



Reference : R0R23928_CCD_ASE_002	Release : 1.3p <small>(Printed copy not controlled: verify the version before using)</small>
Classification : Public	Pages : 104

Security Target Lite for ECC CPU card



Document Releases

Release <i>(X.yy)</i>	Date <i>(dd/mm/yy)</i>	Author	Modifications
1.3p	29/09/14	C. Crippa Martinez	Creation of the Security Target Lite from ST reference R0R23928_CCD_ASE_001, version 1.3



Table of Content

1	Introduction	7
1.1	Generalities	7
1.2	Reference documents	7
1.3	Terminologie	9
1.4	Conventions	9
2	ST Introduction	10
2.1	ST reference, TOE reference	10
2.1.1	ST reference	10
2.1.2	TOE reference	10
2.2	Overview	10
2.2.1	ST overview	10
2.2.2	Product overview	11
2.2.3	TOE overview	15
2.3	TOE description	15
2.3.1	Physical scope of the TOE	16
2.3.2	Logical scope of the TOE	16
2.3.3	TOE life-cycle	18
3	Conformance claims	22
4	Security Problem Definition	23
4.1	Assets	23
4.2	Threats	24
4.2.1	Threats agents	24
4.2.2	Threats	24
4.3	Organizational security policies (OSPs)	26
4.4	Assumptions	26
4.5	Composition tasks – Security problem definition part	26
4.5.1	Statement of Compatibility – Threats part	27
4.5.2	Statement of Compatibility – OSP part	28
4.5.3	Statement of Compatibility – Assumptions part	29
5	Security Objectives	31
5.1	Security objectives for the TOE	31
5.2	Security objectives for the operational environment	32
5.3	Composition tasks – Security objective part	32
5.3.1	Statement of Compatibility – Security objectives for the TOE part	33
5.3.2	Statement of Compatibility – Security objectives for the environment part	35
6	Security Requirements	36
6.1	Security functional requirements (SFRs)	36
6.1.1	FAU – Security audit	39
6.1.2	FCO – Communication	43
6.1.3	FCS – Cryptographic support	45
6.1.4	FDP – User data protection	48
6.1.5	FIA – Identification and authentication	65
6.1.6	FMT – Security management	69
6.1.7	FPR – Privacy	72
6.1.8	FPT – Protection of the TSF	73
6.2	Security assurance requirements (SARs)	78
6.3	Security functional requirements (SFRs) for IT environment	78
6.4	Composition tasks – Security requirements part	78
7	TOE Summary Specification	84
7.1	TOE security functions	84
7.1.1	HW security functions	84
7.1.2	SW security functions	84
7.2	Assurance measures	85



- 8 Annex 87
 - 8.1 Acronyms 87
 - 8.2 Security objectives rationale 87
 - 8.2.1 Assets and Protection – Coverage 87
 - 8.2.2 Threats/OSP/Assumptions and Assets – Coverage 88
 - 8.2.3 Threats/OSP/Assumptions and Security Objectives – Coverage 89
 - 8.3 Security requirements rationale 93
 - 8.3.1 Security Objectives and SFR – Coverage 93
 - 8.3.2 SFR dependencies – Coverage 101
 - 8.3.3 SAR – Rationale 101
 - 8.3.4 SAR dependencies – Coverage 102
 - 8.4 TOE summary specification rationale 103
 - 8.4.1 SAR and Assurances Measures – Coverage 104



List of Tables

Table 1 – Reference list.....	8
Table 2 – ECC CPU life-cycle	19
Table 3 – Actors.....	19
Table 4 – Environnements	20
Table 5 – Platform and application states	21
Table 6 – Assets	23
Table 7 – Threats agents.....	24
Table 8 – Threats.....	25
Table 9 – Organizational security policies (OSPs)	26
Table 10 – Assumptions	26
Table 11 – Composition – Threats part.....	28
Table 12 – Composition – OSPs part	29
Table 13 – Composition – Assumptions part	30
Table 14 – Security objectives of the TOE.....	32
Table 15 – Security objectives of the operational environment.....	32
Table 16 – Composition – Security objectives for the TOE part	35
Table 17 – Composition – Security objectives for the environment part	35
Table 18 – List of SFR	38
Table 19 – FAU_ARP.1 / FAU_SAA.1.....	39
Table 20 – FCS_CKM.3	46
Table 21 – FCS_COP.1	47
Table 22 – FDP_ACC.1/Atomicity.....	48
Table 23 – FDP_ACC.1/Life-cycle	49
Table 24 – FDP_ACC.2/Confidentiality and Integrity	50
Table 25 – Life-cycle SFP rules.....	52
Table 26 – Confidentiality and Integrity SFP rules	57
Table 27 – FDP_IFC.1/Application	59
Table 28 – Transaction SFP rules	61
Table 29 – FMT_MSA.1.....	69
Table 30 – FMT_MSA.3.....	70
Table 31 – FPT_PHP.3.....	76
Table 32 – List of SAR	78
Table 33 – Composition – Security requirements part.....	83
Table 34 – List of Security functions	84
Table 35 – List of Assurance measures	86
Table 36 – Assets and Protection – Coverage.....	87
Table 37 – Threats/OSP/Assumptions and Assets – Coverage	89
Table 38 – Threats/OSP/Assumptions and Security Objectives – Coverage	90
Table 39 – Security Objectives and SFR – Coverage	97
Table 40 – SAR dependencies – Coverage	103



List of Figures

Figure 1 – Architecture of Easy Card Corporation System	11
Figure 2 – ECC CPU Card CTL or DUAL	13
Figure 3 – Architecture of the ECC CPU Card	14
Figure 4 – TOE description (architecture)	16
Figure 5 – Easy Card Corporation life-cycle	18



1 Introduction

1.1 Generalities

This document is one of the Gemalto documents to be provided to Easy Card Corporation for the **ECC CPU card** project.

The aim of this document is to describe the security target (ST) of **ECC CPU card**.

The target of evaluation (TOE) provides the security requirements of embedded software (HW and OS) including the **CPU e-purse on GFCX6-EC** application.

The HW used to support the OS is the **M7794 A12 and G12 (SLE77CLFX2407PM)** component. This component **M7794 A12 and G12 (BSI-DSZ-CC-0883-2013)** is certified CC EAL5+ [9].

The objectives are to describe:

- The target of evaluation
- The security aspects
- The security objectives
- The security functions
- And the related rationale

1.2 Reference documents

The reference documents are given in following table:

	Document	Reference
RD [1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1 Revision 4.	CCMB-2012-09-001, September 2012
RD [2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1 Revision 4.	CCMB-2012-09-002, September 2012
RD [3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4.	CCMB-2012-09-003, September 2012
RD [4]	Common Methodology for Information Technology Security Evaluation Evaluation methodology, Version 3.1 Revision 4.	CCMB-2012-09-004, September 2012
RD [5]	Common Criteria mandatory technical document – Composite product evaluation for smart cards and similar devices, Version 1.2 April 2012	CCDB-2012-04-001, April 2012
RD [6]	Application of Attack Potential to Smartcards, Version 2.9 May 2013.	CCDB-2013-05-002, May 2013
RD [7]	JIL Attack Methods for Smartcards and Similar Devices Confidential document only available on request to JIWG.	ISCI-v2.2 2013- January
PP		
RD [8]	Security IC Platform Protection Profile	BSI-PP-0035-2007
IC		



RD [9]	Certification report, Infineon smartcard IC (Security Controller) M7794 A12 and G12 with optional RSA2048/4096 v1.02.013 EC v1.02.013 and Toolbox v1.02.013 30 September 2013	BSI-DSZ-CC-0883- 2013
RD [10]	Security Target Lite of M7794 A12 and G12, Version 1.3, 2013-08-09	BSI-DSZ-CC-0883
RD [11]	M7794 Security Guidelines, User's Manual 2013-08-06 Confidential document only available on request to Infineon (NDA is required).	M7794_SecurityGuidel ines-2013-08-06
RD [12]	M7794 Hardware Reference Manual Rev 1.1, 2011-11-02 Confidential document only available on request to Infineon (NDA is required).	M7794_ HwRefManual_2011- 11
ISO		
RD [13]	Information technology – Identification cards – Integrated circuit(s) cards with contacts	ISO7816
RD [14]	Contactless integrated circuit(s) cards	ISO14443
JC/GP		
RD [15]	Java Card™ 2.2.1	
RD [16]	GlobalPlatform - Card Spec v2.1.1	
Gemalto		
RD [17]	Functional Specification – CPU Card	CPU_FS_ECC
RD [18]	Functional Specification – SAM Card	SAM_FS_ECC

Table 1 – Reference list



1.3 Terminologie

See Annex section [8.1](#).

1.4 Conventions

Blue mark	hyperlink
Red mark	field to be updated

TBC	To Be Completed
TBD	To Be Defined
NA	Not Applicable



2 ST Introduction

2.1 ST reference, TOE reference

2.1.1 ST reference

Title: **Security Target Lite ECC CPU card**
Reference: **R0R23928_CCD_ASE_002**
Revision: **1.3p**
Author: Gemalto

2.1.2 TOE reference

Product name: **ECC CPU card**
TOE name: **CPU e-purse on GFCX6-EC**
TOE version: **V1.2 on G256**
TOE documentation¹: **R0R23928_CCD_AGD_006**
HW part of TOE: **M7794 A12 AND G12 (BSI-DSZ-CC-0883)**

2.2 Overview

2.2.1 ST overview

The aim of this document is to describe the target of evaluation for the product **ECC CPU card** (section [2.2.2.2](#)) and for the TOE (section [2.2.3](#)).

This document provides:

- The TOE description (section [2.3](#))
- The security problem definition: assets, threats, Organizational Security Policies (OSPs), assumptions (section [3](#))
- The description of security objectives for the TOE and for the operational environment (section [5](#))
- The description of TOE security functions and assurance measures if any (section [7](#))

¹ TOE documentation is guidance for user and administrator.



2.2.2 Product overview

2.2.2.1 Architecture of Easy Card Corporation system

The architecture of Easy Card Corporation system is given in the following figure:

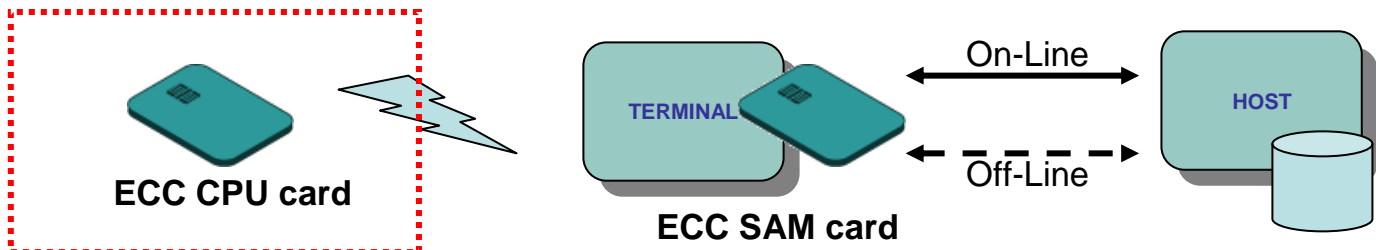


Figure 1 – Architecture of Easy Card Corporation System

The Easy Card Corporation system is a payment system (Electronic Money (EM) system) intended for low value off-line payment transactions. The global functioning of the EM system is based on two cycles:

- a first one consisting in EM creation,
- a second one consisting in payment in counterpart of goods or services,

EM is the counterpart of funds received by the EM issuer. It is defined by the identity of the EM issuer, the currency denomination and the amount: Electronic Purse (EP).

To simplify, these cycles can be summarized as following:

- the Purseholder gives funds to the EM issuer who loads his EP with an equivalent amount of EM (**Credit and Auto-Load transactions**)
- the Purseholder asks the merchant for a service and transfers EM from his EP to the SAM (**EM Payment transactions, corresponding to a Debit operation for the EP**).

During each transaction (Credit/Auto-Load, Debit), EM circulates within a closed loop system. Only the EM issuer is authorized to create or extinguish EM. Furthermore, the EM credited on the one hand should be always equal to the EM debited on the other hand.

2.2.2.1.1 ECC CPU Card

The ECC CPU card is the EP device. An EP is an application executed by an OS embedded into an IC. Its functionalities are similar to traditional purse functionalities with the distinction that it uses EM instead of cash money. An EP is used to facilitate payments of low value. The fully operational EP contains various parameters that could be updated.

The EP is the TOE of the present ST, and is compliant with the specifications given in [\[17\]](#).

As an EP in EM system, the TOE is able to:

- Store its amount of EM which defines the balance of the EP,
- Indicate amount of EM via **Read Purse** command,
- Debit its amount of EM via **Debit Purse** command,
- Credit its amount of EM via **Credit² Purse** command,

² The Credit and Auto-Load transactions are options of the **Credit Purse** command.



- Update parameters via **Put Data** command,

The ECC CPU card primary functionality is to allow the Purseholder to make EM payment in a simple, secure and fast way.

The ECC CPU card services are:

- EM protection in term of integrity during **Credit, Auto-Load** and **Debit** operations.
- Security assets protection in term of integrity and confidentiality when used or stored.
- Mutual authentication between the TOE and the ECC SAM card during **Auto-Load** and **Debit** operations.
- Mutual authentication between the TOE and the Host device during **Credit** operations.
- Invalidation (i.e. de-activation) of the card via **Write Lock** command
- File management commands

2.2.2.1.2 ECC SAM Card

The purchase device is a physical device installed at the merchant or a server used to accept payment from an EP in an EM payment transaction. It includes a Secure Access Module (SAM) - the ECC SAM card -, built on an IC module. The ECC SAM card is compliant with the specifications given in [18].

The ECC SAM card shall provide the necessary security for the EM payment. It contains various parameters that could be updated.

2.2.2.1.3 Easy Card Corporation EM system transactions

- **Credit**

The EP is credited with an amount of EM created by the EM issuer. The Purseholder gives a corresponding amount of funds in turn.

- **Auto-Load**

This operation is used to load EM into the EP. It is processed off-line. The SAM processes this operation. It may be performed when the purchase amount is greater than the balance of the purse in order to go on with the EM payment.

- **Debit (EM Payment)**

The EP is debited from an amount of EM while the SAM is credited with the same amount of EM. The purse holder receives goods or services in turn. The EM payment transaction is presented as a debit operation when it's only related to the EP.

