interpreted methods in their interface or invocation mechanism.

FCS_CKM.1/API_DES Cryptographic Key Generation

FCS_CKM.1.1/API_DES

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[class KeyBuilder cryptographic key generation algorithm]** and specified cryptographic key sizes **[16 / 24 (2x56 bits / 3x56 bits)]** that meet the following: **[JCAPI30]**.

FCS_CKM.1/API_RSA Cryptographic Key Generation

FCS_CKM.1.1/API_RSA

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[classes KeyBuilder and KeyPair cryptographic key generation algorithm]** and specified cryptographic key sizes **[1024~2048 bits]** that meet the following: **[JCAPI30]**.

FCS_CKM.2/API Cryptographic Key Distribution

FCS_CKM.2.1/API

The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[setKey method]** that meets the following: **[JCAPI30]**.

FCS_CKM.3/API Cryptographic Key Access

FCS_CKM.3.1/API

The TSF shall perform **[DES RSA key access]** in accordance with a specified cryptographic key access method **[SetKey、getKey method]** that meets the following: **[JCAPI30]**.

FCS_CKM.4/API Cryptographic Key Destruction

FCS_CKM.4.1/API

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[clearKey method]** that meets the following: **[JCAPI30]**.

FCS_COP.1/API_DES Cryptographic Operation

FCS_COP.1.1/API_DES

The TSF shall perform **[encryption, decryption, signature generation, and signature verification]** in accordance with a specified cryptographic algorithm

[DES] and cryptographic key sizes **[16 / 24 (2x56 bits / 3x56 bits)]** that meet the following: **[JCAPI30]**.

FCS_COP.1/API_RSA Cryptographic Operation

FCS_COP.1.1/API_RSA

The TSF shall perform **[encryption, decryption, signature generation, and signature verification]** in accordance with a specified cryptographic algorithm **[RSA]** and cryptographic key sizes **[1024~2048 bits, SHA-1]** that meet the following: **[JCAPI30]**.

FDP_RIP.1/ABORT Subset Residual Information Protection

FDP_RIP.1.1/ABORT

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

FDP_RIP.1/APDU Subset Residual Information Protection

FDP_RIP.1.1/APDU

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

FDP_RIP.1/bArray Subset Residual Information Protection

FDP_RIP.1.1/bArray

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

FDP_RIP.1/KEYS Subset Residual Information Protection

FDP_RIP.1.1/KEYS

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

FDP_RIP.1/TRANSIENT Subset Residual Information Protection

FDP_RIP.1.1/TRANSIENT

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

FDP_ROL.1/FIREWALL Basic Rollback

FDP_ROL.1.1/FIREWALL

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

FDP_ROL.1.2/FIREWALL

The TSF shall permit operations to be rolled back within the **[scope of a select(), deselect(), process(), install() or uninstall() call, notwithstanding the restrictions given in [JCRE30], §7.7, within the bounds of the Commit Capacity ([JCRE30], §7.8), and those described in [JCAPI30]].**

6.2.3. Card Security Management

FAU_ARP.1 Security Alarms

FAU_ARP.1.1

The TSF shall take one of the following actions: **[**

- **throw an exception,**
- **lock the card session,**
- **reinitialize the Java Card System and its data,**
- **[none]**

] upon detection of a potential security violation.

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

- *CAP file inconsistency,*
- *typing error in the operands of a bytecode,*
- *applet life cycle inconsistency,*
- *card tearing (unexpected removal of the Card out of the CAD) and power failure,*
- *abort of a transaction in an unexpected context, (see abortTransaction(), [JCAPI30] and ([JCRE30], §7.6.2)*
- *violation of the Firewall or JCVM SFPs,*
- *unavailability of resources,*

- *array overflow,*
- I/O exception of some sort,
- Exceptional arithmetic condition occurs,
- Array is accessed with an illegal index,
- Store wrong type of object into an array of objects,
- Cast an object to a subclass which it is not an instance of,
- Index of some sort is out of range,
- Create an array with negative size,
- Use null where an object is required,
- Security violation,
- Communication-related exception occurs during executing a remote method call,
- APDU-related exception occurs,
- OwnerPIN class access related exception occurs,
- JCSystem class related exception occurs,
- Transaction subsystem related exception occurs,
- User exception occurs,
- Service framework related exception occurs,
- Cryptography related exception occurs.

FDP_SDI.2/API Stored Data Integrity Monitoring and Action

FDP_SDI.2.1/API

The TSF shall monitor user data stored in containers controlled by the TSF for **[integrity errors]** on all objects, based on the following attributes: **[cryptographic keys, PIN values and their associated security attributes]**.

FDP_SDI.2.2/API

Upon detection of a data integrity error, the TSF shall **[throw exception or give a warning]**.

FPR_UNO.1 Unobservability

FPR_UNO.1.1

The TSF shall ensure that **[any user]** are unable to observe the operation **[verified]** on **[PIN]** by **[protected users]**.

FPT_FLS.1/CM Failure with Preservation of Secure State

FPT_FLS.1.1/CM

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]**.

FPT_TDC.1 Inter-TSF Basic TSF Data Consistency

FPT_TDC.1.1

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

FPT_TDC.1.2

The TSF shall use **[**

- **the rules defined in [JCVM30] specification,**
- **the API tokens defined in the export files of reference implementation**

] when interpreting the TSF data from another trusted IT product.

6.2.4. AID Management

FIA_ATD.1/AID User Attribute Definition

FIA_ATD.1.1/AID

The TSF shall maintain the following list of security attributes belonging to individual users: **[**

- **Package AID,**
- **Applet's version number,**
- **Registered applet AID,**
- **Applet Selection Status ([JCVM22], §6.5).**

]

Refinement:

"Individual users" stand for applets.

FIA_UID.2/AID User Identification before Any Action

FIA_UID.2.1/AID

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

FIA_USB.1/AID User-Subject Binding

FIA_USB.1.1/AID

The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: **[Package AID]**.

FIA_USB.1.2/AID

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: **[Context shall hold the user security attribute package AID]**.

FIA_USB.1.3/AID

The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: **[Context shall hold the user security attribute package AID]**.

FMT_MTD.1/JCRE Management of TSF Data

FMT_MTD.1.1/JCRE

The TSF shall restrict the ability to [selection: modify] the **[list of registered applets' AIDs]** to **[the JCRE].**

FMT_MTD.3/JCRE Secure TSF Data

FMT_MTD.3.1/JCRE

The TSF shall ensure that only secure values are accepted for **[the registered applets' AIDs]**.

6.3. INSTG Security Functional Requirements

This group consists of the SFRs related to the installation of the applets, which addresses security aspects outside the runtime. The installation of applets is a critical phase, which lies partially out of the boundaries of the firewall, and therefore requires specific treatment.

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 rules when importing user data controlled under the SFP from outside the TOE: **[**
Package loading is allowed only if, for each dependent package, its AID attribute is equal to a resident package AID attribute, the major (minor) Version attribute associated to the dependent package is lesser than or equal to the major (minor) Version attribute associated to the resident package ([JCVM22], §4.5.2).]

FMT_SMR.1/Installer Security Roles

FMT_SMR.1.1/Installer

The TSF shall maintain the roles**: [Installer]**.

FMT_SMR.1.2/Installer

The TSF shall be able to associate users with roles.

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 [JCRE30] §11.1.4]**.

FPT_RCV.3/Installer Automated Recovery without Undue Loss

FPT_RCV.3.1/Installer

When automated recovery from **[none]** is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2/Installer

For **[package loaded and deleted are failed]**, the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3/Installer

The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding **[the max backup memory]** 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.

6.4. ADELG Security Functional Requirements

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

FDP_ACC.2/ADEL Complete Access Control

FDP_ACC.2.1/ADEL

The TSF shall enforce the **[ADEL access control SFP** on **S.ADEL,S.JCRE, S.JCVM, O.JAVAOBJECT, O.APPLET and O.CODE_PKG]** and all operations among subjects and objects covered by the SFP.

Refinement:
The operations involved in the policy are:

- OP.DELETE_APPLET,
- OP.DELETE_PCKG,
- OP.DELETE_PCKG_APPLET.

FDP_ACC.2.2/ADEL

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

FDP_ACF.1/ADEL Security Attribute Based Access Control

FDP_ACF.1.1/ADEL

The TSF shall enforce the **[ADEL access control SFP]** to objects based on the following: **[**

Subject/Object	*Attributes*
S.JCVM	**Active Applets**
S.JCRE	**Selected Applet Context, Registered Applets, Resident Packages**
O.CODE_PKG	**Package AID, Dependent Package AID, Static References**
O.APPLET	**Applet Selection Status**
.JAVAOBJECT	**Owner**

]

FDP_ACF.1.2/ADEL

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**
In the context of this policy, an object O is reachable if and only one of the following conditions hold:

1) ***the owner of O is a registered applet instance A (O is reachable from A),***

2) ***a static field of a resident package P contains a reference to O (O is reachable from P),***

3) ***there exists 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): S.ADEL may perform OP.DELETE_APPLET upon an O.APPLET only if,**

1) ***S.ADEL is currently selected,***

2) ***there is no instance in the context of O.APPLET that is active in any logical channel and***

3) ***there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance distinct from O.APPLET, or O.JAVAOBJECT is reachable from a package P***

- **R.JAVA.15 ([JCRE22], §11.3.4.1, Multiple Applet Instance Deletion):S.ADEL may perform OP.DELETE_APPLET upon several O.APPLET only if,**

1) ***S.ADEL is currently selected,***

2) ***there is no instance of any of the O.APPLET being deleted that is active in any logical channel and***

3) ***there is no O.JAVAOBJECT owned by any of the O.APPLET being deleted such that either O.JAVAOBJECT is reachable from an applet instance distinct from any of those O.APPLET, or O.JAVAOBJECT is reachable from a package P.***

- **R.JAVA.16 ([JCRE22], §11.3.4.2, Applet/Library Package Deletion): S.ADEL may perform OP.DELETE_PCKG upon an O.CODE_PKG only if,**

1) ***S.ADEL is currently selected,***

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

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

- **R.JAVA.17 ([JCRE22], §11.3.4.3, Applet Package and Contained Instances Deletion): S.ADEL may perform OP.DELETE_PCKG_APPLET upon an O.CODE_PKG only if,**

1) ***S.ADEL is currently selected,***

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

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

4) ***for every O.APPLET of those being deleted it holds that: (i) there is no instance in the context of O.APPLET that is active in any logical channel and***

(ii) there is no O.JAVAOBJECT owned by O.APPLET such that either O.JAVAOBJECT is reachable from an applet instance not being deleted, or O.JAVAOBJECT is reachable from a package not being deleted.

]

FDP_ACF.1.3/ADEL

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[none]**.

FDP_ACF.1.4/ADEL

The TSF shall explicitly deny access of **[any subject but S.ADEL to O.CODE_PKG or O.APPLET for the purpose of deleting them from the card]**.

FDP_RIP.1/ADEL Subset Residual Information Protection

FDP_RIP.1.1/ADEL

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

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 [selection: modify] the security attributes **[Registered Applets and Resident Packages to the Java Card RE].**

FMT_MSA.3/ADEL Static Attribute Initialisation

FMT_MSA.3.1/ADEL

The TSF shall enforce the **[ADEL access control SFP]** to provide [selection: restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ADEL

The TSF shall allow the **[none],** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/ADEL Specification of Management Functions

FMT_SMF.1.1/ADEL

The TSF shall be capable of performing the following management functions: **[modify the list of registered applets' AIDs and the Resident Packages].**

FMT_SMR.1/ADEL Security roles

FMT_SMR.1.1/ADEL

The TSF shall maintain the roles**: [applet deletion manager]**.

FMT_SMR.1.2/ADEL

The TSF shall be able to associate users with roles.

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 [JCRE30] §11.3.4]**.

6.5. RMIG Security Functional Requirements

This group specifies the policies that control the access to the remote objects and the flow of information that takes place when the RMI service is used. The rules relate mainly to the lifetime of the 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 must be allocated on the card as global arrays. Therefore, the storage of references to those arrays must be restricted as well.