- **Parameters update**

Internal EP or SAM parameters are updated by a distant server. Parameters that are addressed are, for instance, the expense limit per transaction, the transaction keys.

2.2.2.2 The product ECC CPU Card

The card contents an integrated circuit (IC) and an operating system (OS) where the purse application is loaded.

The **ECC CPU card** is Java card smart card either i) in contactless or ii) in dual interfaces, which means that the services described above are accessible either in contactless mode (based on [14]) only for i) or in contact (based on [13]) or in contactless mode for ii).

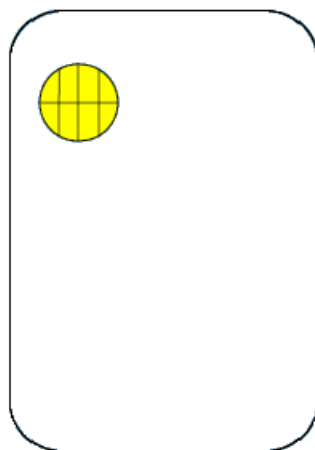


Figure 2 – ECC CPU Card CTL or DUAL

The architecture is based on Java card architecture that contains the following components:

- The Java card kernel that provides a securized framework to Java card application execution and the access management to data securely. The Java Card Runtime Environment, which provides a secure framework for the execution of Java Card programs and data access management (firewall). The Java Card Virtual Machine, which provides the secure interpretation of bytecodes. The APIs. The Open Platform Card Manager, which provides card, key and application management functions (contents and life-cycle) and security control.
- The Native part that provides the basic functionalities to the card (memory management, I/O management and cryptographic primitives) with native interface with the dedicated IC. The cryptographic features implemented in the native layer are SW AES (using the HW AES) and SW RNG (using the HW RNG).

This product uses the Java card technology and provides Java card services to the purse application.

The APDU commands of the **ECC CPU card** are given in the document [\[17\]](#).

- File management commands: Get Data, External Authenticate, Read Record, Update Record, Append Record, Change Key-App Admin
- Payment commands: Read Purse, Initiate Processing, Debit Purse, Credit Purse
- Purse management commands: Put Data, Write Lock

The **ECC CPU card** architecture is given in the next figure.

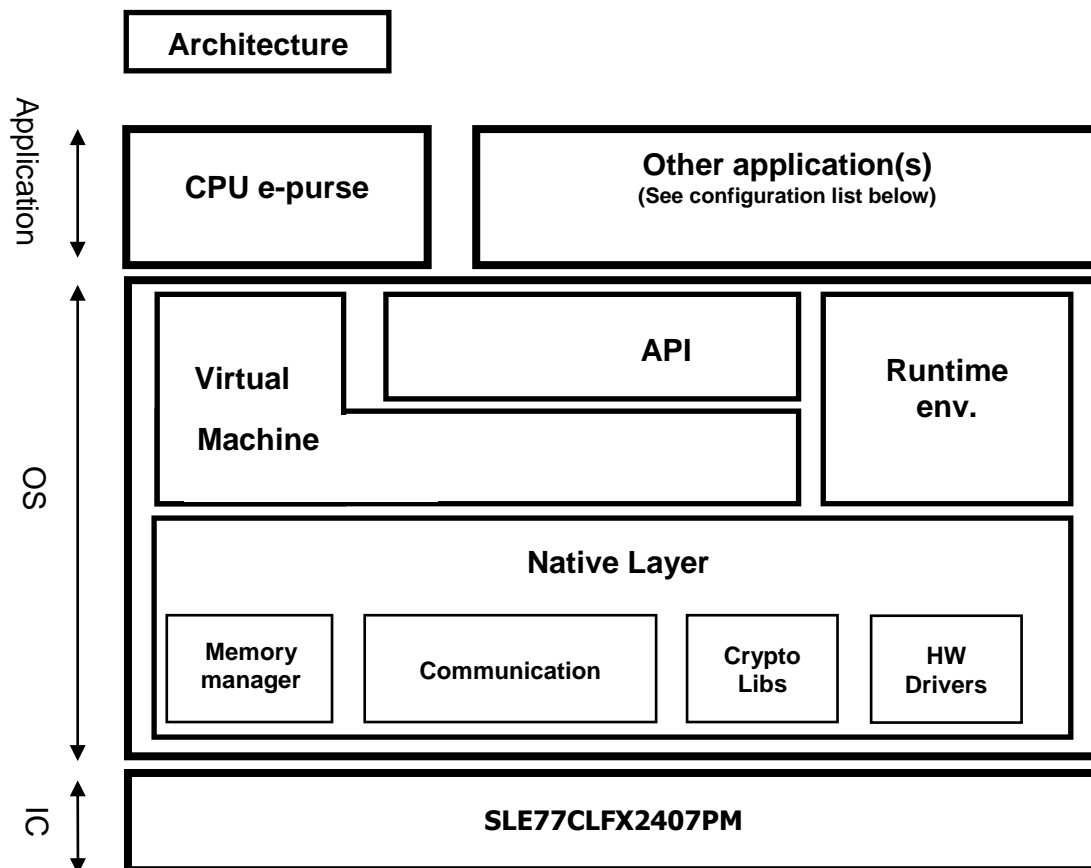


Figure 3 – Architecture of the ECC CPU Card

The **ECC CPU card** characteristics are:

- Conform to component characteristics: [\[10\]](#) [\[11\]](#) [\[12\]](#)
- Conform to ISO 7816 Parts 1-9: [\[13\]](#)
- Conform to ISO 14443: [\[14\]](#)
- Conforms to Sun's Java Card 2.2.1: [\[15\]](#)
- Conforms to Global Platform Card Specification version 2.1.1: [\[16\]](#)
- Conform to specifications **ECC CPU card**: [\[17\]](#)

Applications in the environment could be (depend on the ECC CPU card configurations (see below)):

- MChipPaypass , V 2.E.14D, MasterCard Dual Interface EMV payments applet,
- JCB JSmart V3.0, JCB Dual Interface EMV payments applet,
- VSDC V2.8.1.F1, VISA Dual Interface EMV payments applet,
- PBOC, V3.0, China Union Pay Dual Interface EMV payments applet,
- Dual PSE 1.2 (PSE/PPSE Payment System Environment / Proximity Payment System Environment) application
- MPH110 also named MOC v1.0 transport application
- FISC v1.1 ATM application

ECC CPU card configuration should be:

Configuration 1: CPU e-Purse

Configuration 2: CPU e-Purse + DualPSE + VSDC281F1



Configuration 3: CPU e-Purse + DualPSE + MChipPaypassV2E14D

Configuration 4: CPU e-Purse + DualPSE + J/Smart V3.0 + MChipPaypassV2E14D

Configuration 5: CPU e-Purse + DualPSE + PBOCV3.0

Configuration 6: CPU e-Purse + MOC

Configuration 7: CPU e-Purse + DualPSE + VSDC281F1+ FISC

Configuration 8: CPU e-Purse + DualPSE + MChipPaypassV2E14D+ FISC

Configuration 9: CPU e-Purse + DualPSE + J/Smart V3.0 + MChipPaypassV2E14D+ FISC

Configuration 10: CPU e-Purse + DualPSE + PBOCV3.0+ FISC

2.2.3 TOE overview

The TOE is the **CPU e-purse on GFCX6-EC** application.

The TOE is applied to:

- IC: **M7794 (BSI-DSZ-CC-0883)**
- Code identifier: GFCX6-EC/G256
- The associated documentation: **R0R23928_CCD_AGD_006**

Note: The IC is evaluated and certified (see [9]) then the security requirements specifically applying to the IC [10] are not reproduced in this document.

2.3 TOE description

Legend:

- The TOE limit is represented by red dotted line (- - -).
- All are inside the limit is the TOE part.
- All are outside the limit is the TOE environment part.

The TOE is given in the following figures:

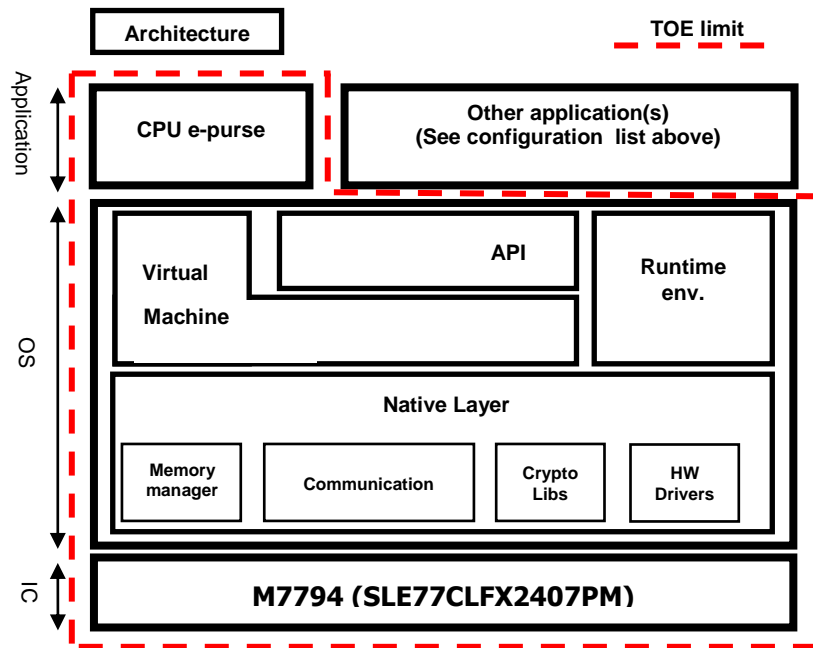


Figure 4 – TOE description (architecture)

2.3.1 Physical scope of the TOE

The elements in the scope of the TOE are:

- The **CPU e-purse on GFCX6-EC** application written in java language
- The OS composed of:
 - Java card kernel:
 - **Runtime environment** written in native language,
 - **Virtual machine** written in native language,
 - **APIs** written in native and java language
 - Native part:
 - Memory management (**Memory manager**) written in native language,
 - Communication management (**Communication**) written in native language,
 - Cryptographic libraries management (and the cryptographic libraries itself) (**Crypto libs**) written in native language,
- The IC **M7794**

The elements out of scope of the TOE are:

- Other application(s) written in java language

2.3.2 Logical scope of the TOE

The TOE provides the following services:

- The **CPU e-purse on GFCX6-EC** application services
- The platform services:
 - Initialization of the Card Manager and management of the card life cycle,
 - Management and control of the communication between the card and the CAD or PCD,
 - Secure installation of the applets under Card Manager control,
 - Deletion of applications under Card Manager control,



- Secure operation of the applications through the API,
- Card basic security services (Checking environmental operating conditions using information provided by the IC, Checking life cycle consistency, Ensuring the security of the assets, Generating random number, Handling secure data object and backup mechanisms, Managing memory content, Ensuring Java Card firewall mechanism, ...).

The **CPU e-purse on GFCX6-EC** application and OS is loaded in flash during phase 4;



2.3.3 TOE life-cycle

Legend:

- The TOE limit is represented by red dotted line (- - -).
- All are inside the limit is covered by the development life-cycle assurance.
- All are outside the limit is covered by the guidance assurance.

2.3.3.1 ECC CPU life-cycle

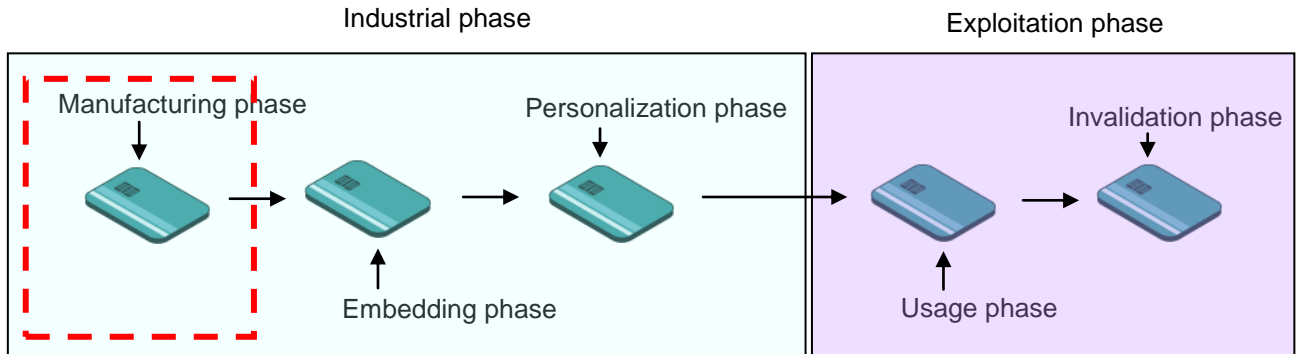


Figure 5 – Easy Card Corporation life-cycle

Phase (name)	Phase (card)	Actor	Comment
Manufacturing	1 OS Development	Developer (Gemalto)	- Development of e-purse application and Java card platform - Generation of .HEX -Script development
	2 HW Development	IC manufacturer (Infineon)	- Development of IC
	3 Mask manufacturing	IC manufacturer (Infineon)	Manufacturing of virgin M7794 integrated circuits embedding the Infineon flash loader, and protected by a dedicated transport key.
	4 Module manufacturing & initialization	Module with embedded software (Gemalto)	IC packaging & testing Loading of the Gemalto GFCX6-EC software (platform and applets on top of it) Set card life-cycle to "OP_READY" phase
Embedding	5 Card embedding	Card manufacturer (Gemalto or other)	Embedding
Personalization	6 Personalization	Personalizer (Easy Card Corporation)	- Personalization - Set card life-cycle to « SECURED»
Usage	7 Invalidation	Issuer (Easy Card Corporation) Purseholder (customer of Issuer)	- The Issuer is responsible of card delivery to the end-user (Purseholder) - Usage of the card by the Purseholder - The Issuer is responsible of card end-of-life process
Invalidation	8	Issuer	- Invalidation by the Issuer



Phase (name)	Phase (card)	Actor	Comment
		(Easy Card Corporation)	- Set card life-cycle to « CARD_LOCK or TERMINATED»

Table 2 – ECC CPU life-cycle

2.3.3.2 Actors

Developer (Phase 1)	Gemalto Roles: <ul style="list-style-type: none"> Develop the smart card OS Develop the application
IC manufacturer (Phases 2 & 3)	Infineon Roles: <ul style="list-style-type: none"> Develop the IC Manufacture the IC
Card manufacturer U.SC_MNF (Phases 4 & 5)	Phase 4: Gemalto Roles: <ul style="list-style-type: none"> Module manufacturing Initialize/Pre-personalize the module Phase 5: Gemalto or other Role: <ul style="list-style-type: none"> Embed the module on the card
Personalizer U.SC_PER (Phase 6)	Easy Card Corporation Role: <ul style="list-style-type: none"> Personalize the card
Issuer U.ISSUER (Phase 7 & 8)	He is the Electronic Money (EM) issuer that guarantees the EM in the EM system. Easy Card Corporation Roles: <ul style="list-style-type: none"> Emit the card Invalid the card
Purseholder U.PURSEHOLDER (Phase 7)	He is the person that is in possession of the Electronic Purse (EP) and uses it for EM payment transactions. Role: <ul style="list-style-type: none"> Use the e-purse application in the CPU Card

Table 3 – Actors



2.3.3.3 Environments

Development	<p><u>Phase 1</u>: Limited to Gemalto site For Gemalto:</p> <ul style="list-style-type: none"> Gemalto Meudon (R&D OS) 6 rue de la Verrerie / 92197 Meudon Cedex, France Gemalto Singapore (R&D application) 12 Ayer Rajah Crescent / 139941, Singapore Gemalto Gemenos (Masking) Avenue du Pic de Bertagne – BP 100 / 13881 Gémenos Cedex, France <p><u>Phase 2</u>: Limited to Infineon sites For Infineon:</p> <ul style="list-style-type: none"> Infineon Semiconductors GmbH see [9]
Production	<p><u>Phase 3</u>: Limited to Infineon site For Infineon:</p> <ul style="list-style-type: none"> Infineon Semiconductors GmbH see [9] <p><u>Phase 4 & 5</u>: Limited to Gemalto sites</p> <ul style="list-style-type: none"> Phase 4: Modules are provided by IC or Card manufacturer Phase 5: Cards are provided by Card manufacturer <p>For Gemalto:</p> <ul style="list-style-type: none"> Gemalto Singapore (Production) 12 Ayer Rajah Crescent / 139941, Singapore Gemalto Gemenos (Production) Avenue du Pic de Bertagne - BP 100/ 13881 Gémenos Cedex, France
Personalization	Phase 6: Limited to Personalization site (Easy Card Corporation)
Usage	Phase 7: by End-user
Invalidation	Phase 8: by Issuer

Table 4 – Environnements

TOE development environment

- Phase 1 :

To assure security, the environment in which the development takes place must be made secure with controllable accesses having traceability. Furthermore, it is important that all authorized personnel involved fully understand the importance and the rigid implementation of defined security procedures.

- Phase 2 :

The phase 2 environment is described in the PP [8].

TOE production environment

- Phase 3 :

The phase 3 environment is described in the PP [8].

- Phases 4 and 5:

If the phase 4 is managed by the IC manufacturer then the environment is described in the PP [8].

Else the environment is managed by the Card manufacturer. To assure security, the environment in which the development takes place must be made secure with controllable accesses having traceability. Furthermore, it is important that all authorized personnel involved fully understand the importance and the rigid implementation of defined security procedures.

TOE personalization environment

- Phase 6:





The phase 6 environment is managed by the Personalizer. To assure security, the environment in which the development takes place must be made secure with controllable accesses having traceability. Furthermore, it is important that all authorized personnel involved fully understand the importance and the rigid implementation of defined security procedures.

Usage environment

- Phase 7:

The phase 7 environment is non-secured environment.

Invalidation environment

- Phase 8:

The phase 8 environment is non-secured environment.

2.3.3.4 Platform and applet states

In accordance with [16] specifications, the platform and the application enforce the life-cycle states mentioned in [Table 2 – ECC CPU life-cycle](#).

Phase		Platform state	Application state
1	SC software development		
2	IC development		
3	IC manufacturing	OS_NATIF	
4	SC manufacturing & pre-personalization	OS_NATIF OP_READY INITIALIZED	
5	SC Embedding	INITIALIZED	
6	SC personalization	SECURED	INSTALLED SELECTABLE PERSONALIZED
7 & 8	SC usage for Issuer and Purseholder	SECURED CARD_LOCKED TERMINATED	PERSONALIZED BLOCKED

Table 5 – Platform and application states



3 Conformance claims

Common criteria Version:

This ST conforms to CC Version 3.1 R4 [1] [2] [3].

Conformance to CC part 2 and 3:

This ST is CC part 2 conformant [2]. It means that all SFRs in that ST are based only upon functional components in CC part 2.

This ST is CC part 3 conformant [3]. It means that all SARs in that ST are based only upon assurance components in CC part 3.

Assurance package conformance: EAL4 augmented (EAL4+)

This ST conforms to the assurance package EAL4 augmented by ALC_DVS.2 and AVA_VAN.5.

Evaluation type

This is a composite evaluation, which relies on the **M7794** chip certificate and evaluation results.

- Certification done under the BSI scheme
- Certification report **BSI-DSZ-CC-0883**
- Security Target [10] strictly conformant to IC Protection Profile [8]
- Common criteria version: 3.1
- Assurance level: EAL5 augmented (EAL5+) by ASE_TSS.2, ALC_DVS.2 and AVA_VAN.5

Consequently, the composite product evaluation (i.e. the present evaluation) includes the additional composition tasks defined in the CC supporting document "Composite product evaluation for smart cards and similar devices" [5].

Protection Profile (PP) conformance claims:

This ST doesn't conform to any Protection Profile.

Note: The ST writer uses the Moneo Electronic Purse Protection Profile (ref: Moneo – Electronic Purse Protection Profile, Ref SFPMEI-CC-PP-EP, Version 1.5, BMS/SFPMEI, February 4th 2010).



4 Security Problem Definition

4.1 Assets

Assets protected by the TOE (considered as User data)	
Name/Description	Protection
D.EM Electronic Money (EM). For the Electronic Purse (EP), EM is an electronic substitute of funds received by the EM issuer. It is represented by the balance of the purse.	Integrity
D.EP_IVDATA The EP identification and validity data includes the Purse Serial Number (PSN), which is a unique sequence of numbers assigned to the EP used for identification purposes, and validity data that allows detecting EP end-of-life.	Integrity
Sensitive assets of the TOE (considered as TSF data)	
Name/Description	Protection
D.EP_CODE The application code embedded in the EP. Both the IC and the TOE shall contribute to the correct execution and integrity of the EP application code. The IC shall also protect the confidentiality of the asset.	Correct execution Integrity Confidentiality
D.KEYS The EP secret keys used for authentication purposes.	Integrity Confidentiality
D.EP_STATE The state of the EP stores information about the EP internal states during its usage phase. The behavior of the EP is modeled using a state machine. A state machine is composed of states defining authorized operations and transitions from one state to another. Transitions are usually triggered by direct or indirect activation of the device inputs (for instance the receipt of a transaction).	Integrity
D.LOG_DATA Flow traceability data stands for information on the last Debit Purse transactions stored in log files.	Integrity
D.COUNTERS The sequence counters count the successive transactions: <ul style="list-style-type: none"> • Credit and Auto-Load operations, • Debit operations. Static counters cover the configuration (parameters) of the EP: <ul style="list-style-type: none"> • the maximum amount of EM stored into the EP, • the EM maximum amount of Debit Purse operations, • the Credit Purse operations amount, • the maximum number of consecutive Auto-Load transactions. 	Integrity

Table 6 – Assets



4.2 Threats

4.2.1 Threats agents

Name	Description
U.ATTACKER	Any human or machine who attempts to passively and/or actively violate the security and behavior of the TOE by: <ul style="list-style-type: none"> Interception, modification or permutation of the messages exchanged between the TOE and any external user or subject Modification or perturbation of card behavior Modification or determination of card data
U.APP	Any software in the on-card environment of the TOE that may attempt to attack the TOE.

Table 7 – Threats agents

4.2.2 Threats

Threat family	General description
COUNTERFEITING	The creation of fake transactions by falsification of assets in order to create or lose EM.
DISCLOSURE	Unauthorized disclosure of assets.
LOSS OF INTEGRITY	Unauthorized modification of assets.
REPLAY	Replay of a previous transaction or the last transaction. Such a replay can be performed immediately or several times after the first send of the transaction.
Other	Other type of threat dedicated to the OS (platform).

Threat family: COUNTERFEITING	
Name/Description	Asset
T.COUNTERFEITING_DEBIT Counterfeiting of a debit operation in order to debit the EP with an EM greater or lesser than the transaction EM amount; it leads to EM creation or loss.	D.EM
T.COUNTERFEITING_CREDIT Counterfeiting of a Credit transaction in order to credit the EP without any financial counterpart; it leads to unauthorized EM creation or loss.	D.EM
T.COUNTERFEITING_AUTOLOAD Counterfeiting of a Auto-Load transaction in order to credit the EP with an EM greater or lesser than the EM amount specified in the transaction; it leads to unauthorized EM creation or loss.	D.EM
T.COUNTERFEITING_AUTH Counterfeiting of parameters in order to bypass the authentication protocol (sequence counter).	D.COUNTERS
T.COUNTERFEITING_UPDATE Counterfeiting of parameters update transaction in order to change the keys values or static counters values.	D.KEYS D.COUNTERS
Threat family: DISCLOSURE	
Name/Description	Asset
T.DISCLOSURE_KEYS Unauthorized access of the secret keys.	D.KEYS
Threat family: LOSS OF INTEGRITY	
Name/Description	Asset



T.INTEG_EM Unauthorized modification of stored EM: An attacker modifies the amount of EM stored in the EP in order to input a greater or lower amount.	D.EM
T.INTEG_EP_IVDATA Unauthorized modification of stored EP identification and validity data: an attacker modifies the value of the EP identification and validity data stored in the EP in order to input another one.	D.EP_IVDATA
T.INTEG_CODE Unauthorized modification of the TOE code: an attacker modifies the code in order to bypass the security policy of the EP.	D.EP_CODE
T.INTEG_KEYS Unauthorized modification of stored keys: an attacker modifies the value of the secret keys and associated attributes stored in the EP in order to input a known key.	D.KEYS
T.INTEG_EP_STATE Unauthorized modification of the State Machine: an attacker modifies or deletes information that defines the current state of the EP in order, for instance, to bypass a secure state.	D.EP_STATE
T.INTEG_LOG_DATA Unauthorized modification of the stored flow traceability data: an attacker modifies the log of the last Debit Purse transactions in order to hide potentially malicious operations performed on the EP.	D.LOG_DATA
T.INTEG_COUNTERS Unauthorized modification of stored sequence counters : an attacker modifies the value of sequence counters in order to force the EP accepting counterfeited or replayed transactions. Unauthorized modification of stored static counters : an attacker modifies the value of static counters which define the configuration of the EP in order to bypass controls or limitations enforced by the EM system.	D.COUNTERS
Threat family: REPLAY	
Name/Description	Asset
T.REPLAY_DEBIT Replay of a Debit: an EP is debited several times via a previous complete sequence of operations (Debit operation); it leads to EM loss.	D.EM
T.REPLAY_CREDIT Replay of a Credit transaction: an EP is loaded several times via a previous complete sequence of operations (Credit transaction); it leads to unauthorized EM creation.	D.EM
T.REPLAY_AUTOLOAD Replay of a AutoLoad transaction: an EP is loaded several times via a previous complete sequence of operations (AutoLoad transaction); it leads to unauthorized EM creation.	D.EM
T.REPLAY_UPDATE Replay of a parameters update transaction: an EP is updated several times via a previous complete sequence of operations (parameters update transaction); it leads to fraudulent changes of parameters stored in the EP (keys and static counters).	D.KEYS D.COUNTERS
Threat family: Other	
Name/Description	Asset
T.Separation Tamper the product through the Application Layer thus bypassing the TOE API (services) to access or modify the OS code or data.	All

Table 8 – Threats



4.3 Organizational security policies (OSPs)

Name / Description	Asset
<p>OSP.SECRET_MNGT Management of secret User/TSF data (e.g. generation, storage, distribution, destruction, loading into the product of cryptographic keys) performed outside the product on behalf of the TOE or SC Manufacturer shall comply with security organizational policies that enforce integrity and confidentiality of these data.</p> <p>Secret data shared with the user of the product shall be exchanged through trusted channels that protect the data against unauthorized disclosure and modification and allow detecting potential security violations.</p>	All
<p>OSP.DEBIT_BEFORE_CREDIT Debit from the EP always precedes credit of the SAM during a payment transaction.</p>	D.EM
<p>OSP.PURSE_BEHAVIOR The Purseholder shall keep the EP as a real purse with money and bank notes and not loan it, especially to untrusted persons.</p>	D.EM

Table 9 – Organizational security policies (OSPs)

4.4 Assumptions

Name / Description	Asset
<p>A.PHYSICAL It is assumed that the Security IC is the certified M7794 A12 and G12 (SLE77CLFX2407PM) (BSI-DSZ-CC-0883-2013).</p>	All
<p>A.PROTECTION_AFTER_TOE_DELIVERY It is assumed that the persons manipulating the TOE in the operational environment follow the TOE guides (user and administrator guidance of the product, installation documentation and personalization guide). It is also assumed that the persons responsible for the application of the procedures contained in the guides, and the persons involved in delivery and protection of the product have the required skills and are aware of the security issues.</p> <p><i>Note:</i> The TOE certificate is valid only when the guides are applied. For instance, for pre-personalization or personalization guides, only the described set-up configurations or personalization profiles are covered by the certificate; any divergence would not be covered by the certificate.</p>	All

Table 10 – Assumptions

4.5 Composition tasks – Security problem definition part

The objective is to determine if the composite-ST (this ST) doesn't contradict the platform-ST [\[10\]](#) in term of threats ([Threats](#)), organizational security policy ([OSP](#)) and assumptions ([Assumptions](#)).