The JCRMI 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.

Refinement:

The operations involved in this policy are:

- OP.GET_ROR,
- OP.INVOKE.

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: **[**

Subjects/Objects	*Attributes*
S.JCRE	**Selected Applet Context**
O.REMOTE_OBJ	**Owner, Class, Identifier, ExportedInfo**
O.REMOTE_MTHD	**Identifier**
O.RMI_SERVICE	**Owner, Returned References**

]

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: **[**

- **R.JAVA.18: S.CAD may perform OP.GET_ROR upon 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: S.JCRE may perform OP.INVOKE upon O.RMI_SERVICE, O.ROR and O.REMOTE_MTHD only if O.ROR is valid (as defined in [JCRE22], §8.5) and it belongs to the Returned References of O.RMI_SERVICE, and if the Identifier of O.REMOTE_MTHD matches one of the remote methods in the Class of the O.REMOTE_OBJ to which O.ROR makes reference.]**

FDP_ACF.1.3/JCRMI

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[none]**.

FDP_ACF.1.4/JCRMI [Editorially Refined]

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[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 S.JCRE, S.CAD, I.RORD and OP.RET_RORD(S.JCRE,S.CAD,I.RORD)].**

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: **[**

Subjects/Information	*Security attributes*
I.RORD	**ExportedInfo**

]

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 rules hold: **[OP.RET_RORD (S.JCRE, S.CAD, I.RORD) is permitted only if the attribute ExportedInfo of I.RORD has the value "true" ([JCRE30], §8.5)]**.

FDP_IFF.1.3/JCRMI

The TSF shall enforce the **[**
Prior to invoking the remote method,

1) If the remote object identifier is not valid, returns an error response;

2) If the remote object identifier has not been returned during the current selection, returns an error response;

3) If the method identifier does not match any remote methods in the remote class associated with the identified remote object, returns an error response;

4) If the length of the INVOKE message is inconsistent with the signature of the remote method, returns an error response;

5) If there is insufficient space to allocate array parameters for the remote method, returns an error response.

Upon return from the remote method,

1) If there is insufficient space to allocate the array response from the remote method, returns an error response;

2) If a remote object is being returned, and its associated remote object identifier has not been previously returned during the current selection session and there is insufficient space to add the remote object identifier to the session remote object identifier list, returns an error response.

].

FDP_IFF.1.4/JCRMI

The TSF shall explicitly authorise an information flow based on the following rules: **[**

Prior to invoking the remote method. The following information is permitted,

1) The remote object identifier is valid;

2) The method identifier does match a remote methods in the remote class associated with the identified remote object;

3) The length of the INVOKE message is consistent with the signature of the remote method;

].

FDP_IFF.1.5/JCRMI

The TSF shall explicitly deny an information flow based on the following rules: **[**

Prior to invoking the remote method, the following information is forbidden.

1) The remote object identifier is not valid;

2) The method identifier does not match any remote methods in the remote class associated with the identified remote object;

3) The length of the INVOKE message is inconsistent with the signature of the remote method;

].

FMT_MSA.1/EXPORT Management of Security Attributes

FMT_MSA.1.1/EXPORT

The TSF shall enforce the **[JCRMI access control SFP]** to restrict the ability to [selection: modify] the security attributes**: [ExportedInfo of O.REMOTE_OBJ to its owner applet]**.

FMT_MSA.1/REM_REFS Management of Security Attributes

FMT_MSA.1.1/REM_REFS

The TSF shall enforce the **[JCRMI access control SFP]** to restrict the ability to [selection: modify] the security attributes **[Returned References of O.RMI_SERVICE to its owner applet]**.

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 [selection: restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/JCRMI

The TSF shall allow the **[none],** to specify alternative initial values to override the default values when an object or information is created.

FMT_REV.1/JCRMI Revocation

FMT_REV.1.1/JCRMI [Editorially Refined]

The TSF shall restrict the ability to revoke the Returned References of **[O.RMI_SERVICE to the Java Card RE]**.

FMT_REV.1.2/JCRMI

The TSF shall enforce the rules that determine the lifetime of remote object references.

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 ExportedInfo of O.REMOTE_OBJ,**
- **modify the security attribute Returned References of O.RMI_SERVICE.**

].

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.

6.6. ODELG Security Functional Requirements

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

FDP_RIP.1/ODEL Subset Residual Information Protection

FDP_RIP.1.1/ODEL

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

FPT_FLS.1/ODEL Failure with Preservation of Secure State

FPT_FLS.1.1/ODEL

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

6.7. CARG Security Functional Requirements

This group includes requirements for preventing the installation of package

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 [Editorially Refined]

The TSF shall be able to relate the **[identity]** of the originator of the information, and the **[application package contained in]** the information to which the evidence applies.

FCO_NRO.2.3/CM

The TSF shall provide a capability to verify the evidence of origin of information to **[recipient given an immediate verification of the origin of the package using an electronic signature mechanism, and no evidence is kept on the card for future verifications]**.

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.INSTALLER, S.BCV, S.CAD and I.APDU]** and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2/CM

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

FDP_IFF.1/CM Simple Security Attributes

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: **[Key used for encryption and decryption, amount of total package block, and package block sequence number]**.

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: **[APDU from a trusted CAD, unmodified and has a right sequence number]**.

FDP_IFF.1.3/CM

The TSF shall enforce the **[none additional information flow control SFP rules]**.

FDP_IFF.1.4/CM

The TSF shall explicitly authorise an information flow based on the following rules: **[none rules, based on security attributes, that explicitly authorise information flows]**.

FDP_IFF.1.5/CM

The TSF shall explicitly deny an information flow based on the following rules: **[none: rules, based on security attributes, that explicitly deny information flows]**.

FDP_UIT.1/CM Data Exchange Integrity

FDP_UIT.1.1/CM

The TSF shall enforce the **[PACKAGE LOADING information flow control SFP]** to [selection: receive] user data in a manner protected from [selection: modification, deletion, insertion, 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.

FIA_UID.1/CM Timing of Identification

FIA_UID.1.1/CM

The TSF shall allow **[applet selection and INITIALIZE UPDATE command, get data command]** 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.

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: change default, modify] the security attributes **[key for decryption and encryption]** to **[card manager]**.

FMT_MSA.3/CM Static Attribute Initialisation

FMT_MSA.3.1/CM

The TSF shall enforce the **[PACKAGE LOADING information flow control SFP]** to provide [selection: restrictive**]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/CM

The TSF shall allow the **[none]** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMF.1/CM Specification of Management Functions

FMT_SMF.1.1/CM

The TSF shall be capable of performing the following management functions: **[add and modify the key which is used to establish the security session]**.

FMT_SMR.1/CM Security Roles

FMT_SMR.1.1/CM

The TSF shall maintain the roles **[CM]**.

FMT_SMR.1.2/CM

The TSF shall be able to associate users with roles.

FTP_ITC.1/CM Inter-TSF Trusted Channel

FTP_ITC.1.1/CM

The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/CM [Editorially Refined]

The TSF shall permit **[the CAD placed in the card issuer secured environment]** to initiate communication via the trusted channel.

FTP_ITC.1.3/CM

The TSF shall initiate communication via the trusted channel for **[loading/installing a new application package on the card]**.

6.8. 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 the Secure State

FPT_FLS.1.1/SCP

The TSF shall preserve a secure state when the following types of failures occur: **[**

- **lack of NVM**
- **random generator and cryptographic co-processor failure**
- **RAM read/write failure**
- **Card Tearing**

].

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: **[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 **[all physical attacks]** to the **[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 **[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 **[all atomic operations on NVM]**, 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 **[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 the NVM to static and objects' fields or native data interrupted by power failure or communication failure]** 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 **[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 **[Data Processing Policy]** on **[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.

FPT_EMSEC.1 TOE Emanation

FPT_EMSEC.1.1

The TOE shall not emit **[electromagnetic field]** in excess of **[allows deduce sensitive information]** enabling access to **[any sensitive data of D.APP_PURSE]** and **[keys of D.APP_EASY_KEYS]**.

FPT_EMSEC.1.2

The TSF shall ensure **[any users]** are unable to use the following interface **[smart card circuit contacts and contactless]** to gain access to **[any sensitive data of D.APP_PURSE]** and **[keys of D.APP_EASY_KEYS]**.

FPT_TST.1/SCP TSF Testing

FPT_TST.1.1/SCP

The TSF shall run a suite of self tests [selection: during initial start-up] to demonstrate the correct operation of [selection: the TSF].

FPT_TST.1.2/SCP

The TSF shall provide authorized users with the capability to verify the integrity of [selection: TSF data].

FPT_TST.1.3/SCP

The TSF shall provide authorized users with the capability to verify the integrity of [selection: TSF].

FCS_RND.1/SCP_true Quality Metric for Random Number

FCS_RND.1.1/SCP_true

The TSF shall provide a mechanism to generate random numbers that meet the **[AIS31 P2 class]**.

FCS_RND.1/SCP_pseudo Quality Metric for Random Number

FCS_RND.1.1/SCP_pseudo

The TSF shall provide a mechanism to generate random numbers that meet the **[AIS20 K4 class with seed entropy at least 112 bits]**.

6.9. CMGRG Security Functional Requirements

This group contains the security requirements for the card manager.

The security requirements below helps defining a policy for controlling access to card content management operations and for expressing card issuer security concerns. This policy shall be highly dependent on the particular security and card management architecture present in the card. Therefore the policy should be accordingly refined when developing conformant Security Targets.

FDP_ACC.1/CMGR Subset Access Control

FDP_ACC.1.1/CMGR

The TSF shall enforce the **[CARD CONTENT MANAGEMENT access control SFP]** on **[**
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: **[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 **[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: **[none]**.

FDP_ACF.1.4/CMGR

The TSF shall explicitly deny access of subjects to objects based on the **[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 [selection: restrictive] default values for security attributes that are used to enforce the *S*FP.

FMT_MSA.3.2/CMGR

The TSF shall allow the **[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: **[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: **[manage card content in GlobalPlatform registry]**.

FIA_UID.1/CMGR Timing of Identification

FIA_UID.1.1/CMGR

The TSF shall allow **[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.

6.10. ISOLATION Security Functional Requirements

This group contains the security requirements for the isolation between Java Card System and the native application (i.e. the EasyCard application in the current ST).

The following security requirements define a security policy for the internal firewall which controls the interaction between the Java Card System and the resident native applications. This security policy is referred as “ISOLATION”, which is a extended firewall access control policy.

FDP_ACC.1/ISOLATION Subset Access Control

FDP_ACC.1.1/ISOLATION

The TSF shall enforce the **[ISOLATION access control SFP]** on **[following subjects, objects, and operations among subjects and objects covered by the ISOLATION]**.

Subject	*Security Attribute*	*Operations*
JavaCard Application	**Context**	**Get Access to DF or EF file**
Native Application	**Context**	**Get Access to JavaCard Object**
Objects	**Security Attribute**	**Operations**
JavaCard Objects	**Context**	**Being access by Native Application**
DF and EF files	**Context**	**Being access by a Java Card Applet**

Note: N/A

FDP_ACF.1/ISOLATION Security Attribute based Access Control

FDP_ACF.1.1/ISOLATION

The TSF shall enforce the **[ISOLATION access control SFP]** to objects based on the following: **[subjects and objects controlled under the ISOLATION, and for each, the relevant security attributes]**.