4.5.1 Statement of Compatibility – Threats part

IC threat label	IC threat title	IC threat content	Link to the composite-product
T.Leak-Inherent	Inherent Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets. No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements.	T.DISCLOSURE_KEYS
T.Phys-Probing	Physical Probing	An attacker may perform physical probing of the TOE in order (i) to disclose User Data (ii) to disclose/reconstruct the Security IC Embedded Software or (iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.	T.DISCLOSURE_KEYS
T.Malfunction	Malfunction due to Environmental Stress	An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to (i) modify security services of the TOE or (ii) modify functions of the Security IC Embedded Software (iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software. This may be achieved by operating the Security IC outside the normal operating conditions.	All T.COUNTERFEITING T.DISCLOSURE_KEYS All T.INTEG All T.REPLAY
T.Phys-Manipulation	Physical Manipulation	An attacker may physically modify the Security IC in order to (i) modify User Data (ii) modify the Security IC Embedded Software (iii) modify or deactivate security services of the TOE, or	All T.COUNTERFEITING T.DISCLOSURE_KEYS All T.INTEG



		(iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.	
T.Leak-Forced	Forced Information Leakage	An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets even if the information leakage is not inherent but caused by the attacker.	T.DISCLOSURE_KEYS
T.Abuse-Func	Abuse of Functionality	An attacker may use functions of the TOE which may not be used after TOE Delivery in order to (i) disclose or manipulate User Data (ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or (iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or (iv) enable an attack disclosing or manipulating the User Data or the Security IC Embedded Software.	T.DISCLOSURE_KEYS All T.INTEG T.Separation
T.RND	Deficiency of Random Numbers	An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.	All T.REPLAY
T.Mem-Access	Memory Access Violation	Parts of the Smartcard Embedded Software may cause security violations by accidentally or deliberately accessing restricted data (which may include code) or privilege levels. Any restrictions are defined by the security policy of the specific application context and must be implemented by the Smartcard Embedded Software.	T.DISCLOSURE_KEYS All T.INTEG T.Separation

Table 11 – Composition – Threats part

4.5.2 Statement of Compatibility – OSP part

IC OSP label	IC OSP content	Link to the composite-product
P.Process-TOE	Protection during TOE Development and Production: An accurate identification must be established for the	No contradiction with the present evaluation; the chip traceability information is used to identify the composite TOE.



	TOE. This requires that each instantiation of the TOE carries this unique identification.	
P.Add-Functions	<p>Additional Specific Security Functionality: The TOE shall provide the following specific security functionality to the Smartcard Embedded Software:</p> <ul style="list-style-type: none"> • Advanced Encryption Standard (AES) • Triple Data Encryption Standard (3DES) • Rivest-Shamir-Adleman Cryptography (RSA) • Elliptic Curve Cryptography (EC) 	The AES hardware functionality is used by the composite TOE. The 3DES, RSA and EC hardware functionalities are not used by the composite TOE.

Table 12 – Composition – OSPs part

4.5.3 Statement of Compatibility – Assumptions part

IC assumption label	IC assumption title	IC assumption content	IrPA ³	CfPA ⁴	SgPA ⁵	Link to the composite-product
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation	It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).			X	A.PROTECTION_AFTER_TOE_DELIVERY
A.Plat-Appl	Usage of Hardware Platform	The Security IC Embedded Software is designed so that the requirements from the following documents are met: (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the hardware data sheet, and the hardware		X		Fulfilled by the composite-SAR ADV_COMP.1 (cf [CCDB], Appendix 1.2, §72 and §73)

³ IrPA means "The assumptions being not relevant for the Composite-ST, e.g. the assumptions about the developing and manufacturing phases of the platform."

⁴ CfPA means "The assumptions being fulfilled by the Composite-ST automatically. Such assumptions of the Platform-ST can always be assigned to the TOE security objectives of the Composite-ST. Due to this fact they will be fulfilled either by the Composite-TSF or by the Composite-TAM automatically."

⁵ SgPA means "The remaining assumptions of the Platform-ST belonging neither to the group IrPA nor CfPA. Exactly this group makes up the significant assumptions for the Composite-ST, which shall be included into the Composite-ST."



		application notes, and (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report.				
A.Resp-Appl	Treatment of User Data	All User Data are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.		X		O.AUTH O.EM O.OPERATE O.TAMPER O.REPLAY
A.Key-Function	Usage of key-dependent functions	Key-dependent functions (if any) shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced). Note that here the routines which may compromise keys when being executed are part of the Smartcard Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines which are part of the TOE		X		O.AUTH

Table 13 – Composition – Assumptions part



5 Security Objectives

5.1 Security objectives for the TOE

Name / Description	Threat
O.AUTH The TOE shall enforce mutual authentication with external devices (load device, SAM) for all the performed transactions.	T.COUNTERFEITING_DEBIT T.COUNTERFEITING_CREDIT T.COUNTERFEITING_AUTOLOAD T.COUNTERFEITING_AUTH T.COUNTERFEITING_UPDATE
O.EM The TOE shall prevent unauthorized creation or loss of EM.	T.COUNTERFEITING_DEBIT T.COUNTERFEITING_CREDIT T.COUNTERFEITING_AUTOLOAD T.INTEG_EM T.REPLAY_DEBIT T.REPLAY_CREDIT T.REPLAY_AUTOLOAD
O.CONF_DATA The TOE shall prevent unauthorized disclosure of TSF data.	T.DISCLOSURE_KEYS
O.INTEG_DATA The TOE shall prevent unauthorized modification of User and TSF data.	T.INTEG_EM T.INTEG_IVDATA T.INTEG_CODE T.INTEG_KEYS T.INTEG_EP_STATE T.INTEG_LOG_DATA T.INTEG_COUNTERS
O.OPERATE The TOE shall ensure the continued correct operation of its security functions in case of abnormal transactions and unexpected interruption.	T.INTEG_EM T.INTEG_IVDATA T.INTEG_CODE T.INTEG_KEYS T.INTEG_EP_STATE T.INTEG_LOG_DATA T.INTEG_COUNTERS
O.REPLAY The TOE shall detect and reject replayed transactions.	T.REPLAY_DEBIT T.REPLAY_CREDIT T.REPLAY_AUTOLOAD T.REPLAY_UPDATE
O.TAMPER The TOE shall prevent physical tampering of its security critical parts. The TOE shall monitor security registers and system flags made available by the IC and respond to potential security violations in a way that preserves a secure state.	T.DISCLOSURE_KEYS T.INTEG_EM T.INTEG_IVDATA T.INTEG_CODE T.INTEG_KEYS T.INTEG_EP_STATE T.INTEG_LOG_DATA T.INTEG_COUNTERS
O.RECORD The TOE shall record the last Debit Purse transactions to support effective security management.	T.COUNTERFEITING_DEBIT T.COUNTERFEITING_CREDIT T.COUNTERFEITING_AUTOLOAD T.INTEG_LOG_DATA
O.Atomicity The TOE shall provide a means to perform memory operation atomically.	All threats
O.Separation	T.DISCLOSURE_KEYS



<p>The TOE shall provide a separation mechanism between itself and Application layer, and control the access to its service and resources.</p>	<p>T.INTEG_EM T.INTEG_IVDATA T.INTEG_CODE T.INTEG_KEYS T.INTEG_EP_STATE T.INTEG_LOG_DATA T.INTEG_COUNTERS T.Separation</p>
--	--

Table 14 – Security objectives of the TOE

5.2 Security objectives for the operational environment

Name / Description	OSP/Assumption
<p>OE.SECRET_MNGT The secret User/TSF data managed outside the TOE shall be protected against unauthorized disclosure and modification.</p>	OSP.SECRET_MNGT
<p>OE.DEBIT_BEFORE_CREDIT Debit must always precede credit during EM payment transaction.</p>	OSP.DEBIT_BEFORE_CREDIT
<p>OE.PURSE_BEHAVIOR The EM issuer shall communicate to the Purse holder the rules dealing with the use of the EP. Especially it must inform the user to keep EP the same way he does for a real purse. The Purseholder shall enforce these rules.</p>	OSP.PURSE_BEHAVIOR
<p>OE.PHYSICAL The Security IC shall detect and respond to invasive physical attacks, to environmental stress and to attempts to access Security IC unauthorized functionality. The Security IC shall prevent leakage of information. The Security IC shall manage its life cycle states and transitions between them; in particular the Security IC shall not allow Test Mode functions once the Security IC has entered the User Mode. The Security IC security features shall resist to high attack potential as defined in [6] and [7]. A Security IC that complies with [8] meets this objective.</p>	A.PHYSICAL
<p>OE.PROTECTION_AFTER_TOE_DELIVERY Procedures and controlled environment shall ensure protection of the TOE and related information after delivery. Procedures shall ensure that people involved in TOE delivery and protection have the required skills. The persons using the TOE in the operational environment shall apply the TOE guides (user and administrator guidance of the product, installation documentation and personalization guide).</p>	A.PROTECTION_AFTER_TOE_DELIVERY

Table 15 – Security objectives of the operational environment

5.3 Composition tasks – Security objective part

The objective is to determine if the composite-ST (this ST) doesn't contradict the platform-ST [10] in term of security objectives for the TOE ([Security objective for the TOE](#)) and for the environment ([Security objective for the operational environment](#)).



5.3.1 Statement of Compatibility – Security objectives for the TOE part

IC TOE security objective Label	IC TOE security objective Title	IC TOE security objective Content	Link to the composite-product
O.Leak-Inherent	Protection against Inherent Information Leakage	<p>The TOE must provide protection against disclosure of confidential data stored and/or processed in the Security IC</p> <ul style="list-style-type: none"> - by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and - by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines). <p>This objective pertains to measurements with subsequent complex signal processing whereas O.Phys-Probing is about direct measurements on elements on the chip surface. Details correspond to an analysis of attack scenarios which is not given here.</p>	O.TAMPER O.CONF_DATA
O.Phys-Probing	Protection against Physical Probing	<p>The TOE must provide protection against disclosure of User Data, against the disclosure/reconstruction of the Security IC Embedded Software or against the disclosure of other critical information about the operation of the TOE. This includes protection against</p> <ul style="list-style-type: none"> - measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or - measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis) with a prior reverse-engineering to understand the design and its properties and functions. 	O.TAMPER O.CONF_DATA
O.Malfunction	Protection against Malfunctions	<p>The TOE must ensure its correct operation.</p> <p>The TOE must indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields.</p>	O.OPERATE O.REPLAY
O.Phys-Manipulation	Protection against Physical Manipulation	<p>The TOE must provide protection against manipulation of the TOE (including its software and Data), the Security IC Embedded Software and the User Data. This includes protection against</p> <ul style="list-style-type: none"> - reverse-engineering (understanding the design and its properties and functions), - manipulation of the hardware and any data, as well as - controlled manipulation of memory contents (Application Data). 	O.OPERATE O.TAMPER O.CONF_DATA O.INTEG_DATA
O.Leak-	Protection	The Security IC must be protected against disclosure of confidential data processed in the Security IC (using	O.TAMPER



Forced	against Forced Information Leakage	methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker - by forcing a malfunction (refer to "Protection against Malfunction due to Environmental Stress (O.Malfunction)" and/or - by a physical manipulation (refer to "Protection against Physical Manipulation (O.Phys-Manipulation)". If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.	O.CONF_DATA
O.Abuse-Func	Protection against Abuse of Functionality	The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order to (i) disclose critical User Data, (ii) manipulate critical User Data of the Security IC Embedded Software, (iii) manipulate Soft-coded Security IC Embedded Software or (iv) bypass, deactivate, change or explore security features or security services of the TOE. Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Test Software which are not specified here.	O.OPERATE O.TAMPER O.CONF_DATA O.INTEG_DATA
O.Identification	TOE Identification	The TOE must provide means to store Initialisation Data and Pre-personalisation Data in its non-volatile memory. The Initialisation Data (or parts of them) are used for TOE identification.	No direct link to the composite-product. However chip traceability information stored in NVM is used by the TOE to answer identification CC assurance requirements.
O.RND	Random Numbers	The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have asufficient entropy. The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.	O.REPLAY
O.Add-Functions	Additional Specific Security Functionality	The TOE must provide the following specific security functionality to the Smartcard Embedded Software: Advanced Encryption Standard (AES), Triple Data Encryption Standard (3DES),, Rivest-Shamir-Adleman (RSA), Elliptic Curve Cryptography (EC)	Irrelevant for , 3DES, RSA, EC whose IC implementations are not used by the composite Relevant for AES linked to the O.AUTH security objective of the present ST.
O.MEM_ACCE	Area based	The TOE must provide the Smartcard Embedded Software with the capability to define restricted access memory	O.OPERATE



SS	Memory Access Control	areas. The TOE must then enforce the partitioning of such memory areas so that access of software to memory areas and privilege levels is controlled as required, for example, in a multi-application environment.	O.TAMPER O.CONF_DATA O.INTEG_DATA O.Separation
----	-----------------------	--	---

Table 16 – Composition – Security objectives for the TOE part

5.3.2 Statement of Compatibility – Security objectives for the environment part

IC ENV security objective label	IC ENV security objective title	IC ENV security objective content	Link to the composite-product
OE.Plat-Appl	Usage of Hardware Platform	To ensure that the TOE is used in a secure manner the Security IC Embedded Software shall be designed so that the requirements from the following documents are met: <ul style="list-style-type: none"> – (i) hardware data sheet for the TOE, – (ii) data sheet of the IC Dedicated Software of the TOE, – (iii) TOE application notes, other guidance documents, and – (iv) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report. 	Fulfilled by the composite-SAR ADV_COMP.1 (cf [CCDB], Appendix 1.2, §72 and §73)
OE.Resp-Appl	Treatment of User Data	Security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as required by the security needs of the specific application context. For example the Security IC Embedded Software will not disclose security relevant User Data to unauthorized users or processes when communicating with a terminal.	O.OPERATE O.TAMPER O.CONF_DATA O.INTEG_DATA
OE.Process-Sec-IC	Protection during composite product manufacturing	Security procedures shall be used after TOE Delivery up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use). This means that Phases after TOE Delivery up to the end of Phase 6 (refer to Section 1.2.3) must be protected appropriately.	OE_PROTECTION_AFTER_TOE_DELIVERY

Table 17 – Composition – Security objectives for the environment part



6 Security Requirements

This section specifies the requirements that apply to the TOE:

- Security Functional Requirements (SFRs)
- Security Assurance Requirements (SARs)

Note: The IC is evaluated and certified (see [9]) then the security requirements specifically applying to the IC [10] are not reproduced in this document.

Name	Description
U.CARD_MANAGER (Phases 4 to 7)	Card Manager He is able to access the services provided by the card to manage the smart card and its content. U.CARD_MANAGER is typically the Personalizer, the Issuer. Typical operations are application installation, application and card life-cycle evolutions. The Card Manager is authenticated by the OS through a dedicated secure mechanism.
S.EP (Phases 4 to 7)	EP Embedded application The EP subject corresponds to the embedded EP application code and any smart card related HW and SW required to process EP services. During the phase 7, S.EP is under the responsibility of U.ISSUER.
S.CARD_MANAGER (Phases 4 to 7)	Card management software The Card management software is responsible on behalf of the U.CARD_MANAGER for: <ul style="list-style-type: none"> • Providing a secure execution environment to the S.EP • Managing the card and EP (selection, lifecycle, card reset ...)

6.1 Security functional requirements (SFRs)

Functional component	Description	Dependencies	
FAU	FAU_ARP.1	Security alarms	FAU_SAA.1
	FAU_GEN.1	Audit data generation	FPT_STM.1
	FAU_SAA.1	Potential violation analysis	FAU_GEN.1
	FAU_SAR.1	Audit review	FAU_GEN.1
	FAU_STG.1	Protected audit trail storage	FAU_GEN.1
FCO	FCO_NRO.2	Enforced proof of origin	FIA_UID.1
FCS	FCS_CKM.1/ Session key	Cryptographic key generation	FCS_CKM.2 or FCS_COP.1, FCS_CKM.4
	FCS_CKM.2/ AES	Cryptographic key distribution	FCS_CKM.1 or FCS_COP.1, FCS_CKM.4
	FCS_CKM.3/ AES	Cryptographic key access	FCS_COP.1, FCS_CKM.4
	FCS_CKM.4	Cryptographic key destruction	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1
	FCS_COP.1	Cryptographic operation	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1, FCS_CKM.4



Functional component		Description	Dependencies
FDP	FDP_ACC.1/ Atomicity Life-cycle	Subset access control	FDP_ACF.1
	FDP_ACC.2/ Confidentiality Integrity Firewall	Complete access control	FDP_ACF.1
	FDP_ACF.1/ Life-cycle Confidentiality Integrity Firewall	Security attribute based access control	FDP_ACC.1, FMT_MSA.3
	FDP_IFC.1/ JCMV Application	Subset information flow control	FDP_IFF.1
	FDP_IFF.1/ JCMV Application	Simple security attributes	FDP_IFC.1, FMT_MSA.3
	FDP_ITC.1/ Application	Import of user data without security attributes	FDP_ACC.1 or FDP_IFC.1 or FTP_ITC.1 or FTP_TRP.1, FPT_TDC.1
	FDP_RIP.1/ JCS Key	Subset residual information protection	No dependencies
	FDP_ROL.1/ Firewall Atomicity	Atomic basic rollback	FDP_ACC.1 or FDP_IFC.1
	FDP_SDI.2	Stored data integrity monitoring and action	No dependencies
	FDP_UCT.1	Basic data exchange confidentiality	FTP_ITC.1 or FTP_TRP.1, FDP_ACC.1 or FDP_IFC.1
	FDP_UIT.1	Data exchange integrity	FDP_ACC.1 or FDP_IFC.1, FTP_ITC.1 or FTP_TRP.1
FIA	FIA_ATD.1/ AID	User attribute definition	No dependencies
	FIA_SOS.2	TSF generation of secrets	No dependencies
	FIA_UAU.1/ SAM	Timing of authentication	FIA_UID.1
	FIA_UAU.3	Unforgeable authentication	No dependencies
	FIA_UAU.4/ SAM	Single-use authentication mechanisms	No dependencies
	FIA_UAU.6/ SAM	Re-authenticating	No dependencies
	FIA_UID.1	Timing of identification	No dependencies
	FIA_UID.2/AID	User identification before any action	No dependencies
FIA_USB.1/AID	User-subject binding	FIA_ATD.1	
FMT	FMT_MSA.1/	Management of security attributes	FDP_ACC.1 or



Functional component		Description	Dependencies
	Confidentiality Integrity Life-cycle Application JCVM JCRE		FDP_IFC.1, FMT_SMR.1, FMT_SMF.1
	FMT_MSA.2/JCRE	Secure security attributes	FDP_ACC.1 or FDP_IFC.1, FMT_MSA.1, FMT_SMR.1
	FMT_MSA.3/ Confidentiality Integrity Life-cycle Application JCRE	Static attribute initialisation	FMT_MSA.1, FMT_SMR.1
	FMT_MTD.1	Management of TSF data	FMT_SMR.1, FMT_SMF.1
	FMT_SMF.1/JCS	Specification of management functions	No dependencies
	FMT_SMR.1	Security roles	FIA_UID.1
	FPR	FPR_UNO.1	Unobservability
FPT	FPT_FLS.1	Failure with preservation of secure state	No dependencies
	FPT_ITC.1	Inter-TSF confidentiality during transmission	No dependencies
	FPT_ITI.1	Inter-TSF detection of modification	No dependencies
	FPT_PHP.2	Notification of physical attack	FMT_MOF.1
	FPT_PHP.3	Resistance to physical attack	No dependencies
	FPT_RCV.4	Function recovery	No dependencies
	FPT_RPL.1	Replay detection	No dependencies
	FPT_TDC.1/JCS	Inter-TSF basic TSF data consistency	No dependencies
FPT_TST.1	TSF testing	No dependencies	

Table 18 – List of SFR



6.1.1 FAU – Security audit

FAU_ARP.1 / FAU_SAA.1	
Potential security violation	Actions
<ul style="list-style-type: none"> ▪ Integrity error regarding the patch executable code (if any) 	<ul style="list-style-type: none"> ▪ Terminate the card by irreversibly switching to the GP "TERMINATED" state. ▪ Then mute the card.
<ul style="list-style-type: none"> ▪ Integrity errors regarding applicative data mentioned (including keys), card state, data involved in the backup mechanism. ▪ Integrity errors regarding the patch table (if any) ▪ Unauthorized access to memories ▪ Abnormal code execution regarding: GP card state, IO emission, backup mechanism, Javacard flow ▪ Cryptography attack detection ▪ EEPROM programming failure 	<ul style="list-style-type: none"> ▪ Increment the fault detection counter (FDC). ▪ If the FDC max value is reached, terminate the card by irreversibly switching to the GP "TERMINATED" state. ▪ Then mute the card.
<ul style="list-style-type: none"> ▪ IC-triggered errors (captors e.g. external frequency/voltage out of range, glitch detection, chip component corruption e.g. internal clock/timer, die integrity protection alteration). 	<ul style="list-style-type: none"> ▪ Mute the card
<ul style="list-style-type: none"> ▪ Wrong commands ▪ Inconsistent user or TSF data ▪ Bad sequence of commands ▪ APDU inadequate replay 	<ul style="list-style-type: none"> ▪ Bring the card to a secure state by aborting command execution with emission of an adequate status word.
<ul style="list-style-type: none"> ▪ Card tearing 	<ul style="list-style-type: none"> ▪ Bring the card to a secure state.

Table 19 – FAU_ARP.1 / FAU_SAA.1

FAU_ARP.1: SECURITY ALARMS

Hierarchical to:	No other components.
FAU_ARP.1.1	The TSF shall take [assignment: list of actions] upon detection of a potential security violation.
Dependencies:	FAU_SAA.1

Hierarchical to:	No other components.
FAU_ARP.1.1	The TSF shall take actions given in Table 19 – FAU_ARP.1 / FAU_SAA.1 upon detection of a potential security violation.
Dependencies:	FAU_SAA.1



FAU_GEN.1: AUDIT DATA GENERATION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FAU_GEN.1.1</i>	<p>The TSF shall be able to generate an audit record of the following auditable events:</p> <ul style="list-style-type: none"> a) Start-up and shutdown of the audit functions; b) All auditable events for the [selection, choose one of: minimum, basic, detailed, not specified] level of audit; and c) [assignment: other specifically defined auditable events].
<i>FAU_GEN.1.2</i>	<p>The TSF shall record within each audit record at least the following information:</p> <ul style="list-style-type: none"> a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: other audit relevant information].
<i>Dependencies:</i>	<i>FPT_STM.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FAU_GEN.1.1	<p>The TSF shall be able to generate an audit record of the following auditable events:</p> <ul style="list-style-type: none"> a) Start-up and shutdown of the audit functions; b) All auditable events for the not specified level of audit; and c) The following auditable events: <ul style="list-style-type: none"> - last 6 Debit operations (APDU commands "DEBIT PURSE") - last Credit operations (APDU commands "CREDIT PURSE") <p><u>Application note:</u> <i>In the CREDIT PURSE and DEBIT PURSE command, the Transaction Data, Transaction Receipt and Transaction Signature is stored in the EEPROM memory and returned in the READ PURSE command.</i></p> <p><u>Refinement:</u> <i>The audit functions are active all the time, hence item a) Start-up and shutdown of the TOE is not relevant.</i></p>
FAU_GEN.1.2	<p>The TSF shall record within each audit record at least the following information:</p> <ul style="list-style-type: none"> a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, the following audit relevant information: <ul style="list-style-type: none"> - Transaction sequence number - Transaction date - Transaction amount - Electronic value - Device ID - Purse balance <p><u>Refinement:</u> <i>Date and time of events are determined by the terminal.</i></p>
<i>Dependencies:</i>	<i>FPT_STM.1 not satisfied (see §8.3.2 #1)</i>



 FAU_SAA.1: POTENTIAL VIOLATION ANALYSIS

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FAU_SAA.1.1</i>	<i>The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.</i>
<i>FAU_SAA.1.1</i>	<i>The TSF shall enforce the following rules for monitoring audited events:</i> <i>a) Accumulation or combination of [assignment: subset of defined auditable events] known to indicate a potential security violation;</i> <i>b) [assignment: any other rules].</i>
<i>Dependencies:</i>	<i>FAU_GEN.1</i>

Hierarchical to:	No other components.
FAU_SAA.1.1	The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the enforcement of the SFRs.
FAU_SAA.1.2	The TSF shall enforce the following rules for monitoring audited events: a) Accumulation or combination of defined auditable events given in Table 19 – FAU_ARP.1 / FAU_SAA.1 known to indicate a potential security violation; b) any other rules: none . <i>Refinement:</i> <i>The "audited events" are detected by the TSF or by the underlying IC. In particular, the TSF shall monitor all the events generated by the Security IC physical detectors that are made available to the TSF (e.g. by interrupt routines or status flags).</i>
Dependencies:	FAU_GEN_1

 FAU_SAR.1: AUDIT REVIEW

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FAU_SAR.1.1</i>	<i>The TSF shall provide [assignment: authorized users] with the capability to read [assignment: list of audit information] from the audit records.</i>
<i>FAU_SAR.1.2</i>	<i>The TSF shall provide the audit records in a manner suitable for the user to interpret the information.</i>
<i>Dependencies:</i>	<i>FAU_GEN.1 Audit data generation</i>

Hierarchical to:	No other components.
FAU_SAR.1.1	The TSF shall provide all users with the capability to read the following list of audit information from the audit records: - Transaction sequence number - Transaction date - Transaction amount - Electronic value - Device ID - Purse balance
FAU_SAR.1.2	The TSF shall provide the audit records in a manner suitable for the user to interpret the information.
Dependencies:	FAU_GEN_1



 FAU_STG.1: PROTECTED AUDIT TRAIL STORAGE

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FAU_STG.1.1</i>	<i>The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.</i>
<i>FAU_STG.1.2</i>	<i>The TSF shall be able to [selection, choose one of: prevent, detect] unauthorized modifications to the stored audit records in the audit trail.</i>
<i>Dependencies:</i>	<i>FAU_GEN.1 Audit data generation</i>

Hierarchical to:	No other components.
FAU_STG.1.1	The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.
FAU_STG.1.2	The TSF shall be able to detect unauthorized modifications to the stored audit records in the audit trail.
Dependencies:	FAU_GEN_1



6.1.2 FCO – Communication

FCO_NRO.2: ENFORCED PROOF OF ORIGIN

Hierarchical to:	FCO_NRO.1
FCO_NRO.2.1	The TSF shall enforce the generation of evidence of origin for transmitted [assignment: list of information types] at all times.
FCO_NRO.2.2	The TSF shall be able to relate the [assignment: list of attributes] of the originator of the information, and the [assignment: list of information fields] of the information to which the evidence applies.
FCO_NRO.2.3	The TSF shall provide a capability to verify the evidence of origin of information to [selection: originator, recipient, [assignment: list of third parties]] given [assignment: limitations on the evidence of origin] .
Dependencies:	FIA_UID.1

Hierarchical to:	FCO_NRO.1						
FCO_NRO.2.1 /CPU authentication	The TSF shall enforce the generation of evidence of origin for transmitted CPU authentication by the Terminal at all times.						
FCO_NRO.2.2 /CPU authentication	The TSF shall be able to relate the list of attributes of the originator of the information, and the list of information fields of the information to which the evidence applies. <table border="1" data-bbox="544 1039 1450 1171"> <thead> <tr> <th>List of attribute</th> <th>List of information field</th> <th>Command</th> </tr> </thead> <tbody> <tr> <td>TOKEN</td> <td>Card Random number Terminal Random</td> <td>INITIATE PROCESSING</td> </tr> </tbody> </table>	List of attribute	List of information field	Command	TOKEN	Card Random number Terminal Random	INITIATE PROCESSING
List of attribute	List of information field	Command					
TOKEN	Card Random number Terminal Random	INITIATE PROCESSING					
FCO_NRO.2.3 /CPU authentication	The TSF shall provide a capability to verify the evidence of origin of information to originator, recipient, U.ISSUER given the limitations on the evidence of origin: none.						
Dependencies:	FIA_UID.1 not satisfied (see §8.3.2 #2)						

Hierarchical to:	FCO_NRO.1									
FCO_NRO.2.1 /SAM authentication	The TSF shall enforce the generation of evidence of origin for transmitted SAM authentication by the CPU at all times.									
FCO_NRO.2.2 /SAM authentication	The TSF shall be able to relate the list of attributes of the originator of the information, and the list of information fields of the information to which the evidence applies. <table border="1" data-bbox="544 1628 1450 2018"> <thead> <tr> <th>List of attribute</th> <th>List of information field</th> <th>Command</th> </tr> </thead> <tbody> <tr> <td>TOKEN</td> <td>Random number Transaction amount Transaction time</td> <td>Payment commands: INITIATE PROCESSING, DEBIT PURSE (DEBIT, EXTENDED DEBIT, CANCEL DEBIT) CREDIT PURSE (CREDIT, LOAD, REFUND)</td> </tr> <tr> <td>MAC</td> <td></td> <td>Purse Management commands: PUT DATA (EXTEND CARD) WRITE LOCK (ACTIVATE, CANCEL ACTIVATE, BLOCK</td> </tr> </tbody> </table>	List of attribute	List of information field	Command	TOKEN	Random number Transaction amount Transaction time	Payment commands: INITIATE PROCESSING, DEBIT PURSE (DEBIT, EXTENDED DEBIT, CANCEL DEBIT) CREDIT PURSE (CREDIT, LOAD, REFUND)	MAC		Purse Management commands: PUT DATA (EXTEND CARD) WRITE LOCK (ACTIVATE, CANCEL ACTIVATE, BLOCK
List of attribute	List of information field	Command								
TOKEN	Random number Transaction amount Transaction time	Payment commands: INITIATE PROCESSING, DEBIT PURSE (DEBIT, EXTENDED DEBIT, CANCEL DEBIT) CREDIT PURSE (CREDIT, LOAD, REFUND)								
MAC		Purse Management commands: PUT DATA (EXTEND CARD) WRITE LOCK (ACTIVATE, CANCEL ACTIVATE, BLOCK								