Subject	*Security Attribute*
JavaCard Application	**Context**
Native Application	**Context**
Objects	**Security Attribute**
JavaCard Objects	**Context**
DF and EF files	**Context**

Note: N/A

FDP_ACF.1.2/ISOLATION

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

The Extended Firewall access control SFP shall be enforced on all Java Objects based on the Context with the following rules:

1) **If the Context is neither 0xFF nor 0xFE, the Java Applications namespace is processed, other than the native Applications namespace (i.e. OS File System);**
2) **The access to a Java Object (including Java Applet instance) shall comply with the rules of Firewall as [JCRE 6.2.8 "Class and Object Access Behavior"**

The Extended Firewall access control SFP shall enforced on all native Applications based on the current Selected ADF with the following rules:

1) **If the current Context is 0xFF, the native Applications namespace is processed, other than the normal Java Applications namespace (i.e. GP Registry);**
2) **If the target data is located in an EF under the ADF associated with another native Application other than the current selected native Application, the access is denied;**
3) **Otherwise, the access is allowed**

FDP_ACF.1.3/ISOLATION

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

1) **For EasyCard Native application, it is allowed to access child EFs under the associated ADF.**
2) **For javacard applet, see JCRE $6.2.8.**

FDP_ACF.1.4/ISOLATION

The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

1) **For EasyCard Native application, it is not allowed to access child EFs under the the ADF associated with another native application.**
2) **For Java Card Applet, see JCRE $6.2.8.**

FMT_MSA.1/ISOLATION Management of Security Attributes

FMT_MSA.1.1/ISOLATION

[Editorially Refined] The TSF shall enforce the **[ISOLATION access control SFP]** to restrict the ability to [selection: change **[none]**] the security attributes **[Context]** to **[Issuer and Card Holder]**.

FMT_MSA.3/ISOLATION Static attribute initialization

FMT_MSA.3.1/ISOLATION

The TSF shall enforce the **[ISOLATION access control SFP]** to provide [selection: restrictive **[none]**] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/ ISOLATION

The TSF shall allow the **[none]** to specify alternative initial values to override the default values when an object or information is created.

FMT_SMR.1/ ISOLATION Security Roles

FMT_SMR.1.1/ISOLATION

The TSF shall maintain the roles **[as following]**.

- **Java Card RE (JCRE)**
- **CPU Easy Card Application**
- **Mifare Legacy Application.**

FMT_SMR.1.2/ISOLATION

The TSF shall be able to associate users with roles.

FMT_SMF.1/ISOLATION Specification of Management Functions

FMT_SMF.1.1/ISOLATION

The TSF shall be capable of performing the following management functions:
- **Assign values to the Context**

FIA_UID.1/ISOLATION Timing of Identification

FIA_UID.1.1/ISOLATION

The TSF shall allow **[the GET DATA command]** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2/ISOLATION

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

FDP_ACC.1/MIFARE Subset Access Control

FDP_ACC.1.1/MIFARE

The TSF shall enforce the **[MIFARE ISOLATION access control SFP]** on [CPU Application, Legacy Application, and the transaction data]

FDP_ACF.1/MIFARE Security Attribute based Access Control

FDP_ACF.1.1/MIFARE

The TSF shall enforce the **[MIFARE ISOLATION access control SFP]** to objects based on the following:
[Subjects:
- **CPU application**

Objects:
- **Data of Mifare application**

SFP-relevant security attributes:
- **Purse Version Number**

]

FDP_ACF.1.2/MIFARE

The TSF shall enforce the following rules to determine if an operation among

controlled subjects and controlled objects is allowed:

[

- **When the Purse Version Number byte has been set with a value of 2 (i.e. Level 2), the Mifare application will be disabled, and the data of Mifare application will not be allowed to access in any condition.]**

FDP_ACF.1.3/MIFARE

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[none]**

FDP_ACF.1.4/MIFARE

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[none]**

FMT_MSA.3/MIFARE Static Attribute Initialisation

FMT_MSA.3.1/MIFARE

The TSF shall enforce the **[MIFARE ISOLATION access control SFP]** to provide [selection: permissive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/MIFARE

The TSF shall allow the **[none]** to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.1/MIFARE Management of Security Attributes

FMT_MSA.1.1/MIFARE

The TSF shall enforce the **[MIFARE ISOLATION access control SFP]** to restrict the ability to [selection: **[switch from Level 1 to Level 2]**] the security attributes [Purse Version Number] to [Issuer].

FMT_SMR.1/MIFARE Security Roles

FMT_SMR.1.1/MIFARE

The TSF shall maintain the roles **[Issuer, CPU application, Java Card System]**.

FMT_SMR.1.2/MIFARE

The TSF shall be able to associate users with roles.

FIA_UID.1/MIFARE Timing of Identification

FIA_UID.1.1/MIFARE

The TSF shall allow **[to read the Purse Version Number]** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2/MIFARE

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

6.11. EasyCard Security Functional Requirements

This group contains the security requirements for the EasyCard application, which apply the security specifications defined in [CPU_FS_ECC] and [KMS_ECC].

FCS_CKM.1/APP_AES Cryptographic Key Generation

FCS_CKM.1.1/APP_AES

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **[AES-Session Key Generation Algorithm]** and specified cryptographic key sizes **[128 bit]** that meet the following standard: **[FIPS-197]**.

Application note: The generation of the session keys and derived keys uses a AES with 16 bytes keys with a input generated by the Random Number Generator, defined in FCS_RND.1/SCP_pseudo

FCS_CKM.2/APP_AES Cryptographic key distribution

FCS_CKM.2.1/APP_AES

The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method **[the CHANGE KEY [with ciphertext and KCV] command on APP ADMIN KEY in Level 2]** that meets the following standard: **[none]**.

Application Note:

An AES Key object shall be represented with the following attributes: 1-byte Key Type; 1-byte Key Version Number; 16-byte Key Value.

The ciphertext of APP ADMIN KEY is calculated using AES-128 ECB mode encryption process with Session Administrative Key.

The Key Check Value (KCV) is used to check the validity of the current APP ADMIN KEY. The KCV is calculated using AES-128 ECB mode encryption process with the current APP ADMIN KEY on 16-byte "00 00 …", and only fetches the first 3 bytes of the result.

The following keys are concerned by this SFR:

- **CPU ADMIN KEY:** for CPU Card authentication and administrative
- **DEBIT/CREDIT KEY:** for payment
- **SIGN KEY:** for signature generation for Debit/Credit
- **ISSUER KEY:** for purse parameter updating
- **APP ADMIN KEY:** for Application authentication and data integrity

FCS_CKM.4/APP_AES Cryptographic Key Destruction

FCS_CKM.4.1/APP_AES

The TSF shall destroy cryptographic keys in accordance with a specified key destruction method **[volatile memory erasure at the end of session]** that meets the following standard: **[None]**.

FCS_COP.1/APP_AES_MAC Cryptographic Operation

FCS_COP.1.1/APP_ AES_MAC

The TSF shall perform **[AES secure messaging-message authentication code]** in accordance with a specified cryptographic algorithm **[AES-CBC-MAC]** and cryptographic key sizes **[128 bits]** that meet the following specification: **[ISO 9797-1 (MAC algorithm 3, block cipher AES)]**.

FCS_COP.1/APP_AES Cryptographic Operation

FCS_COP.1.1/APP_AES

The TSF shall perform **[AES 128 data encryption and decryption]** in accordance with a specified cryptographic algorithm **[AES]** and cryptographic key sizes **[128 bits]** that meet the following standards: **[FIPS-197]**.

Application note: The AES 128 data encryption and decryption (ASE-128-ENCRYPT) is used for generate the card and terminal tokens, generating the card and the terminal random data input. The EasyCard Application uses the FCS_RND.1/SCP_pseudo to generate its random data.

Application note: The AES 128 data encryption and decryption (ASE-128-ENCRYPT) is used also for generating the debit and credit token and receipts.

FDP_ACC.1/APP_EASY Subset Access Control

FDP_ACC.1/APP_EASY

The TSF shall enforce **[EasyCard Access Control Policy SFP]** on **[terminals gaining write, read and modification access to data in the User Data]**.

Application note: The data of the D.APP_DT_LOGs are user data. The TSF shall enforce the following rules to determine if an operation among the Debit Transaction Log File is allowed:

- *It has a Read Access Condition of Free and*
- *It has an Update Access Condition of Locked and*
- *It can be read freely using the Read Record command and*
- *It is updated during the Debit Purse command.*

FDP_ACF.1/APP_EASY Security Attribute Based Access Control

FDP_ACF.1/APP_EASY

The TSF shall enforce the **[EasyCard Access Control Policy SFP]** to objects based on the following according to **[**

1. **Subjects:**
   a) **Load Agent,**
   b) **Merchant,**
   c) **Terminal Device,**
2. **Objects:**
   a) **the User Data,**
3. **Security attributes**
   a) **Authentication status of terminals,**
   b) **Terminal Authentication token called TATOKEN,**
   c) **Card Authentication token called CATOKEN**

].

FDP_ACF.1.2/APP_EASY

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[The successfully authenticated Load Agent or Merchant must follow the rules for all access methods and the access rules defined in the specification of the**

[CPU_FS_ECC]].

FDP_ACF.1.3/APP_EASY

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[The User Data can be updated during the Debit Purse command and the Debit Purse command must following the rules according to the specification of the [CPU_FS_ECC]. The D.APP_DT_LOGs can be read freely using the READ RECORD command]**.

FDP_ACF.1.4/APP_EASY

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[none]**

FDP_SDI.2/APP_EASY Stored Data Integrity Monitoring and Action

All of the User data and the TSF Data persistently stored by TOE have the user attribute “integrity checked persistent stored data”.

The following volatile data used by TOE have the user attribute “integrity checked volatile data”:

- AES cryptographic key (volatile keys as session keys) used by EasyCard Application.
- Security relevant status variables of the card (e. g. transaction status for the transactions or the authentication status for mutual authenticate) (volatile)

FDP_SDI.2.1/APP_EASY

The TSF shall monitor user data stored within the TSF for **[integrity errors]** on all objects, based on the following attributes: **[integrity checked persistent or volatile stored data]**.

FDP_SDI.2.2/APP_EASY

Upon detection of a data integrity error, the TSF shall: **[**

- **Prohibit the use of the altered data**
- **Inform the user about integrity error.**

].

FDP_UIT.1/APP_EASY Basic Data Exchange Integrity

FDP_UIT.1.1/APP_EASY

The TSF shall enforce the **[EasyCard Access Control Policy SFP]** to be able to [selection: transmit and receive] user data in a manner protected from [selection: modification, deletion, insertion and replay] errors.

FDP_UIT.1.2/APP_EASY

The TSF shall be able to determine on receipt of user data, whether [selection: modification, deletion, insertion and replay] has occurred.

FIA_UID.1/Purse_Manage Timing of Identification

FIA_UID.1.1/Purse_manage

The TSF shall allow **[**

- **to transfer the information necessary to perform parameters update transactions from the EasyCard application to the terminal device owned by the issuer,**
- **to initiate a session and authenticate the EasyCard application by checking the CATOKEN**

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

FIA_UID.1.2/Purse_manage

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

FIA_UID.1/Terminal_device Timing of Identification

FIA_UID.1.1/Terminal_device

The TSF shall allow **[**

- **to transfer the information necessary to perform credit/debit transactions from the EasyCard application to the terminal device owned by the load agent or the merchant,**
- **to initiate a session and authenticate the EasyCard application by checking the TATOKEN,**

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

FIA_UID.1.2/Terminal_device

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

FIA_UAU.1/Purse_manage Timing of Authentication

FIA_UAU.1.1/Purse_manage

The TSF shall allow **[**

- **to transfer the information necessary to perform parameters update transactions from the EasyCard application to the terminal device owned by the issuer,**
- **to initiate a session and authenticate the EasyCard application by checking the CATOKEN**

] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2/Purse_manage

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

Application note: The user stands for the terminal device owned by the issuer. User authentication stands for the terminal device owned by the issuer authentication by the EasyCard application*. Purse parameters update transactions cannot be performed before the user is authenticated.*

FIA_UAU.1/Terminal_device Timing of Authentication

FIA_UAU.1.1/Terminal_device

The TSF shall allow **[**

- **to transfer the information necessary to perform credit/debit transactions from the ECCC to the terminal device owned by the load agent or the merchant,**
- **to initiate a session and authenticate the EasyCard application by checking the TATOKEN**

] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2/Terminal_device

The TSF shall require each user to be successfully authenticated before allowing any other credit/debit actions on behalf of that user.