			PAYMENT, UNBLOCK PAYMENT, ENABLE AUTO LOAD, DISABLE AUTO LOAD)
	MAC TOKEN		File Management commands: (MAC, TOKEN) READ RECORD, UPDATE RECORD, APPEND RECORD (TOKEN) CHANGE KEY – APP ADMIN (TOKEN) EXTERNAL AUTHENTICATE
<p><i>Application note:</i> <i>(Free) GET DATA</i> <i>(Free) READ RECORD, UPDATE RECORD, APPEND RECORD</i></p>			
FCO_NRO.2.3 /SAM authentication	The TSF shall provide a capability to verify the evidence of origin of information to originator, recipient, U.ISSUER given the limitations on the evidence of origin: none.		
Dependencies:	FIA_UID.1 not satisfied (see §8.3.2 #2)		



6.1.3 FCS – Cryptographic support

FCS_CKM.1: CRYPTOGRAPHIC KEY GENERATION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FCS_CKM.1.1</i>	<i>The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meets the following: [assignment: list of standards].</i>
<i>Dependencies:</i>	<i>FCS_CKM.2 or FCS_COP.1, FCS_CKM.4</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FCS_CKM.1.1/ Session	The TSF shall generate session cryptographic keys in accordance with a specified cryptographic AES algorithm and specified cryptographic sizes 128 bits that meets the following: [AES] . <i>Application note:</i> <i>Only the application session keys are concerned by this SFR.</i>
<i>Dependencies:</i>	FCS_COP.1 , FCS_CKM.4

FCS_CKM.2: CRYPTOGRAPHIC KEY DISTRIBUTION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FCS_CKM.2.1</i>	<i>The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: cryptographic key distribution method] that meets the following: [assignment: list of standards].</i>
<i>Dependencies:</i>	<i>FCS_CKM.1 or FCS_COP.1, FCS_CKM.4</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FCS_CKM.2.1/AES	The TSF shall distribute AES cryptographic keys in accordance with specified cryptographic " javacard.security " package AESKey.SetKEY that meet the following: JCAPI . <i>Application note:</i> <i>The following keys are concerned by this SFR.</i> <ul style="list-style-type: none"> ▪ APP ADMIN – File Access ▪ DEBIT/CREDIT – Payment ▪ SIGNATURE KEY – Payment signature generation ▪ CPU ADMIN – CPU Administrative purpose ▪ ISSUER – CPU Parameters update
<i>Dependencies:</i>	FCS_COP.1 , FCS_CKM.4



FCS_CKM.3: CRYPTOGRAPHIC KEY ACCESS

Hierarchical to:	No other components.
FCS_CKM.3.1	The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: cryptographic key distribution method] that meets the following: [assignment: list of standards].
Dependencies:	FCS_CKM.1 or FCS_COP.1, FCS_CKM.4

Hierarchical to:	No other components.
FCS_CKM.3.1 /AES	The TSF shall distribute cryptographic keys in accordance with specified cryptographic key distribution methods that meet the following: list of standards . Refinement given in Table 20 – FCS_CKM.3. <u>Application note:</u> The following keys are concerned by this SFR. <ul style="list-style-type: none"> ▪ APP ADMIN – File Access ▪ DEBIT/CREDIT – Payment ▪ Any application session keys
Dependencies:	FCS_COP.1 , FCS_CKM.4

FCS_CKM.3			
Cryptographic keys	Objective	Key access methods	List of standards
AES key (APP ADMIN KEY)	Update	javacard.security package AESKey.setKey	JCAPI
AES key (DEBIT/CREDIT KEYS and any application session keys)	Usage	Javacardx.crypto package Cipher.init Cipher.doFinal	JCAPI

Table 20 – FCS_CKM.3

FCS_CKM.4: CRYPTOGRAPHIC KEY DESTRUCTION

Hierarchical to:	No other components.
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: cryptographic key destruction method] that meets the following: [assignment: list of standards].
Dependencies:	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1

Hierarchical to:	No other components.
FCS_CKM.4.1	The TSF shall destroy all cryptographic keys in accordance with a specified cryptographic key destruction method clearKey() method that meets the following: none . <u>Application note:</u> Any application session keys are concerned by this SFR.
Dependencies:	FCS_CKM.1



 FCS_COP.1: CRYPTOGRAPHIC OPERATION

Hierarchical to:	No other components.
FCS_COP.1.1	The TSF shall perform [assignment: cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment; cryptographic key sizes] that meets the following: [assignment: list of standards].
Dependencies:	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1, FCS_CKM.4

Hierarchical to:	No other components.
FCS_COP.1.1	The TSF shall perform cryptographic operations in accordance with a specified cryptographic algorithms and cryptographic key sizes that meets the following: list of standards . Refinement given in Table 21 – FCS_COP.1.
Dependencies:	FCS_CKM.1 , FCS_CKM.4

FCS_COP.1				
Cryptographic operations	Objectives	Cryptographic algorithms	Key sizes	List of standards
Session key	To compute all cryptographic operations	AES	128 bits	AES Encrypt
MAC	To compute MAC for Purse Management and File Management commands (see FCO NRO 2)	AES	128 bits	AES Encrypt
TOKEN	To compute TOKEN for Payment and File Management commands (see FCO NRO 2)	AES	128 bits	AES Encrypt

Table 21 – FCS_COP.1



6.1.4 FDP – User data protection

FDP_ACC.1: SUBSET ACCESS CONTROL

<i>Hierarchical to:</i>	<i>No other components</i>
<i>FDP_ACC.1.1</i>	<i>The TSF shall enforce the [assignment; access control SFP] on [assignment: list of subjects and objects and operations among subjects and objects covered by the SFP].</i>
<i>Dependencies:</i>	<i>FDP_ACF.1</i>

<i>Hierarchical to:</i>	<i>No other components</i>
FDP_ACC.1.1/Atomicity	The TSF shall enforce the Atomicity access control SFP on list of subjects and objects and operations among subjects and objects covered by the SFP. See Table 22 – FDP_ACC.1/Atomicity. <i>Application note:</i> <i>This SFP determines which memory operations have to be performed atomically, meaning that memory rollback is performed if the corresponding sequence is interrupted. Such atomicity mechanisms permit to avoid reaching inconsistent states through incomplete memory updates.</i>
<i>Dependencies:</i>	<i>FDP_ACF.1 for Atomicity not satisfied (see §8.3.2 #3)</i>

SP item	Subjects	Objects	Operations
#1	S.EP	Any EF/Purse	Append or Update
#2	S.EP	APP ADMIN KEY	Update

Table 22 – FDP_ACC.1/Atomicity

<i>Hierarchical to:</i>	<i>No other components</i>
FDP_ACC.1.1/Life-cycle	The TSF shall enforce the Life-cycle access control SFP on list of subjects and objects and operations among subjects and objects covered by the SFP. See Table 23 – FDP_ACC.1/Life-cycle. <i>Application note:</i> <i>This SFP deals with life-cycle states and the impact on authorized operations. Two life-cycles have to be considered for the TOE:</i> <ul style="list-style-type: none"> ▪ <i>The platform life-cycle.</i> ▪ <i>The applet life-cycle.</i>
<i>Dependencies:</i>	FDP_ACF.1



SP item	Subjects	Objects	Operations
#1	S.EP	Assets and files related to application	Payment commands Applet and platform state transitions
#2	S.CARD_MANAGER	Assets related to platform	Availability of embedded applications Availability of administrative commands Applet and platform state transitions

Table 23 – FDP_ACC.1/Life-cycle

FDP_ACC.2: COMPLETE ACCESS CONTROL

<i>Hierarchical to:</i>	FDP_ACC.1
FDP_ACC.2.1	The TSF shall enforce the [assignment; access control SFP] on [assignment: list of subjects and objects] and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2	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.
<i>Dependencies:</i>	FDP_ACF.1

SP item	Subjects (APDU interpreter)	Access mode	Objects	Access control ⁶			
				State	A	P	N
#1	READ RECORD	Read	Any EF	Started	X	X	
	UPDATE RECORD	Update	Any EF	Started	X	X	
	APPEND RECORD	Append	Any EF	Started	X	X	
#2	READ RECORD	Read	Transaction log file	Started	Free		
	- DEBIT PURSE - CREDIT PURSE	Update	Transaction log file	Active		X	
#3	READ PURSE	Read	Purse balance	Ready	Free		
	- DEBIT PURSE - CREDIT PURSE	Update	Purse balance	Active		X	
#4	READ PURSE	Read	(last) CTC	Ready	Free		
	INITIATE PROCESSING	Update	CTC	Started		X	
#5	-	Read	Applicative session keys	-			X
#6	-	Read	Debit/Credit keys	-			X
#7	-	Read	APP ADMIN KEY	-			X
	CHANGE KEY – APP ADMIN	Update	APP ADMIN KEY	Active		X	
#8	READ PURSE	Read	Purse administration data (Lock)	Ready	Free		
	WRITE LOCK	Update	Purse administration data (PUC)	Active		X	
#9	READ PURSE	Read	Purse attribute (limits)	Ready	Free		
	PUT DATA	Update	Purse attribute (limits)	Active		X	
#10	GET DATA	Read	DGI data	Ready	Free		
	-	Update	DGI data	-			X

⁶ Access control is split in State for State machine, A for Always, P for Protected, and N for Never.



Table 24 – FDP_ACC.2/Confidentiality and Integrity

Hierarchical to:	FDP_ACC.1
FDP_ACC.2.1/Confidentiality	The TSF shall enforce the Confidentiality access control SFP on list of subjects and objects and all operations among subjects and objects covered by the SFP. See Table 24 – FDP_ACC.2/Confidentiality and Integrity
FDP_ACC.2.2/Confidentiality	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.
Dependencies:	FDP_ACF.1

Hierarchical to:	FDP_ACC.1
FDP_ACC.2.1/Integrity	The TSF shall enforce the Integrity access control SFP on list of subjects and objects and all operations among subjects and objects covered by the SFP. See Table 24 – FDP_ACC.2/Confidentiality and Integrity
FDP_ACC.2.2/Integrity	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.
Dependencies:	FDP_ACF.1

Hierarchical to:	FDP_ACC.1
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.
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.
Dependencies:	FDP_ACF.1

FDP_ACF.1: SECURITY ATTRIBUTE BASED ACCESS CONTROL

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FDP_ACF.1.1</i>	<i>The TSF shall enforce the [assignment: access control SFP] to objects based on the following: [assignment: list of subjects and objects controlled under the indicated SFP, and for each the SFP-relevant security attributes, or named groups of SFP-relevant security attributes].</i>
<i>FDP_ACF.1.2</i>	<i>The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].</i>
<i>FDP_ACF.1.3</i>	<i>The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects].</i>
<i>FDP_ACF.1.4</i>	<i>The TSF shall explicitly deny access of subjects to objects based on the [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].</i>
<i>Dependencies:</i>	<i>FDP_ACC.1, FMT_MSA.3</i>



Hierarchical to:	No other components.
FDP_ACF.1.1/ Life-cycle	The TSF shall enforce the Life-cycle access control SFP to objects based on the following: list of subjects and objects controlled under the indicated SFP, and for each the SFP-relevant security attributes, or named groups of SFP-relevant security attributes. See Table 25 – Life-cycle SFP rules
FDP_ACF.1.2/ Life-cycle	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects.
FDP_ACF.1.3/ Life-cycle	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: rules, based on security attributes, that explicitly authorize access of subjects to objects.
FDP_ACF.1.4/ Life-cycle	The TSF shall explicitly deny access of subjects to objects based on the rules, based on security attributes, that explicitly deny access of subjects to objects.
Dependencies:	FDP_ACF.1 , FMT_MSA.3



SP item	Security attributes	Rules governing access	Rules explicitly authorize access	Rules explicitly deny access
#1	Platform state Applet state	<p>During usage phase, the purse services shall be available only if:</p> <ul style="list-style-type: none"> - platform state is SECURED - applet state is SELECTABLE <p>In any other case, the applet (S.EP) shall not process any operation.</p>	None	No subject (on behalf on any user) shall be able to put back the applet state in INSTALLED.
#2	Platform state	<p>During usage phase, the platform administrative services shall be available only if:</p> <ul style="list-style-type: none"> - platform state is SECURED <p>When the platform state is CARD_LOCKED, the S.CARD_MANAGER shall only Select ISD or Respond to GET DATA.</p> <p>When the platform state is TERMINATED, the S.CARD_MANAGER shall only Respond to GET DATA. All other commands shall be rejected.</p>	The platform or any privileged applet shall always be able to put the card in the CARD_LOCKED or TERMINATED (in case of security violation detection).	No subject (on behalf on any user) shall be able to put back the platform state in OS_NATIF, OP_READY or INITIALIZED.