Application note: The user stands for the terminal device owned by the load agent or the merchant. User authentication stands for load device authentication by the ECCC. Credit/debit transactions cannot be performed before user authentication.

FIA_UAU.4/Terminal_device Single-Use Authentication Mechanisms

FIA_UAU.4.1/Terminal_device

The TSF shall prevent reuse of authentication data related to the **[terminal**

authentication mechanism].

FIA_UAU.4/Purse_manage Single-use authentication mechanisms

FIA_UAU.4.1/Purse_manage

The TSF shall prevent reuse of authentication data related to the **[card authentication mechanism]**.

Application note: e.g the INITIATE PROCESSING command.

FIA_UAU.5/APP_EASY Multiple Authentication Mechanisms

FIA_UAU.5.1/APP_EASY

The TSF shall provide **[**

- **Card Authentication Protocol**
- **Terminal Authentication Protocol**

] to support user authentication.

FIA_UAU.5.2/ APP_EASY

The TSF shall authenticate any user's claimed identity according to **[the rules specified by the specification of the [CPU_FS_ECC], the TOE accepts the authentication attempt as the user during the using phase]**.

FIA_UAU.6/Terminal_device Re-Authenticating

FIA_UAU.6.1/Terminal_device

The TSF shall re-authenticate the user under the conditions **[**

- **PAYMENT COMMAND: Read Purse/INITATE Processing/Debit Purse/Credit Purse**
- **PURSE MANAGEMENT COMMAND: Put Data/Write Lock**
- **FILE MANAGEMENT COMMAND: Get Data/External Authenticate/Read Record/Update Record/Append Record**

]

Application note: The "user" stands for the terminal device.

FIA_UAU.6/Purse_manage Re-Authenticating

FIA_UAU.6.1/Purse_manage

The TSF shall re-authenticate the user under the conditions **[beginning of a purse parameters update transaction]**.

Application note: The "user" stands for the terminal device.

FIA_AFL.1/APP_EASY Authentication Failure Handling

FIA_AFL.1.1/APP_EASY

The TSF shall detect when [selection: 200] unsuccessful authentication attempts occur related to **[a certain authentication event with an unsuccessful result]**.

FIA_AFL.1.2/APP_EASY

When the defined number of unsuccessful authentication has been [selection: met], the TSF **[shall perform actions that block the keys of the same type and the commands which used the this type of key will never be handled]**.

FMT_MSA.1/APP_EASY Management of Security Attributes

FMT_MSA.1.1/APP_EASY

The TSF shall enforce the **[EasyCard Access Control Policy SFP]** to restrict the ability to [selection: change_default] the security attributes **[personalization keys]** to the **[personalization agent]**.

FMT_MSA.3/APP_EASY Static Attribute Initialization

FMT_MSA.3.1/APP_EASY

The TSF shall enforce the **[EasyCard Access Control Policy SFP]** to provide [selection: permissive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/APP_EASY

The TSF shall allow the **[personalization agent]** to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1/APP_EASY_MNG Management of TSF Data

FMT_MTD.1.1/ APP_EASY_MNG

The TSF shall restrict the ability to [selection: modify] the **[Purse data]** to the **[Issuer]**.

FMT_MTD.1/APP_EASY_KEY Management of TSF Data

FMT_MTD.1.1/ APP_EASY_KEY

The TSF shall restrict the ability to [selection: change] the **[APP ADMIN KEY]** to **[the Issuer]**.

FMT_SMF.1/APP_EASY Specification of Management Functions

FMT_SMF.1.1/APP_EASY

The TSF shall be capable of performing the following security management functions: **[**

- **Purse management**
- **Change APP ADMIN KEY**

].

FMT_SMR.1/APP_EASY Security Roles

FMT_SMR.1.1/APP_EASY

The TSF shall maintain the roles **[**

- **Issuer,**
- **Terminal device,**
- **Load agent,**
- **Merchant,**
- **Card holder**

].

FMT_SMR.1.2/APP_EASY

The TSF shall be able to associate users with roles.

FPT_RPL.1/APP_EASY Replay Detection

FPT_RPL.1.1/APP_EASY

The TSF shall detect replay for the following entities **[**

- **debit transactions,**
- **credit transactions,**
- **APP ADMIN KEY change transactions,**
- **purse parameters update transactions**

].

FPT_RPL.1.2/APP_EASY

The TSF shall perform **[the abort of the transaction]** in process when replay is detected.

FTP_ITC.1/APP_EASY Inter-TSF Trusted Channel

FTP_ITC.1.1/APP_EASY

The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/APP_EASY

The TSF shall permit [the remote trusted IT product **[terminal to communicate with EasyCard Application]** to initiate communication via the trusted channel.

FTP_ITC.1.3/APP_EASY

The TSF shall initiate communication via the trusted channel for **[EasyCard user data and TSF data exchanging functionalities]**.

7. Security Assurance Requirements

The security assurance requirements are based on the package EAL4 augmented with AVA_VAN.5 and ALC_DVS.2.

8. Security Requirement Rationale

8.1. Mapping between Security Objectives and Security Requirements

Security Objective	*Security Functional Requirement*
O.SID	FIA_ATD.1/AID, FIA_UID.2/AID, FMT_MSA.1/JCRE, FMT_MSA.3/JCRMI, FMT_MSA.1/REM_REFS, FMT_MSA.1/EXPORT, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_MSA.3/FIREWALL, FMT_MSA.1/CM, FMT_MSA.3/CM, FDP_ITC.2/Installer, FMT_SMF.1/CM, FMT_SMF.1/ADEL, FMT_SMF.1/JCRMI,

	FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FIA_USB.1/AID, FMT_MSA.1/JCVM, FMT_MSA.3/JCVM
O.FIREWALL	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FMT_SMR.1/Installer, FMT_MSA.1/CM, FMT_MSA.3/CM, FMT_SMR.1/CM, FMT_MSA.3/FIREWALL, FMT_SMR.1/JC, FMT_MSA.1/ADEL, FMT_MSA.3/ADEL, FMT_SMR.1/ADEL, FMT_MSA.1/EXPORT, FMT_MSA.1/REM_REFS, FMT_MSA.3/JCRMI, FMT_REV.1/JCRMI, FMT_SMR.1/JCRMI, FMT_MSA.1/JCRE, FDP_ITC.2/Installer, FDP_ACC.2/JCRMI, FDP_ACF.1/JCRMI, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FMT_SMF.1/ADEL, FMT_SMF.1/JCRMI, FMT_SMF.1/CM, FMT_SMF.1/JC, FMT_MSA.2/FIREWALL_JCVM, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE, FMT_MSA.1/JCVM, FMT_MSA.3/JCVM
O.GLOBAL_ARRAYS_CONFID	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FDP_RIP.1/bArray, FDP_RIP.1/APDU, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT
O.GLOBAL_ARRAYS_INTEG	FDP_IFC.1/JCVM, FDP_IFF.1/JCVM
O.NATIVE	FDP_ACF.1/FIREWALL
O.OPERATE	FAU_ARP.1, FDP_ROL.1/FIREWALL, FIA_ATD.1/AID, FPT_FLS.1/ADEL, FPT_FLS.1, FPT_FLS.1/ODEL, FPT_FLS.1/Installer, FDP_ITC.2/Installer, FPT_RCV.3/Installer, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL, FPT_TDC.1, FIA_USB.1/AID
O.REALLOCATION	FDP_RIP.1/ABORT, FDP_RIP.1/APDU,

	FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/TRANSIENT, FDP_RIP.1/ADEL, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS
O.RESOURCES	FAU_ARP.1, FDP_ROL.1/FIREWALL, FMT_SMR.1/Installer, FMT_SMR.1/JC, FMT_SMR.1/ADEL, FMT_SMR.1/JCRMI, FPT_FLS.1/Installer, FPT_FLS.1/ODEL, FPT_FLS.1, FPT_FLS.1/ADEL, FPT_RCV.3/Installer, FMT_SMR.1/CM, FMT_SMF.1/ADEL, FMT_SMF.1/JCRMI, FMT_SMF.1/CM, FMT_SMF.1, FMT_MTD.1/JCRE, FMT_MTD.3/JCRE
O.ALARM	FPT_FLS.1/Installer, FPT_FLS.1, FPT_FLS.1/ADEL, FPT_FLS.1/ODEL, FAU_ARP.1
O.CIPHER	FCS_CKM.1/API_DES, FCS_CKM.1/API_RSA, FCS_CKM.2/API, FCS_CKM.3/API, FCS_CKM.4/API, FCS_COP.1/API_DES, FCS_COP.1/API_RSA, FPR_UNO.1
O.KEY-MNGT	FCS_CKM.1/API_DES, FCS_CKM.1/API_RSA, FCS_CKM.2/API, FCS_CKM.3/API, FCS_CKM.4/API, FCS_COP.1/API_DES, FCS_COP.1/API_RSA, FPR_UNO.1, FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FDP_SDI.2/API, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT
O.PIN-MNGT	FDP_RIP.1/ODEL, FDP_RIP.1/OBJECTS, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/ABORT, FDP_RIP.1/KEYS, FPR_UNO.1, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_ROL.1/FIREWALL, FDP_SDI.2/API, FDP_ACC.2/FIREWALL, FDP_ACF.1/FIREWALL
O.REMOTE	FDP_ACC.2/JCRMI, FDP_ACF.1/JCRMI, FDP_IFC.1/JCRMI, FDP_IFF.1/JCRMI,

	FMT_MSA.1/EXPORT, FMT_MSA.1/REM_REFS, FMT_MSA.3/JCRMI, FMT_REV.1/JCRMI, FMT_SMR.1/JCRMI
O.TRANSACTION	FDP_ROL.1/FIREWALL, FDP_RIP.1/ABORT, FDP_RIP.1/ODEL, FDP_RIP.1/APDU, FDP_RIP.1/bArray, FDP_RIP.1/KEYS, FDP_RIP.1/ADEL, FDP_RIP.1/TRANSIENT, FDP_RIP.1/OBJECTS
O.OBJ-DELETION	FDP_RIP.1/ODEL, FPT_FLS.1/ODEL
O.DELETION	FDP_ACC.2/ADEL, FDP_ACF.1/ADEL, FDP_RIP.1/ADEL, FPT_FLS.1/ADEL, FPT_RCV.3/Installer, FMT_MSA.1/ADEL,,FMT_MSA.3/ADEL, FMT_SMR.1/ADEL
O.LOAD	FCO_NRO.2/CM, FDP_IFC.2/CM, FDP_IFF.1/CM, FDP_UIT.1/CM, FIA_UID.1/CM, FTP_ITC.1/CM
O.INSTALL	FDP_ITC.2/Installer, FPT_RCV.3/Installer, FPT_FLS.1/Installer
O.SCP.IC	FPT_PHP.3/SCP, FPT_ITT.1/SCP,
O.RND	FCS_RND.1/SCP_true FCS_RND.1/SCP_pseudo
O.SCP.SUPPORT	FPT_RCV.3/SCP,FPT_RCV.4/SCP FDP_ITT.1/SCP, FPT_ITT.1/SCP FDP_IFC.1/SCP FPT_TST.1/SCP
O.SCP.RECOVERY	FPT_FLS.1/SCP, FRU_FLT.1/SCP FPT_RCV.3/SCP FPT_RCV.4/SCP
O.CARD-MANAGER	FDP_ACC.1/CMGR, FDP_ACF.1/CMGR, FMT_MSA.1/ CMGR, FMT_MSA.3/ CMGR, FMT_SMR.1/ CMGR, FMT_SMF.1/ CMGR, FIA_UID.1/ CMGR
O.ISOLATION	FDP_ACC.1/ISOLATION, FDP_ACF.1/ISOLATION, FMT_MSA.1/ISOLATION, FMT_MSA.3/ISOLATION, FMT_SMR.1/ISOLATION,