Table 25 – Life-cycle SFP rules



Hierarchical to:	No other components.										
FDP_ACF.1.1/ Firewall	<p>The TSF shall enforce the Firewall access control SFP to objects based on the following:</p> <table border="1" data-bbox="536 414 1441 582"> <thead> <tr> <th>Subject/Object</th> <th>Attributes</th> </tr> </thead> <tbody> <tr> <td>S.PACKAGE</td> <td>Context</td> </tr> <tr> <td>S.JCRE</td> <td>Selected Applet Context</td> </tr> <tr> <td>S.JCVM</td> <td>Currently Active Context</td> </tr> <tr> <td>O.JAVAOBJECT</td> <td>Sharing, Context, Life Time</td> </tr> </tbody> </table> <p>.</p>	Subject/Object	Attributes	S.PACKAGE	Context	S.JCRE	Selected Applet Context	S.JCVM	Currently Active Context	O.JAVAOBJECT	Sharing, Context, Life Time
Subject/Object	Attributes										
S.PACKAGE	Context										
S.JCRE	Selected Applet Context										
S.JCVM	Currently Active Context										
O.JAVAOBJECT	Sharing, Context, Life Time										
FDP_ACF.1.2/ Firewall	<p>The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:</p> <ul style="list-style-type: none"> ▪ R.JAVA.1 ([JCRE]) An S.PACKAGE may freely perform any of OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE, OP.THROW or OP.TYPE_ACCESS upon any O.JAVAOBJECT whose Sharing attribute has value "JCRE entry point" or "global array". ▪ R.JAVA.2 ([JCRE]) An S.PACKAGE may freely perform any of OP.ARRAY_ACCESS, OP.INSTANCE_FIELD, OP.INVK_VIRTUAL, OP.INVK_INTERFACE or OP.THROW upon any O.JAVAOBJECT whose Sharing attribute has value "Standard" and whose Lifetime attribute has value "PERSISTENT" only if O.JAVAOBJECT's Context attribute has the same value as the active context. ▪ R.JAVA.3 ([JCRE]) An S.PACKAGE may perform OP.TYPE_ACCESS upon an O.JAVAOBJECT whose Sharing attribute has value "SIO" only if O.JAVAOBJECT is being cast into (checkcast) or is being verified as being an instance of (instanceof) an interface that extends the Shareable interface. ▪ R.JAVA.4 ([JCRE]) An S.PACKAGE may perform OP.INVK_INTERFACE upon an O.JAVAOBJECT whose Sharing attribute has the value "SIO", and whose Context attribute has the value "Package AID", only if the invoked interface method extends the Shareable interface and one of the following conditions applies: a) The value of the attribute Selection Status of the package whose AID is "Package AID" is "Multiselectable", b) The value of the attribute Selection Status of the package whose AID is "Package AID" is "Non-multiselectable", and either "Package AID" is the value of the currently selected applet or otherwise "Package AID" does not occur in the attribute ActiveApplets. ▪ R.JAVA.5 An S.PACKAGE may perform an OP.CREATE only if the value of the Sharing parameter is "Standard". 										
FDP_ACF.1.3/ Firewall	<p>The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:</p> <ul style="list-style-type: none"> ▪ 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). 										



FDP_ACF.1.4/ Firewall	<p>The TSF shall explicitly deny access of subjects to objects based on the</p> <ul style="list-style-type: none"> ▪ 1) Any subject with OP.JAVA upon an O.JAVAOBJECT whose LifeTime attribute has value "CLEAR_ON_DESELECT" if O.JAVAOBJECT's Context attribute is not the same as the Selected Applet Context. Java Card System? Closed Configuration Protection Profile October 2009 Version 2.2 Sun Confidential / Proprietary Page 57/91 ▪ 2) Any subject attempting to create an object by the means of OP.CREATE and a "CLEAR_ON_DESELECT" Life Time parameter if the active context is not the same as the Selected Applet Context.
Dependencies:	FDP_ACF.1 , FMT_MSA.3



SP item	Security attributes	Rules governing access	Rules explicitly authorize access	Rules explicitly deny access
#1	<ul style="list-style-type: none"> ▪ File access condition ▪ State machine ▪ Card status ▪ Session key 	-	<ul style="list-style-type: none"> ▪ State machine = Started ▪ Card is "Activated" <u>Read:</u> <ul style="list-style-type: none"> ▪ Read access allowed <u>Update:</u> <ul style="list-style-type: none"> ▪ Card is not "Refunded" ▪ Update access allowed <u>Append:</u> <ul style="list-style-type: none"> ▪ Card is not "Refunded" ▪ Append access allowed <p>For Protected access:</p> <ul style="list-style-type: none"> ▪ Successful INITIATE PROCESSING ▪ Required session key available 	<ul style="list-style-type: none"> ▪ Incorrect state machine ▪ Card is not "Activated" <u>Read:</u> <ul style="list-style-type: none"> ▪ Read access not allowed <u>Update:</u> <ul style="list-style-type: none"> ▪ Card is "Refunded" ▪ Update access not allowed <u>Append:</u> <ul style="list-style-type: none"> ▪ Card is "Refunded" ▪ Append access not allowed <p>For Protected access:</p> <ul style="list-style-type: none"> ▪ INITIATE PROCESSING not successful ▪ Required session key not available
#2	<ul style="list-style-type: none"> ▪ File access condition ▪ State machine ▪ Card status ▪ TSQN ▪ Purse attributes 	-	<ul style="list-style-type: none"> ▪ State machine = Started ▪ Card is "Activated" <u>Read:</u> <ul style="list-style-type: none"> ▪ Read access allowed <u>Update:</u> <ul style="list-style-type: none"> ▪ Card is not "Refunded" ▪ TSQN is valid ▪ Successful INITIATE PROCESSING ▪ Card is not "Blocked" ▪ Purse attributes are valid 	<ul style="list-style-type: none"> ▪ Incorrect state machine ▪ Card is not "Activated" <u>Read:</u> <ul style="list-style-type: none"> ▪ Read access not allowed <u>Update:</u> <ul style="list-style-type: none"> ▪ Card is "Refunded" ▪ TSQN is not valid ▪ INITIATE PROCESSING not successful ▪ Card is "Blocked" ▪ Purse attributes are not valid
#3	<ul style="list-style-type: none"> ▪ State machine ▪ Card status ▪ TSQN ▪ Debit/Credit key ▪ Purse attributes 	-	<ul style="list-style-type: none"> ▪ Card is "Activated" <u>Read:</u> <ul style="list-style-type: none"> ▪ State machine = Ready ▪ Free <u>Update:</u> <ul style="list-style-type: none"> ▪ State machine = Active ▪ Card is not "Refunded" 	<ul style="list-style-type: none"> ▪ Card is not "Activated" <u>Read:</u> <ul style="list-style-type: none"> ▪ Incorrect state machine <u>Update:</u> <ul style="list-style-type: none"> ▪ Incorrect state machine ▪ Card is "Refunded"



	<ul style="list-style-type: none"> ▪ File access condition 		<ul style="list-style-type: none"> ▪ TSQN is valid ▪ Successful INITIATE PROCESSING ▪ Debit/Credit keys are valid ▪ Purse attributes are valid ▪ Update access allowed 	<ul style="list-style-type: none"> ▪ TSQN is not valid ▪ INITIATE PROCESSING not successful ▪ Debit/Credit keys are not valid ▪ Purse attributes are not valid ▪ Update access not allowed
#4	<ul style="list-style-type: none"> ▪ State machine ▪ Card status ▪ CTC ▪ Derived keys (CPU admin, Debit, ADMIN) 	-	<ul style="list-style-type: none"> ▪ Card is "Activated" <p><u>Read:</u></p> <ul style="list-style-type: none"> ▪ State machine = Ready ▪ Free <p><u>Update:</u></p> <ul style="list-style-type: none"> ▪ State machine = Started ▪ Successful READ PURSE ▪ CTC is valid ▪ Derived keys available 	<ul style="list-style-type: none"> ▪ Card is not "Activated" <p><u>Read:</u></p> <ul style="list-style-type: none"> ▪ Incorrect state machine <p><u>Update:</u></p> <ul style="list-style-type: none"> ▪ Incorrect state machine ▪ READ PURSE not successful ▪ CTC is not valid ▪ Derived keys not available
#5	Key ID	-	-	<p><u>Read:</u></p> <p>Key value reading is never authorized</p>
#6	Key ID	-	-	<p><u>Read:</u></p> <p>Key value reading is never authorized</p>
#7	<ul style="list-style-type: none"> ▪ Key ID ▪ State machine ▪ Card status 	-	<p><u>Update:</u></p> <ul style="list-style-type: none"> ▪ Card is "Activated" ▪ Card is not "Refunded" ▪ State machine = Active ▪ Successful EXTERNAL AUTHENTICATE 	<p><u>Read:</u></p> <p>Key value reading is never authorized</p> <p><u>Update:</u></p> <ul style="list-style-type: none"> ▪ Card is not "Activated" ▪ Card is "Refunded" ▪ Incorrect state machine ▪ EXTERNAL UTHENTICATE not successful
#8	<ul style="list-style-type: none"> ▪ State machine ▪ Card status ▪ TSQN ▪ Lock 	-	<p><u>Read:</u></p> <ul style="list-style-type: none"> ▪ Card is "Activated" ▪ State machine = Ready ▪ Free <p><u>Update:</u></p> <ul style="list-style-type: none"> ▪ Card is "Activated" if Block Payment, Enable/Disable Auto-Load, Cancel Activate Card is to be performed ▪ Card is not "Activated" if Activate Card is to be performed 	<p><u>Read:</u></p> <ul style="list-style-type: none"> ▪ Card is not "Activated" ▪ Incorrect state machine <p><u>Update:</u></p> <ul style="list-style-type: none"> ▪ Card is "Activated" if Block Payment, Enable/Disable Auto-Load, Cancel Activate Card is to be performed ▪ Card is not "Activated" if Activate Card is to be performed ▪ Card is "Blocked"



			<ul style="list-style-type: none"> ▪ Card is not "Blocked" ▪ Card is not "Refunded" ▪ TSQN is valid ▪ Lock is valid ▪ State machine = Active ▪ Successful INITIATE PROCESSING 	<ul style="list-style-type: none"> ▪ Card is "Refunded" ▪ TSQN is not valid ▪ Lock is not valid ▪ Incorrect state machine ▪ INITIATE PROCESSING not successful
#9	<ul style="list-style-type: none"> ▪ State machine ▪ Card status ▪ Derived Issuer key 	-	<p>Read:</p> <ul style="list-style-type: none"> ▪ Card is "Activated" ▪ State machine = Ready ▪ Free <p>Update:</p> <ul style="list-style-type: none"> ▪ Card is "Activated" ▪ Card is not "Refunded" ▪ State machine = Active ▪ Successful INITIATE PROCESSING ▪ Derived Issuer key available 	<p>Read:</p> <ul style="list-style-type: none"> ▪ Card is not "Activated" ▪ Incorrect state machine <p>Update:</p> <ul style="list-style-type: none"> ▪ Card is not "Activated" ▪ Card is "Refunded" ▪ Incorrect State machine ▪ INITIATE PROCESSING not successful ▪ Derived Issuer key not available
#10	<ul style="list-style-type: none"> ▪ State machine ▪ Card status 	-	<p>Read:</p> <ul style="list-style-type: none"> ▪ Card is "Activated" ▪ State machine = Ready ▪ Free 	<p>Read:</p> <ul style="list-style-type: none"> ▪ Card is not "Activated" ▪ Incorrect state machine <p>Update:</p> <p>DGI value update is never authorized</p>

Table 26 – Confidentiality and Integrity SFP rules



Hierarchical to:	No other components.
FDP_ACF.1.1/ Confidentiality	The TSF shall enforce the Confidentiality access control SFP to objects based on the following: list of subjects and objects controlled under the indicated SFP, and for each the SFP-relevant security attributes, or named groups of SFP-relevant security attributes. See Table 26 – Confidentiality and Integrity SFP rules.
FDP_ACF.1.2/ Confidentiality	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects.
FDP_ACF.1.3/ Confidentiality	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: rules, based on security attributes, that explicitly authorize access of subjects to objects.
FDP_ACF.1.4/ Confidentiality	The TSF shall explicitly deny access of subjects to objects based on the rules, based on security attributes, that explicitly deny access of subjects to objects.
Dependencies:	FDP_ACF.1 , FMT_MSA.3

Hierarchical to:	No other components.
FDP_ACF.1.1/ Integrity	The TSF shall enforce the Integrity access control SFP to objects based on the following: list of subjects and objects controlled under the indicated SFP, and for each the SFP-relevant security attributes, or named groups of SFP-relevant security attributes. See Table 26 – Confidentiality and Integrity SFP rules.
FDP_ACF.1.2/ Integrity	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects.
FDP_ACF.1.3/ Integrity	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: rules, based on security attributes, that explicitly authorize access of subjects to objects.
FDP_ACF.1.4/ Integrity	The TSF shall explicitly deny access of subjects to objects based on the rules, based on security attributes, that explicitly deny access of subjects to objects.
Dependencies:	FDP_ACF.1 , FMT_MSA.3

FDP_IFC.1: SUBSET INFORMATION FLOW CONTROL

Hierarchical to:	<i>No other components</i>
FDP_IFC.1.1	<i>The TSF shall enforce the [assignment: information flow control SFP] on [assignment: list of subjects, information, and operations that cause controlled information to flow and from controlled subjects covered by the SFP].</i>
Dependencies:	<i>FDP_IFF.1</i>



Hierarchical to:	No other components
FDP_IFC.1.1/ JCVM	The TSF shall enforce the JCVM information flow control SFP on S.LOCAL, S.MEMBER, I.DATA and OP.PUT(S1, S2, I) .
Dependencies:	FDP_IFF.1

Hierarchical to:	No other components
FDP_IFC.1.1/ Application	The TSF shall enforce the Transaction information flow control SFP on list of subjects, information, and operations that cause controlled information to flow and from controlled subjects covered by the SFP . See Table 27 – FDP_IFC.1/Application.
Dependencies:	FDP_IFF.1

SP item	Subjects	Informations	Operations
#1	S.EP	Data used for authentication purpose, exchanged between the TOE and the SAM.	<p>From SAM to CPU:</p> <ul style="list-style-type: none"> (TOKEN) INITIATE PROCESSING command. <p>From CPU to SAM:</p> <ul style="list-style-type: none"> (TOKEN) Payment commands. (MAC) Purse Management commands. (TOKEN or MAC) File Management commands.
#2	S.EP	The EM amount exchanged between the TOE and the SAM.	All kinds of Payment commands.
#3	S.EP	The sequence of commands accepted by the TOE during a transaction.	All kinds of Payment commands.