	FMT_SMF.1/ISOLATION, FIA_UID.1/ISOLATION FDP_ACC.1/MIFARE, FDP_ACF.1/MIFARE, FMT_MSA.3/MIFARE, FMT_MSA-1/MIFARE, FMT_SMR.1/MIFARE, FIA_UID.1/MIFARE
O.CRYPTO	FCS_CKM.1/APP_AES FCS_CKM.2/APP_AES FCS_CKM.4/APP_AES FCS_COP.1/APP_ AES_MAC FCS_COP.1/APP_AES FCS_RND.1/SCP_true FCS_RND.1/SCP_pseudo
O.APP_INTEG	FDP_SDI.2/APP_EASY FDP_UIT.1/APP_EASY
O.LEAKAGE	FDP_UCT.1/APP_EASY FPT_EMSEC.1
O.APP_ACL	FDP_ACC.1/APP_EASY FDP_ACF.1/APP_EASY FDP_SDI.2/APP_EASY
O.APP_MNG	FCS_COP.1/APP_AES FMT_MSA.1/APP_EASY FMT_MSA.3/APP_EASY FMT_MTD.1/APP_EASY_INI FMT_MTD.1/APP_EASY_PER FMT_MTD.1/ APP_EASY_MNG FMT_MTD.1/APP_EASY_KEY FMT_SMF.1/APP_EASY FMT_SMR.1/APP_EASY FIA_UID.1/Purse_manage FIA_UAU.1/Purse_manage FIA_UAU.4/Purse_manage FIA_UAU.6/Purse_manage FIA_AFL.1
O.TRANS	FCS_COP.1/APP_AES FCS_COP.1/APP_AES_MAC FIA_UID.1/Terminal_device FIA_UAU.1/Terminal_device FIA_UAU.6/Terminal_device FIA_AFL. FTP_ITC.1/APP_EASY

	FPT_RPL.1/APP_EASY

8.1.1. Rationale of Coverage between Requirements and Objectives

Identification

O.SID

Coverage rationale provided in the section 7.3.1.1.1 of the [JCS-OP-PP].

Execution

O.FIREWALL

Coverage rationale provided in the section 7.3.1.1.2 of the [JCS-OP-PP].

O.GLOBAL_ARRAYS_CONFID

Coverage rationale provided in the section 7.3.1.1.2 of the [JCS-OP-PP].

O.GLOBAL_ARRAYS_INTEG

Coverage rationale provided in the section 7.3.1.1.2 of the [JCS-OP-PP].

O.NATIVE

Coverage rationale provided in the section 7.3.1.1.2 of the [JCS-OP-PP].

O.OPERATE

Coverage rationale provided in the section 7.3.1.1.2 of the [JCS-OP-PP].

O.REALLOCATION

Coverage rationale provided in the section 7.3.1.1.2 of the [JCS-OP-PP].

O.RESOURCES

Coverage rationale provided in the section 7.3.1.1.2 of the [JCS-OP-PP].

Services

O.ALARM

Coverage rationale provided in the section 7.3.1.1.3 of the [JCS-OP-PP].

O.CIPHER

Coverage rationale provided in the section 7.3.1.1.3 of the [JCS-OP-PP].

O.KEY-MNGT

Coverage rationale provided in the section 7.3.1.1.3 of the [JCS-OP-PP].

O.PIN-MNGT

Coverage rationale provided in the section 7.3.1.1.3 of the [JCS-OP-PP].

O.REMOTE

Coverage rationale provided in the section 7.3.1.1.3 of the [JCS-OP-PP].

O.TRANSACTION

Coverage rationale provided in the section 7.3.1.1.3 of the [JCS-OP-PP].

Object Deletion

O.OBJ-DELETION

Coverage rationale provided in the section 7.3.1.1.4 of the [JCS-OP-PP].

Applet Deletion

O.DELETION

Coverage rationale provided in the section 7.3.1.1.5 of the [JCS-OP-PP].

O.LOAD

Coverage rationale provided in the section 7.3.1.1.5 of the [JCS-OP-PP].

O.INSTALL

Coverage rationale provided in the section 7.3.1.1.5 of the [JCS-OP-PP].

Miscellaneous

O.SCP.IC

The TOE by means of the IC underlying platform includes protecting data from physical probing and chip analysis which includes user data transfers as stated in FPT_ITT.1/SCP. And it is directly met by FPT_PHP.3/SCP. Finally it requires a random number generator that is provided as stated in FCS_RND.1/SCP_true and FCS_RND.1/SCP_pseudo.

O.SCP.SUPPORT

O.SCP.SUPPORT is met by FPT_RCV.3/SCP and FPT_RCV.4/SCP. O.SCP.SUPPORT includes protecting internal user data transfers as stated in FDP_ITT.1/SCP. 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. O.SCP.SUPPORT includes protecting internal data transfers as stated in FPT_ITT.1/SCP.

OS.SCP.SUPPORT is also covered the testing part by means of FPT_TST.1/SCP.

O.SCP.RECOVERY

O.SCP.RECOVERY is met by FPT_FLS.1/INSTALLER. 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. And It met directly by FPT_FLS.1/ADEL, FPT_FLS.1/ODEL, FPT_FLS.1/ADEL, RU_FLT.1/SCP, FPT_RCV.3/SCP.

O.SCP.RECOVERY uses the FPT_RCV.4/SCP prevents information abnormity by recovers to a consistent and secure state in case of detected errors.

Card Manager

O.CARD-MANAGER

O.CARD-MANAGER access control and the policy are defined in FDP_ACC.1/CMGR, FDP_ACF.1/CMGR. And the definitions of the security attributes are defined in FMT_MSA.1/ CMGR, FMT_MSA.3/ CMGR. The FMT_SMR.1/ CMGR defines the role for the card manager and with FMT_SMF.1/CMGR the management functionalities. These management functionalities have not performed without identification as defined in FIA_UID.1/ CMGR.

Isolation

O.ISOLATION

O.ISOLATION is defined by means of the extended firewall as defined the combination of the following SFR: FDP_ACC.1/ISOLATION, FDP_ACF.1/ISOLATION, FMT_MSA.1/ISOLATION, FMT_MSA.3/ISOLATION, FMT_SMR.1/ISOLATION, FMT_SMF.1/ISOLATION, FIA_UID.1/ISOLATION, FDP_ACC.1/MIFARE, FDP_ACF.1/MIFARE, FMT_MSA.3/MIFARE, FMT_MSA-1/MIFARE, FMT_SMR.1/MIFARE, FIA_UID.1/MIFARE.

O.CRYPTO

O.CRYPTO is implemented by the SFR of the FCS class. They include symmetric algorithms as used for secure messaging, signature and random number generation, defined as a combination of the following SFR: FCS_CKM.1/APP_AES, FCS_CKM.2/APP_AES, FCS_CKM.4/APP_AES, , FCS_COP.1/APP_ AES_MAC, FCS_COP.1/APP_AES, , FCS_COP.1/APP_SIGN,.

O.RND

The quality of the random number generator is directly implemented by SFR FCS_RND.1/SCP_true, FCS_RND.1/SCP_pseudo

O.APP_INTEG

O.APP_INTEG is implemented by FDP_SDI.2/APP_EASY in the storage of data and with FDP_UIT.1/APP_EASY during the internal transference.

O.LEAKAGE

O.LEAKAGE is protected against leakage with FPT_EMSEC.1 and internally the confidentiality by FDP_UCT.1/APP_EASY.

O.APP_ACL

O.APP_ACL is mainly implemented by following SFRs the SFRs FDP_ACC.1/APP_EASY and FDP_ACF.1/APP_EASY, which require implementing the access rules defined in the security policy EasyCard Access Control Policy SFP as defined in O.APP_ACL. And the FDP_SDI.2/APP_EASY for the integrity of the relevant data.

O.APP_MNG

O.APP_MNG identifies the roles by means of FMT_SMR.1/APP_EASY allowing for each role the functions defined in FMT_SMF.1/APP_EASY. Then before performing the management operation: FMT_MTD.1/APP_EASY_INI, FMT_MTD.1/APP_EASY_PER, FMT_MTD.1/ APP_EASY_MNG, FMT_MTD.1/APP_EASY_KEY. The user should be identified as FIA_UID.1/Purse_manage, FIA_UAU.1/Purse_manage, FIA_UAU.6/Purse_manage and FIA_AFL.1. For perform the operations the TOE performs a mutual authentication with the terminal using FCS_COP.1/APP_AES.

O.TRANS

O.TRANS uses the FCS_COP.1/APP_AES for the mutual authentication and identifies the terminal by means of FIA_UID.1/Terminal_device, FIA_UAU.1/Terminal_device, FIA_UAU.6/Terminal_device and FIA_AFL.1. Then it uses the access control policy for define the operations FDP_ACC.1/APP_EASY, FDP_ACF.1/APP_EASY. And sign the operation done by FCS_COP.1/APP_SIGN.

All the replays are detected and avoided by FPT_RPL.1/APP_EASY. Finally the transactions use FCS_COP.1/APP_AES_MAC to sign the debit and credit token and receipts.

8.1.2. Dependencies Justification

All the dependences are covered in the current security target, expecting:

- **The dependency FIA_UID.1 of FMT_SMR.1/Installer is unsupported. This PP does not require the identification of the "installer" since it can be considered as part of the TSF.**
- **The dependency FIA_UID.1 of FMT_SMR.1/ADEL is unsupported. This PP does not require the identification of the "deletion manager" since it can be considered as part of the TSF.**
- **The dependency FMT_SMF.1 of FMT_MSA.1/JCRE is unsupported. The dependency between FMT_MSA.1/JCRE and FMT_SMF.1 is not satisfied because no management functions are required for the Java Card RE.**
- **The dependency FAU_SAA.1 of FAU_ARP.1 is unsupported. The dependency of FAU_ARP.1 on FAU_SAA.1 assumes that a "potential security violation" generates an audit event. On the contrary, the events listed in FAU_ARP.1 are self-contained (arithmetic exception, ill-formed bytecodes, access failure) and ask for a straightforward reaction of the TSFs on their occurrence at runtime. The JCVM or other components of the TOE detect these events during their usual working order. Thus, there is no mandatory audit recording in this PP.**

8.1.3. Security Assurance Requirement Justification

EAL4 is required for this type of TOE and product since it is intended to defend against sophisticated attacks. This evaluation assurance level allows a developer to gain maximum assurance from positive security engineering based on good practices. EAL4 represents the highest practical level of assurance expected for a commercial grade product. In order to provide a meaningful level of assurance that the TOE and its embedding product provide an adequate level of defense against such attacks: the evaluators should have access to the low level design and source code. The lowest for which such access is required is EAL4.

ALC_DVS.2

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE and the embedding product. The standard ALC_DVS.1 requirement mandated by EAL4 is not enough.

Due to the nature of the TOE and embedding product, it is necessary to justify the sufficiency of these procedures to protect their confidentiality and integrity. ALC_DVS.2 has no dependencies.

AVA_VAN.5

The TOE is intended to operate in hostile environments. AVA_VAN.5 "Advanced methodical vulnerability analysis" is considered as the expected level for Java Card technology-based products hosting sensitive applications, in particular in payment and identity areas.

9. TOE Summary specifications

The following lists describe how the TOE meets the security requirements detailed in the section 6.

FDP_ACC.2/FIREWALL

To meet FDP_ACC.2, the TSF implements the FIREWALL mechanism compatible with [JCRE30].

FDP_ACF.1/FIREWALL

To meet FDP_ACF.1, the TSF implements the rules of FIREWALL SFP according to [JCRE30] §6.2.8.

FDP_IFC.1/JCVM

To meet FDP_IFC.1, the TSF implements the Java Card Virtual Machine compatible with Java Card v3.0.1.

FDP_IFF.1/JCVM

To meet FDP_IFF.1, the TSF implements the Java Card Virtual Machine compatible with Java Card v3.0.1, which applies the FIREWALL mechanism, do runtime checking, and may throw exceptions defined in [JCAPI30].

FDP_RIP.1/OBJECTS

To meet FDP_RIP.1, the TSF fills all the associated memory units to 0x00 when the resource is allocated to class instances or arrays.

FMT_MSA.1/JCRE

To meet FMT_MSA.1, the TSF only accepts modifications of the Selected Applet Context to the JCRE applying the rules defined in [JCRE30] §4.

FMT_MSA.1/JCVM

To meet FMT_MSA.1, the TSF only accepts modifications of the Currently Active Context to the JCRE applying the rules defined in [JCVM30] §3.4.

FMT_MSA.2/FIREWALL

To meet FMT_MSA.2, the TSF checks the values of security attributes, and ensures that they are well defined accordance with [JCRE30].

FMT_MSA.3/FIREWALL

To meet FMT_MSA.3, the TSF creates all objects with series of security attributes with well defined specific values. And managing the permissive or restrictive setting of default values for a given access control SFP. For security attributes of the Firewall, Selection Status default value is none selection. And when an applet is selected, the value changed; Active Applets default value is none. When an applet is in selecting status, the value changed. Currently active context default value is none. Selected applet context is none. Sharing value default value is none. Context default value is none. Life Time default value is none.

FMT_MSA.3/JCVM

To meet FMT_MSA.3, the TSF creates all objects with series of security attributes with well defined specific values. And managing the permissive or restrictive setting of default values for a given access control SFP. For JCVM, the security attributes: the Stack Pointer default value is bottom of the stack. The PC pointer default value is none. The Frame Pointer default value is bottom of the stack. The Stack TOP pointer default value is top of the stack. The context default value is none.

FMT_SMF.1/JC

To meet FMT_SMF.1, the TSF implements the functions of modifying the Currently Active Content, the Selected Applet Context, and the Active Applets. The means of modifying the currently active context is only by selecting an applet. Other method to modify is forbidden. The selected applet context is a group context according to logical channels. Those value is only modified by selecting an applet or selecting another applet on an logical channel. To

modify the active applets, only method is by selecting this applet or selecting another applet on corresponding logical channel.

FMT_SMR.1/JC

To meet FMT_SMR.1, the TSF has two roles: Java Card RE (JCRE) and Java Card VM (JCVM).

FCS_CKM.1/API_DES

To meet FCS_CKM.1, the TSF supports the key generation and diversification according to [JCAPI30]. See the class of KeyBuilder in [JCAPI30], the current evaluation appraisal this class with the following key sizes: 16/24 (2x56 bits / 3x56 bits).

FCS_CKM.1/API_RSA

To meet FCS_CKM.1, the TSF supports the key generation and diversification according to [JCAPI30]. See the class of KeyBuilder and KeyPair in [JCAPI30], the current evaluation appraisal these two classes with the following key sizes: from 1024 to 2048.

FCS_CKM.2/API

To meet FCS_CKM.2, the TSF supports the key setting according to [JCAPI30].

FCS_CKM.3/API

To meet FCS_CKM.3, the TSF implements the Key class and its subclasses according to [JCAPI30].

FCS_CKM.4/API

To meet FCS_CKM.4, the TSF implements the clearKey() method in the Key class according to [JCAPI30].

FCS_COP.1/API_DES

To meet FCS_COP.1, the TSF implements the signature generation and verification, cryptographic checksum generation for integrity and versification of checksum, secure hash (message digest), cryptographic key encryption and decryption, and cryptographic key agreement according to [JCAPI30], the

current evaluation appraisal these implementation with the following key sizes: 16/24 (2x56 bits / 3x56 bits).

FCS_COP.1/API_RSA

To meet FCS_COP.1, the TSF implements the signature generation and verification, cryptographic checksum generation for integrity and versification of checksum, secure hash (message digest), cryptographic key encryption and decryption, and cryptographic key agreement according to [JCAPI30], the current evaluation appraisal these implementation with the following key sizes from 1024 to 2048 bits.

FDP_RIP.1/ABORT

To meet FDP_RIP.1, the TSF fills the associated memory units to NULL (i.e. 0x00) of any reference to an object instance created during an aborted transaction.

FDP_RIP.1/APDU

To meet FDP_RIP.1, the TSF fills all the associated memory units to 0x00 for the APDU buffer when allocate resource to it.

FDP_RIP.1/bArray

To meet FDP_RIP.1, the TSF fills all the related memory units to 0x00 when de-allocate resource from the bArray argument of install() method.

FDP_RIP.1/KEYS

To meet FDP_RIP.1, the TSF fills all the related memory units to 0x00 when de-allocate resource from the cryptographic buffer.

FDP_RIP.1/TRANSIENT

To meet FDP_RIP.1, the TSF fills all the related memory units to 0x00 when de-allocate resource from any transient object.

FDP_ROL.1/FIREWALL

To meet FDP_ROL.1, the TSF implements the rollback mechanism for all the writing type operations in [JCRE30] §6.2.8 and Object creations.

FAU_ARP.1

To meet FAU_ARP.1, the TSF throws SecurityException against violation of Firewall SFP, throws RuntimeException against violation of JCVM SFP, throws SystemException for unavailability of resources, and throws TransactionException against transaction errors.

FDP_SDI.2/API

To meet FDP_SDI.2, the TSF throws an exception when integrity errors are detected, the integrity checking is CRC algorithm over the keys, pins and its associated security attributes.

FPR_UNO.1

To meet FPR_UNO.1, the TSF implements the PIN Timing protection. The PIN timing protection is that checking the PIN value process is all the same, whether the PIN value is right or not. So this can avoid comparison the difference between the right PIN value and wrong PIN value.

FPT_FLS.1/CM

To meet FPT_FLS.1, the TSF shall preserve a secure state when the FAU_ARP.1 type of failures occur and SFR sets JCRE as the current context when the Java Card VM begins running after a card reset ([JCRE30] §6.2.3), performs on power loss and reset as the description in [JCRE30] §3.6 and §7.1, and performs on RF signal loss as the description in [JCRE30] §3.6.1. those potential security violations include: the power loss when loading package or installing an applet or deleting package or destroying an applet. So the card will be not in a secure state. When power on again, the card will return it to a secure state.

FPT_TDC.1

To meet FPT_TDC.1, the TSF interpreters and links the information in the CAP file according to [JCVM30] §6.

FIA_ATD.1/AID

To meet FIA_ATD.1, the TSF holds the package AID, applet's version number, registered applet AID, and applet Selection Status.

FIA_UID.2/AID

To meet FIA_UID.2, the TSF does the runtime checking for every operation against the currently selected applet, the package that is the subject's owner, or the JCRE.

FIA_USB.1/AID

To meet FIA_USB.1, the TSF associates every package with a package AID. And every applet has an AID to identify it. And AID is an object of java card, It is created when installing applet. For package's AID, it is a data in cap file, so the AID is in code stored in java card system.

FMT_MTD.1/JCRE

To meet FMT_MTD.1, the TSF only allows the JCRE to modify the list of registered applets' AIDs.

FMT_MTD.3/JCRE

To meet FMT_MTD.3, the TSF checks the validity of the registered applets' AIDs.

FDP_ITC.2/Installer

To meet FDP_ITC.2, the TSF implements the installer compatible with the one defined in GlobalPlatform v2.1.1.

FMT_SMR.1/Installer

To meet FMT_SMR.1, the TSF has a role of installer. The installer is belong to ISD, the installer is authenticated by external authentication according to GlobalPlatform v2.1.1 specification.

FPT_FLS.1/Installer

To meet FPT_FLS.1, the TSF rollbacks all operations done during the loading of a package or installation of an applet.

FPT_RCV.3/Installer

To meet FPT_RCV.3, the TSF rollbacks all operations done during the loading of a package or installation of an applet. In card, there is an flag to identify the state of package or applet. On card reset, the SCP will detect this flag. If it

is unsuccessful, the card should return to state before loading package or installing applet.

FDP_ACC.2/ADEL

To meet FDP_ACC.2, the TSF implements the applet deletion manager according to [JCRE30] §11.3.4.

FDP_ACF.1/ADEL

To meet FDP_ACF.1, the TSF implements the applet deletion manager according to [JCRE30] §11.3.4.

FDP_RIP.1/ADEL

To meet FDP_RIP.1, the TSF fills all the related memory units to 0x00 when de-allocate resource from applet instances and/or packages when they are deleted or required to be deleted.

FMT_MSA.1/ADEL

To meet FMT_MSA.1, the TSF only allows JCRE to modify the list of Registered Applets and Resident Packages. When applet is deleted, the card should checking the its dependency relationship. If the applet is permit to delete, then should delete all objects belonging to it, but the static objects are not permit to delete, only the package is deleted, the static objects should delete. And modify the applet register, such as the state and privilege of the applet.

FMT_MSA.3/ADEL

To meet FMT_MSA.3, the TSF sets the ISD as the Default Selected Application when the applet instance with the privilege of Default Selected Application is deleted.

FMT_SMF.1/ADEL

To meet FMT_SMF.1, the TSF implements the functions of modifying the list of registered applets' AIDs and the Resident Packages.

FMT_SMR.1/ADEL

To meet FMT_SMR.1, the TSF has a role of applet deletion manager.

FPT_FLS.1/ADEL

To meet FPT_FLS.1, the TSF processes the deletion of package or applet instance as atomic.

FDP_ACC.2/JCRMI

To meet FDP_ACC.2, the TSF implements the RMIService class according to [JCRE30] §8.5.

FDP_ACF.1/JCRMI

To meet FDP_ACF.1, the TSF implements the RMIService class according to [JCRE30] §8.5.

FDP_IFC.1/JCRMI

To meet FDP_IFC.1, the TSF allocates array parameters of remote method invocations as global array objects, sends the initial remote object reference descriptor when SELECT FILE is successfully processed, and returns the result of a remote method invocation as [JCRE30] §8.3.5.

FDP_IFF.1/JCRMI

To meet FDP_IFF.1, the TSF allocates array parameters of remote method invocations as global array objects, sends the initial remote object reference descriptor when SELECT FILE is successfully processed, and returns the result of a remote method invocation as [JCRE30] §8.3.5.

FMT_MSA.1/EXPORT

To meet FMT_MSA.1, the TSF only allows the Object's owner to modify the Exported status of a remote Object by invoking its method export() or unexport(). Other method to modify the security attributes is forbidden.

FMT_MSA.1/REM_REFS

To meet FMT_MSA.1, the TSF only allows the remote Object's owner applet to modify that remote Object's Returned References.

FMT_MSA.3/JCRMI

To meet FMT_MSA.3, the TSF sets the default value of the Exported attribute as true.

FMT_REV.1/JCRMI

To meet FMT_REV.1, the TSF manages the lifetime of remote object references as [JCRE30] §8.5.

FMT_SMF.1/JCRMI

To meet FMT_SMF.1, the TSF implements the functions of modifying the Export Info of remote Objects and the Returned References of remote Objects.

FMT_SMR.1/JCRMI

To meet FMT_SMR.1, the TSF has a role of applet.

FDP_RIP.1/ODEL

To meet FDP_RIP.1, the TSF fills all the related memory units to 0x00 when de-allocate resource from the objects owned by the context of an applet instance which triggered the execution of requestObjectDelection() method.

FPT_FLS.1/ODEL

To meet FPT_FLS.1, the TSF holds a flag indicating the Object deletion request. So, if Object deletion is started, it will be executed until all unreferenced Objects owned by the applet that requested the execution of the method are deleted successfully.

FCO_NRO.2/CM

To meet FCO_NRO.2, the TSF supports the Token verification as defined in GlobalPlatform v2.1.1.

FDP_IFC.2/CM

To meet FDP_IFC.2, the TSF implements the installer compatible with the one defined in GlobalPlatform v2.1.1.

FDP_IFF.1/CM

To meet FDP_IFF.1, the TSF implements the installer compatible with the one defined in GlobalPlatform v2.1.1.

FDP_UIT.1/CM

To meet FDP_UIT.1, the TSF implements the INSTALL [for load] with Load File Data Block Hash, which ensures the integrity of loaded package. For the detail of Load File Data Block Hash, see [GPCS] §7.7.7.

To protect the loading application from replaying attack, a DAP associated with the first LOAD command may be applied. The DAP data is a signature generated on the Load File Data Block Hash with Asymmetric Encryption algorithm (i.e. RSA). For the detail of DAP mechanisms, see [GPCS] §7.7.8.

FIA_UID.1/CM

To meet FIA_UID.1, the TSF allows the process of GET DATA and SELECT and INITIALIZE UPDATE command before the user is identified.

FMT_MSA.1/CM

To meet FMT_MSA.1, the TSF only allows the Card Manager to register the package AID.

FMT_MSA.3/CM

To meet FMT_MSA.3, the TSF sets the default Associated SD of loaded package as ISD (Issuer SD).

FMT_SMF.1/CM

To meet FMT_SMF.1, the TSF implements the functions of modifying the list of registered applets' AIDs and the Resident Packages.

FMT_SMR.1/CM

To meet FMT_SMR.1, the TSF has a role of card manager.

FTP_ITC.1/CM

To meet FTP_ITC.1, the TSF implements the Secure Channel Protocol compatible with GlobalPlatform v2.1.1 SCP02 (i='15').

FPT_FLS.1/SCP

To meet FPT_FLS.1, the TSF preserve a secure state when failure occurs for future use of deal with fails. When a random generator or cryptographic co-processor failure and lack NVM, the TSF should lock the card.

FRU_FLT.2/SCP

To meet FRU_FLT.2, the TSF ensures the operation of all the TOE's capabilities when following failures occur:

1. When one bit error in the memory (NVM or RAM) is detected by the hardware, it will be automatically corrected by the hardware, and the operation of the TOE's capability will be ensured.

2. When writing data into NVM, and the power is cut off, the TSF will apply transaction protecting functions of Watchdata OS, to ensure the operations of TOE's capabilities correct after power on.

FPT_PHP.3/SCP

To meet FPT PHP.3, The TSF checks all alarm-generating security features. When physical attack is detected by the IC, it will generate a security reset. For example, when the abnormity detected by the frequency sensor or light sensor and so on, a security reset will be executed automatically.

FPT_RCV.4/SCP

To meet FPT_RCV.4, the TSF adopts the transaction mechanism in Watchdata OS to protect the interrupted writing into NVM by power failure or communication failure.

When some data is writing into the NVM, and the power is cut off, the TSF will recover the just written data in the NVM to the previous status.

FDP_ITT.1/SCP

To meet FDP_ITT.1, the TSF can prevent some control of user data when it is transmitted between memories and CPU then the disclosure is protected.FPT_ITT.1/SCP

FDP_IFC.1/SCP

To meet FDP_IFC.1,the TSF enforces the Data Processing Policy on all confidential data.

FPT_TST.1/SCP

To demonstrate the correct operation of TSF, we use CRC algorithm for TSF check. To demonstrate the integrity of data, we use CRC check for it.

FPT_EMSEC.1

The IC is designed to avoid disclosing of sensitive information by means of electromagnetic fields. Moreover the TSF provides balancing conditions that avoid disclosing when sensitive information is handled having into account the hamming weight to the sensitive data.

FCS_RND.1/SCP_true

The TSF uses random number generated by the underlying platform IC which ensures the level of entropy specified in the IC's ST. The TSF also execute a chi-square test of goodness recommended in the IC's guidance. Finally, the TSF does not use the random number in case of any test failure.

FCS_RND.1/SCP_pseudo

The pseudo random generation is used for Java Card API, which requires the user (Java Card Applet) to input the seed for pseudo random generation. With the external seed, the random is generated by the underlying platform IC which ensures the level of entropy specified in the IC's ST. The TSF also execute a chi-square test of goodness recommended in the IC's guidance. Finally, the TSF does not use the random number in case of any test failure.

FDP_ACC.1/CMGR

To meet FDP_ACC.1, the TSF implements the ISD defined as GlobalPlatform v2.1.1 to manage the Card Content.

FDP_ACF.1/CMGR

To meet FDP_ACF.1, the TSF implements the ISD defined as GlobalPlatform v2.1.1 to manage the Card Content.

FMT_MSA.1/CMGR

Before loading a Java Card Applet package into card, the off-card entity that issues the loading process is authenticated by the Secure Channel protocol of GlobalPlatform. When the off-card entity is successfully authenticated, the loading is processed by the Card Manager. When loading is done successfully,

the Card Manager registers the package AID and Applet AIDs on card according to the input AIDs hold by the loaded package.

FMT_MSA.3/CMGR

To meet FMT_MSA.3, the TSF sets the default Associated SD of loaded package as ISD (Issuer SD).

FMT_SMR.1/CMGR

To meet FMT_SMR.1, the TSF has a role of ISD, which acts representing the Issuer.

FMT_SMF.1/CMGR

To meet FMT_SMF.1, the TSF supports the installation of post-issued applications and the deletion of packages or applet instances.

FIA_UID.1/CMGR

Before process the card content management functionalities, the Card Manager requires the off-card entity to be successfully authenticated with the Secure Channel protocol.

FDP_ACC.1/ISOLATION

To meet FDP_ACC.1, the TSF adopts the Java Card Firewall access control between Java Card Applets. And adopt extended Firewall access control between Java Card environment and EasyCard native application.

The TSF introduces an Owner byte in the file header of each DF and EF in the Native File System. Before access the target DF or EF, the TSF will firstly check its Owner byte with the currently active application (Native application). If its Owner byte is not matched the currently active application, the following access to the target DF or EF will be denied.

FDP_ACF.1/ISOLATION

To meet FDP_ACF.1, the TSF adopts the Native File System SFP, which controls the reading and writing operations on the files associated with a special native Application.

FMT_MSA.1/ISOLATION

To meet FMT_MSA.1, the TSF only allows JCRE to modify the currently active context.

FMT_MSA.3/ISOLATION

To meet FMT_MSA.3, the TSF sets the JCRE as the initial currently active context.

FMT_SMR.1/ ISOLATION

To meet FMT_SMR.1, the TSF has the roles of JCRE, JCVM, and native Applications.

FMT_SMF.1/ISOLATION

To meet FMT_SMF.1, the TSF supports the functionality of context switching between JCRE, applet instance, and native Applications.

FIA_UID.1/ISOLATION

To meet FIA_UID.1, the TSF allows the process of GET DATA before the user is identified.

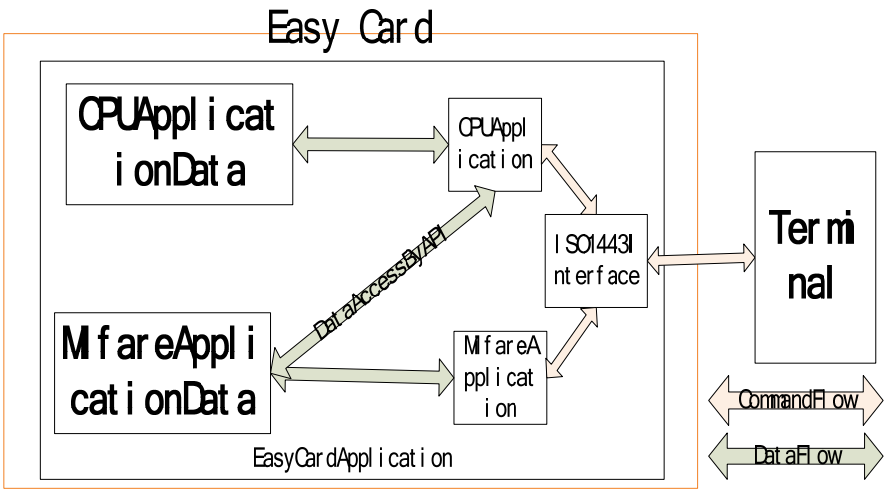
FDP_ACC.1/MIFARE

To meet the FDP_ACC.1/MIFARE, the MIFARE ISOLATION access control mechanism based on the following technology is used:

Mifare Sector's Privilege: When the Mifare application is determined to disable, the Mifare sector's privilege byte value will be set as "never", which means that the sectors can not be allowed to access in any condition.

Purse Version Number – Level 1 / Level 2: The first byte of the Electronic Purse file is the Purse Version Number, which is used to control the EasyCard operation mode: Level 1 or Level 2. When the Purse Version Number is personalized as Level 2, or is switch from Level 1 to Level 2, all the routines in the EasyCard application will query the Purse Version Number byte, and make a decision that the Mifare application can not be allowed to access any more. For another word, when Level 2 is the current Purse Version Number value, the Level 1 can not be returned back or switched to in any condition.

The following figure describes the relationship between Mifare Application (i.e. Legacy Application) and CPU Application:



When the Card is initially issued, and the legacy terminals (CAD) are not upgraded to support the CPU functions. In this case, the Purse Balance and Transaction Log are stored in the Mifare sectors, and are accessed by both Legacy Application and CPU Application. The other data are stored in the CPU data domain, and can not be accessed by Legacy Application dual to the Mifare technology. It is called as Level 1 in [CPU_FS_ECC].

After the terminals are updated to support the CPU functions, the Card may be switched to Level 2, and the Legacy Application is disabled by the CPU Application and can not be access again. Simultaneously, the data previously stored in Mifare sectors are automatically copied to the associated CPU data domain.

The TOE of this ST focuses on the Level 2 and the CPU Application, while the Mifare Application (i.e. Legacy Application) was disabled.

FDP_ACF.1/MIFARE

To meet FDP_ACF.1/MIFARE, the MIFARE ISOLATION mechanism is used to control the access to the data of Mifare application, which is described in FDP_ACC.1/MIFARE.

FMT_MSA.3/MIFARE

To meet FMT_MSA.3/MIFARE, the default value of the Purse Version Number is initialized in card manufacturing.

FMT_MSA.1/MIFARE

To meet FMT_MSA.1/MIFARE, the Purse Number Version value is only allowed to switch from 0 (i.e. Level 1) to 1 (i.e. Level2). No roles are allowed, and there is no way, to switch the Purse Number Version from Level 2 to Level 1.

FMT_SMR.1/MIFARE

To meet FMT_SMR.1/MIFARE, only the Issuer is maintained to switch the Purse Version Number from Level 1 to Level 2 when its initial value is Level 1.

FIA_UID.1/MIFARE

To meet FIA_UID.1/MIFARE, the Electronic Purse with the Purse Version Number is allowed to be read before the roles are identified, by which the EasyCard operation mode (whether the Mifare application access is forbidden) is determined.

FCS_CKM.1/APP_AES

The TSF, meeting the FCS_CKM.1, uses 128-bit key to encode data in accordance with AES-Session Key Generation Algorithm. AES is symmetric cipher.

A Derived Key shall be calculated using AES-128-ENCRYPT by the Master key .The Master key is a 16-byte AES Key. Encryped data is a 16-byte value. Then, A Session Key shall be calculated using DERIVED KEY.A Card Transaction Counter used for Session Key is a 3-byte couter which is incremented by 1 for every session.

Application Note: Only the application session keys are concerned by this SFR.

FCS_CKM.2/APP_AES

The TSF, meeting the FCS_CKM.2, requires cryptographic keys to be distributed to the EasyCard application with the CHANGE KEY command, which may update the value of the Application Administrative Key.

Application Note:

The following keys are concerned by this SFR.

- CPU ADMIN KEY: CPU Card authentication and administrative
- DEBIT/CREDIT KEY: payment
- SIGN KEY: Signature generation for Debit/Credit
- ISSUER KEY: Purse parameter update

- App ADMIN KEY: Application authentication and data integrity

FCS_CKM.4/APP_AES

The TSF, met the FCS_CKM.4, destroys cryptographic keys in the volatile memory at the end of session.

The transaction related data is encrypted/decrypted using session key stored in the volatile memory. In the same card session, the previous session key is overwritten by the new session key. When the card session is end, the last session key disappears dual to the volatile memory clearing on power off.

FCS_COP.1/APP_AES_MAC

The TSF, meeting the FCS_COP.1, performs AES secure messaging-message authentication code in accordance with cryptographic algorithm AES-CBC and cryptographic key sizes 128 bits according to [KMS_ECC].

The AES-CBC MAC shall be calculated by AES-128 encryption in CBC mode. The IV (Initial Vector) used for AES-CBC MAC is a 16-byte special constant value, i.e. ‘EasyCardVersion2’.

FCS_COP.1/APP_AES

The TSF, meeting the FCS_COP.1, do data encryption with AES-128 algorithm as [FIPS PUB 197].

FDP_ACC.1/APP_EASY

The TSF, meeting the FDP_ACC.1, gains write, read and modification access to data in the User Data according to EasyCard access control policy SFP defined in [CPU_FS_ECC].

According to the File Access Condition, the file access functions are divided into two groups. The first group is protected by the Authentication procedure, and the second group is protected by the Authentication or/and MAC mechanism.

The file access functions are controlled by the 3-byte privilege in the file header. The first byte defines the Access Condition; the second byte defines the Key Type to be used for the protection; the third byte defines the Key Version Number to be used for the protection.

FDP_ACF.1/APP_EASY

The TSF, meeting the FDP_ACF.1, follows the rules for all access methods and the access rules defined in the specification of the [CPU_FS_ECC].

The Access Condition is defined as following:

PROTECTION	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	OPTION
Free	0	0	0	0	1	1	1	1	Free
Locked	1	1	1	1	1	1	1	1	Locked
External Authentcate	1	0	0	0	0	0	0	0	Required
MAC	0	1	0	0	0	0	0	0	Required
External Authenticate AND MAC	1	1	0	0	0	0	0	0	Required

FDP_SDI.2/APP_EASY

The TSF, meeting the FDP_SDI.2, requires all of the user data and the TSF data persistently stored by TOE have the user attribute “integrity checked persistent stored data”. AES and TDES cryptographic keys and security relevant status variables of the card have the user attribute ”integrity checked volatile data”.

FDP_UCT.1/APP_EASY

The TSF, meeting the FDP_UCT.1, transmits and receives secure messaging with AES-128 algorithm (including AES-128 cryptogram and AES-128 MAC).

FDP_UIT.1/APP_EASY

The TSF, meeting the FDP_UIT.1, transmits and receives user data in a manner protected from modification, deletion, insertion and replay errors. It requires exchanging secure messaging with AES-128 MAC.

Padding prior to performing a CBC-MAC-AES operation across a block of data is achieved in the following manner: Append a mandatory length byte,which is the size of the data block,to the left of the data block.Append a mandatory’80’to the right of the data block.If the resultant data block length is a multiple of bytes,no further padding is required.If the resultant data block

length is not a multiple of 16 bytes,append binary zero to the right of the data block until the data block length is a multiple of 16.

Application Note:

Transmit: The terminal device generates MAC of the data;

Receive: EasyCard application check the validity of MAC.

FIA_UID.1/Purse_Manage

The TSF, meeting the FIA_UID.1, transfers the information necessary to perform parameters update transactions from the EasyCard application to the terminal device owned by the issuer and initiating a session and authenticate the EasyCard application by checking the CATOKEN.

FIA_UID.1/Terminal_device

The TSF, meeting the FIA_UID.1, transfers the information necessary to perform credit/debit transactions from the EasyCard application to the terminal device owned by the load agent or the merchant and initiating a session and authenticate the EasyCard application by checking the CATOKEN.

FIA_UAU.1/Purse_manage

The TSF, meeting the FA_UAU.1, requires each user to be successfully authenticated before allowing any purse management actions on behalf of that user.

INITIATE PROCESSING command initiates a session and returns a CATOKEN to the terminal for card authentication purpose.

FIA_UAU.1/Terminal_device

The TSF, meeting the FA_UAU.1, requires each user to be successfully authenticated before allowing any other credit/debit actions on behalf of that user.

PAYMENT COMMAND: Read Purse/INITATE Processing/Debit Purse/Credit Purse

PURSE MANAGEMENT COMMAND: Put Data/Write Lock

FILE MANAGEMENT COMMAND: Get Data/External Authenticate/Read Record/Update Record/Append Record

FIA_UAU.4 /Purse_manage

The TSF, meeting the FIA_UAU.4, prevent reuse of authentication data related to the card authentication mechanism.

The related commands are: INITIATE PROCESSING.

The session key is used for the card authentication. The session is calculated by AES-128 encryption operation on the special data, which includes the Card Transaction Counter (CTC), according to [KMS_ECC]. The Card Transaction Counter (CTC) is a 3-byte counter, and incremented by 1 for every session.

FIA_UAU.4 /Terminal_device

The TSF, meeting the FIA_UAU.4, prevent reuse of authentication data related to the terminal authentication mechanism.

The related commands are:

- Payment commands

- Purse Management commands

- File Management commands.

The terminal authentication also uses the session key as described in FIA_UAU.4 /Purse_manage.

FIA_UAU.5/APP_EASY

The TSF, meeting the FIA_UAU.5, uses different mechanism to support user authentication. It requires authenticating any user's claimed identity according to the rules specified by the specification of the [CPU_FS_ECC].

For identifying the user, the TSF applies the mutual authentication: the terminal authenticates the card based on INITIATE PROCESSING with the Card Authentication Token (CATOKEN), and the card authenticates the terminal based on EXTERNAL AUTHENTICATE with the Terminal Authentication Token (TATOKEN).

FIA_UAU.6/Terminal_device

The TSF, meeting the FIA_UAU.6, re-authenticates the terminal device under the conditions as following:

Beginning of payment commands

Beginning of purse Management commands

Beginning of File management commands;

Application Note: The terminal device stands for the EasyCard application.

FIA_UAU.6/Purse_manage

The TSF, meeting the FIA_UAU.6, re-authenticates the terminal device under the conditions beginning of a purse parameters update transaction (i.e. payment command).

Application Note: The terminal device stands for SAM terminal device.

FIA_AFL.1

The TSF, meeting the FIA_AFL.1, requires the TSF be able to terminate the session establishment process after a specified number of unsuccessful user authentication attempts.

FMT_MSA.1/APP_EASY

The TSF, meeting the FMT_MSA.1, requires authorized users to manage the specified security attributes.

FMT_MSA.3/APP_EASY

The TSF, meeting the FMT_MSA.3, performs the EasyCard Access Control Policy SFP to provide permissive default values for security attributes that are used to enforce the SFP.

FMT_MTD.1/APP_EASY_MNG

The TSF, meeting the FMT_MTD.1, requires the issuer to modify the purse.

FMT_MTD.1/APP_EASY_KEY

The TSF, meeting the FMT_MTD.1, requires the issuer to change the APP ADMIN KEY.

FMT_SMF.1/APP_EASY

The TSF, meeting the FMT_SMF.1, performs the security management functions of initialization, personalization, purse management and change APP ADMIN KEY.

FMT_SMR.1/APP_EASY

The TSF, meeting the FMT_SMR.1, maintains the roles of issuer, terminal device, load agent, merchant and card holder.

FPT_RPL.1/APP_EASY

The TSF, meeting the FPT_RPL.1, performs the abort of the transaction in process when replay is detected.

FTP_ITC.1/APP_EASY

The TSF, met the FTP_ITC.1, builds a communication channel between easy card application and the remote trusted IT product.

The TSF builds a communication channel with the INITIATE PROCESSING command and the EXTERNAL AUTHENTICATE command. The INITIATE PROCESSING command initiates a session, and returns a Card Authentication Token (CATOKEN) to the terminal for card authentication. The EXTERNAL AUTHENTICATE command authenticates the terminal based on the Terminal Authentication Token (TATOKEN) in the command data. To issue the EXTERNAL AUTHENTICATE command, the INITIATE PROCESSING command shall be firstly executed successfully.

10. Glossary

Glossary	*Description*	*Remark*
API	Application Programming Interface	N/A
CAP	Converted Applet	N/A
CC	Common Criteria	N/A
COS	Card Operating System	N/A
RTR	Reset Transient Resource	N/A
DAP	Data Authentication Pattern	N/A
DGI	Data Grouping Identifier	N/A
DTR	Deselect Transient Resource	N/A
GP	Global Platform	N/A
AID	Application IDentifier	N/A
APDU	Application Protocol Data Unit	N/A
ATR	Answer To Reset	N/A
CAD	Card Acceptance Device	N/A
CBC	Cipher Block Chaining	N/A
CLA	Class	N/A
COS	Card Operation System	N/A

CRC	Cyclic Redundancy Code	N/A
DES	Data Encryption Standard	N/A
DF	Dedicate File	N/A
ECB	Electronic Code Book	N/A
ECC	Error Correction Code	N/A
EDU	Error Detection Unit	N/A
EF	Elementary File	N/A
EM	Electronic Money	N/A
ESN	Electronic Serial Number	N/A
ISD	Issuer Secure Domain	N/A
JCRE	Java Card Runtime Environment	N/A
JCVM	Java Card Virtual Machine	N/A
NVM	No-Volatile Memory	N/A
OS	Operating System	N/A
SD	Secure Domain	N/A
SIO	Shareable Interface Object	N/A
TOE	Target of Evaluation	N/A
TSF	TOE Security Function	N/A
TSP	TOE Security Policy	N/A
S-ENC	Secure Channel Encryption Key	N/A
S-MAC	Secure Channel Message Authentication Code Key	N/A
DEK	Data Encryption Key	N/A
SW	State Word	N/A

Note: N/A

11. References

[JCS-OP-PP] Java Card System Protection Profile Open Configuration, v2.6, April 19th 2010. ANSSI-CC-PP-2010/03. Sun Microsystems, Inc.

[JCSPP] Java Card System Protection Profile Collection Version 1.0b, August 2003. Sun Microsystems, Inc.

[JCRE30] Java Card Platform Runtime Environment Specification, Version 3.0.1, Classic Edition, May 2009. Sun Microsystems, Inc.

[JCAPI30] Java Card Platform Application Programming Interface, Version 3.0.1, Classic Edition, May 2009. Sun Microsystems, Inc.

[JCVM30] Java Card Platform Virtual Machine Specification, Version 3.0.1, Classic Edition, May 2009. Sun Microsystems, Inc.

[GPCS] GlobalPlatform Card Specification, Version 2.1.1, March 2003. GlobalPlatform Inc.

[GPCSRS] GlobalPlatform Card Security Requirements Specification, v1.0, May 2003. GlobalPlatform Inc.

[GPCSTG] GlobalPlatform Smart Card Security Target Guidelines, v1.0, October 2005. GlobalPlatform Inc.

[VGPCIR] Visa GlobalPlatform 2.1.1 Card Implementation Requirements, v2.0, July 2007. Visa International Service Association.

[CC-1] Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 3, July 2009. Part 1: Introduction and general model. CCMB-2009-07-001

[CC-2] Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 3, July 2009. Part 2: Security functional components. CCMB-2009-07-002

[CC-3] Common Criteria for Information Technology Security Evaluation, Version 3.1, revision 3, July 2009. Part 3: Security assurance components. CCMB-2009-07-003

[CPU_FS_ECC] EasyCard Phoenix Project: Functional Specification – CPU CARD, Release A19, December 9th, 2010. Gemalto.

[KMS_ECC] EasyCard Phoenix Project: Key Management Specification, Release A19, December 9th, 2010. Gemalto.

[ICST] Security Target (ST) M7892 A21 and comprises the Infineon Technologies Security Controller M7892 A21 with specific IC dedicated software and optional RSA v1.02.013, EC v1.02.013, SHA-2 v1.01 and Toolbox v1.02.013 libraries.

[AGDOPE] Operational User Guidance, version 0.8, January 16, 2013. SEC_20110121_963_AGD_OPE.1. Watchdata System Co., Ltd.

[AGDPRE] Preparative Procedures, version 0.6, January 16, 2013. SEC_20110121_963_AGD_PRE.1. Watchdata System Co., Ltd.