Table 27 – FDP_IFC.1/Application

FDP_IFF.1: SUBSET INFORMATION FLOW CONTROL

Hierarchical to:	No other components
FDP_IFF.1.1	The TSF shall enforce the [assignment: information flow control SFP] based on the following types of subject and information security attributes: [assignment: list of subjects and information controlled under the indicated SFP, and for each the security attributes] .
FDP_IFF.1.2	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes] .
FDP_IFF.1.3	The TSF shall enforce the [assignment: additional information flow control SFP] .
FDP_IFF.1.4	The TSF shall explicitly authorize an information flow based on the following



	<i>rules: [assignment: rules, based on security attributes, that explicitly authorize information flows].</i>
FDP_IFF.1.5	<i>The TSF shall explicitly deny an information flow based on the following rules: [assignment: rules, based on security attributes, that explicitly deny information flows].</i>
Dependencies:	<i>FDP_IFC.1, FMT_MSA.3</i>

Hierarchical to:	No other components
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: S.JCVM and the attribute "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: <ul style="list-style-type: none"> ▪ 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 .
FDP_IFF.1.4/ JCVM	The TSF shall explicitly authorize an information flow based on the following rules: <ul style="list-style-type: none"> ▪ All JCRE Permanent Entry Point Object may be stored in a S.MEMBER.
FDP_IFF.1.5/ JCVM	The TSF shall explicitly deny an information flow based on the following rules: <ul style="list-style-type: none"> ▪ The storage of the reference of an object with attribute JCRE Temporary Entry Point Object or Global Array in a static field, instance field or array element is forbidden.
Dependencies:	FDP_IFC.1 , FMT_MSA.3

Hierarchical to:	No other components
FDP_IFF.1.1/ Application	The TSF shall enforce the Transaction information flow control SFP based on the following types of subject and information security attributes: List of subjects and information controlled under the indicated SFP, and for each the security attributes. See Table 28 – Transaction SFP rules.
FDP_IFF.1.2/ Application	The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: for each operation, the security attribute-based relationship that must hold between subject and information security attributes.
FDP_IFF.1.3/ Application	The TSF shall enforce the additional information flow control SFP .
FDP_IFF.1.4/ Application	The TSF shall explicitly authorize an information flow based on the following rules: rules, based on security attributes, that explicitly authorize information flows.
FDP_IFF.1.5/ Application	The TSF shall explicitly deny an information flow based on the following rules: rules, based on security attributes, that explicitly deny information flows.
Dependencies:	FDP_IFC.1 , FMT_MSA.3



SP item	Security attribute	for each operation, the security attribute-based relationship that must hold between subject and information security attributes	additional information flow control SFP	rules, based on security attributes, that explicitly authorize information flows	rules, based on security attributes, that explicitly deny information flows
#1	From SAM to CPU: <ul style="list-style-type: none"> ▪ TOKEN From CPU to SAM: <ul style="list-style-type: none"> ▪ TOKEN ▪ MAC ▪ TOKEN or MAC 	TOKEN valid MAC valid	None	None	When TOKEN/MAC is wrong the information is not take into account by the subject.
#2	<ul style="list-style-type: none"> ▪ maximum balance ▪ maximum amount for Debit transaction 	Purse attributes valid	None	None	When Purse attributes are wrong the information is not take into account by the subject.
#3	<ul style="list-style-type: none"> ▪ State machine 	State machine transition valid Ready -> Started -> Active -> Debit/Credit/Auto-Load	None	None	When State machine transition is wrong the information is not take into account by the subject.

Table 28 – Transaction SFP rules



 FDP_ITC.1: IMPORT OF USER DATA WITHOUT SECURITY ATTRIBUTES

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FDP_ITC.1.1</i>	<i>The TSF shall enforce the [assignment: access control SFP(s) and/or information flow control SFP(s)] when importing user data, controlled under the SFP, from outside of the TOE.</i>
<i>FDP_ITC.1.2</i>	<i>The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.</i>
<i>FDP_ITC.1.3</i>	<i>The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [assignment: additional importation control rules].</i>
<i>Dependencies:</i>	<i>FDP_ACC.1 or FDP_IFC.1, FMT_MSA.3</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FDP_ITC.1.1/ Application	The TSF shall enforce the Transaction access control SFP when importing user data, controlled under the SFP, from outside of the TOE. <i>Refinement:</i> <i>"User data" stands for user/TSF data entering the EP during Debit, Credit or Auto-Load transactions.</i>
FDP_ITC.1.2/ Application	The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.
FDP_ITC.1.3/ Application	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: no additional importation control rules.
<i>Dependencies:</i>	FDP_IFC.1 , FMT_MSA.3

 FDP_RIP.1: SUBSET RESIDUAL INFORMATION PROTECTION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FDP_RIP.1.1</i>	<i>The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: allocation of the resource to, deallocation of the resource from] the following objects: [assignment: list of objects].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FDP_RIP.1.1/JCS	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 instance and arrays. <i>Application note:</i> <i>The semantics of the java programming language requires for any object field and array position to be initialized with default values when resource is allocated [JVM, §2.5.1].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FDP_RIP.1.1/Key	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: key values.



	<i>Refinement:</i> <i>"Information content" stands for user/TSF data with confidentiality constraints.</i>
Dependencies:	No dependencies.

FDP_ROL.1: ATOMIC BASIC ROLLBACK

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FDP_ROL.1.1</i>	<i>The TSF shall enforce [assignment: access control SFP(s) and/or information flow control SFP(s)] to permit the rollback of the [assignment: list of operations] on the [assignment: information and/or list of objects].</i>
<i>FDP_ROL.1.2</i>	<i>The TSF shall permit operations to be rolled back within the [assignment: boundary limit to which rollback may be performed].</i>
<i>Dependencies:</i>	<i>FDP_ACC.1 or FDP_IFC.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FDP_ROL.1.1/Firewall	The TSF shall enforce Firewall access control SFP(s) and JVM information flow control SFP(s) to permit the rollback of the operations OP.JAVA and OP.CREATE on the O.JAVAOBJECTS .
FDP_ROL.1.2/Firewall	The TSF shall permit operations to be rolled back within the scope of the select(), deselect(), process() or install() call, notwithstanding the restrictions given in [JCRE], within the bounds of the Commit Capacity [JCRE], and those described in [JCAPI].
Dependencies:	FDP_ACC.1

<i>Hierarchical to:</i>	<i>No other components.</i>
FDP_ROL.1.1/Atomicity	The TSF shall enforce Atomicity access control SFP(s) to permit the rollback of the Append/Update and of the Update operations on the any EF/Purse and APP ADMIN KEY objects .
FDP_ROL.1.2/Atomicity	The TSF shall permit operations to be rolled back within the following limits: none .
Dependencies:	FDP_ACC.1

FDP_SDI.2: STORED DATA INTEGRITY MONITORING AND ACTION

<i>Hierarchical to:</i>	<i>FDP_SDI.1</i>
<i>FDP_SDI.2.1</i>	<i>The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: integrity errors] on all objects, based on the following attributes: [assignment: user data attributes].</i>
<i>FDP_SDI.2.2</i>	<i>Upon detection of a data integrity error, the TSF shall [assignment: action to be taken].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>FDP_SDI.1</i>
<i>FDP_SDI.2.1</i>	The TSF shall monitor user data stored in containers controlled by the TSF for integrity errors on all objects, based on the following attributes: checksum .
<i>FDP_SDI.2.2</i>	Upon detection of a data integrity error, the TSF shall take the following actions: <ul style="list-style-type: none"> ▪ Increment the FDC ▪ If FDC max value is reached, terminate the card by



	<p>irreversibly switching to the GP "TERMINATED" state</p> <ul style="list-style-type: none"> ▪ Then mute the card <p><i>Refinement:</i> "User data" stands for user/TSF data with integrity constraints.</p>
Dependencies:	No dependencies.

FDP_UCT.1: BASIC DATA EXCHANGE CONFIDENTIALITY

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FDP_UCT.1.1</i>	<i>The TSF shall enforce the [assignment: access control SFP(s) and/or information flow control SFP(s)] to be able to [selection: transmit, receive] user data in a manner protected from unauthorized disclosure.</i>
<i>Dependencies:</i>	<i>FTP_ITC.1 or FTP_TRP.1, FDP_ACC.1 or FDP_IFC.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FDP_UCT.1.1	The TSF shall enforce the Confidentiality access control SFP(s) to be able to receive user data in a manner protected from unauthorized disclosure.
	<i>Refinement:</i> "User data" stands for user data with confidentiality constraints.
Dependencies:	FTP_ITC.1 or FTP_TRP.1 not supported (see §8.3.2 #4), FDP_ACC.1

FDP_UIT.1: DATA EXCHANGE INTEGRITY

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FDP_UIT.1.1</i>	<i>The TSF shall enforce the [assignment: access control SFP(s) and/or information flow control SFP(s)] to be able to [selection: transmit, receive] user data in a manner protected from [selection: modification, deletion, insertion, replay] errors.</i>
<i>FDP_UIT.1.2</i>	<i>The TSF shall be able to determine on receipt of user data, whether [selection: modification, deletion, insertion, replay] has occurred.</i>
<i>Dependencies:</i>	<i>FDP_ACC.1 or FDP_IFC.1, FTP_ITC.1 or FTP_TRP.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FDP_UIT.1.1	The TSF shall enforce the Integrity access control SFP(s) to be able to transmit/receive user data in a manner protected from modification/deletion/insertion/replay errors.
FDP_UIT.1.2	The TSF shall be able to determine on receipt of user data, whether modification/deletion/insertion/replay has occurred.
	<i>Refinement:</i> "User data" stands for user data with integrity constraints.
Dependencies:	FDP_ACC.1 , FTP_ITC.1 or FTP_TRP.1 not supported (see §8.3.2 #5)



6.1.5 FIA – Identification and authentication

FIA_ATD.1: USER ATTRIBUTE DEFINITION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FIA_ATD.1.1</i>	<i>The TSF shall provide a mechanism to generate secrets that meet [assignment: a defined quality metric].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FIA_ATD.1/AID	The TSF shall maintain the following list of security attributes belonging to individual users: <ul style="list-style-type: none"> ▪ Package AID ▪ Applet's version number ▪ Registered applet AID ▪ Applet selection status <p><i>Refinement:</i> <i>"Individual users" stands for applets.</i></p>
<i>Dependencies:</i>	<i>No dependencies.</i>

FIA_SOS.2: TSF GENERATION OF SECRETS

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FIA_SOS.2.1</i>	<i>The TSF shall provide a mechanism to generate secrets that meet [assignment: a defined quality metric].</i>
<i>FIA_SOS.2.2</i>	<i>The TSF shall be able to enforce the use of TSF generated secrets for [assignment: list of TSF functions].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FIA_SOS.2.1	The TSF shall provide a mechanism to generate secrets that meet the standard level of French scheme ANSSI requirements for RNG. <p><i>Refinement:</i> <i>"Secrets" stands for random values.</i></p>
FIA_SOS.2.2	The TSF shall be able to enforce the use of TSF generated secrets for the generation of 8-bytes challenge (APDU command INITIATE PROCESSING).
<i>Dependencies:</i>	<i>No dependencies.</i>

FIA_UAU.1: TIMING OF AUTHENTICATION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FIA_UAU.1.1</i>	<i>The TSF shall allow [assignment: list of TSF mediated actions] on behalf of the user to be performed before the user is authenticated.</i>
<i>FIA_UAU.1.2</i>	<i>The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.</i>
<i>Dependencies:</i>	<i>FIA_UID.1 Timing of identification</i>



Hierarchical to:	No other components.
FIA_UAU.1.1/ CPU authentication	The TSF shall allow the CPU authentication by the Terminal on behalf of the user to be performed before the user is authenticated. <ul style="list-style-type: none"> ▪ INITIATE PROCESSING command <p><i>Refinement:</i> <i>"User" stands for Terminal.</i></p>
FIA_UAU.1.2/ CPU authentication	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	FIA_UID.1 not supported (see §8.3.2 #6)

Hierarchical to:	No other components.
FIA_UAU.1.1/ SAM authentication	The TSF shall allow the SAM authentication by the CPU on behalf of the user to be performed before the user is authenticated. <ul style="list-style-type: none"> ▪ Payment commands ▪ Purse Management commands ▪ File Management commands <p><i>Refinement:</i> <i>"User" stands for CPU.</i></p>
FIA_UAU.1.2/ SAM authentication	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	FIA_UID.1 not supported (see §8.3.2 #6)

FIA_UAU.3: UNFORGEABLE AUTHENTICATION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FIA_UAU.3.1</i>	<i>The TSF shall [selection: detect, prevent] use of authentication data that has been forged by any user of the TSF.</i>
<i>FIA_UAU.3.2</i>	<i>The TSF shall [selection: detect, prevent] use of authentication data that has been copied from any other user of the TSF.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	No other components.
FIA_UAU.3.1	The TSF shall prevent use of authentication data that has been forged by any user of the TSF.
FIA_UAU.3.2	The TSF shall prevent use of authentication data that has been copied from any other user of the TSF.
Dependencies:	No dependencies.

FIA_UAU.4: SINGLE-USE AUTHENTICATION MECHANISMS

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FIA_UAU.4.1</i>	<i>The TSF shall prevent reuse of authentication data related to [assignment: identified authentication mechanism(s)].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	No other components.
FIA_UAU.4.1/ CPU authentication	The TSF shall prevent reuse of authentication data related to the CPU authentication by the Terminal mechanism <ul style="list-style-type: none"> ▪ INITIATE PROCESSING command
Dependencies:	No dependencies.



Hierarchical to:	No other components.
FIA_UAU.4.1/ SAM authentication	The TSF shall prevent reuse of authentication data related to the Terminal authentication mechanisms: <ul style="list-style-type: none"> ▪ Payment commands ▪ Purse Management commands ▪ File Management commands
Dependencies:	No dependencies.

FIA_UAU.6: RE-AUTHENTICATING

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FIA_UAU.6.1</i>	<i>The TSF shall re-authenticate the user under the conditions [assignment: list of conditions under which re-authentication is required].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	No other components.
FIA_UAU.6.1/ CPU authentication	The TSF shall re-authenticate the user under the conditions: <ul style="list-style-type: none"> ▪ Beginning of Payment commands <p><i>Refinement:</i> <i>"User" stands for Terminal.</i></p>
Dependencies:	No dependencies.

Hierarchical to:	No other components.
FIA_UAU.6.1/ SAM authentication	The TSF shall re-authenticate the user under the conditions: <ul style="list-style-type: none"> ▪ Beginning of Payment commands ▪ Beginning of Purse Management commands ▪ Beginning of File Management commands <p><i>Refinement:</i> <i>"User" stands for CPU.</i></p>
Dependencies:	No dependencies.

FIA_UID.2: USER IDENTIFICATION BEFORE ANY ACTION

<i>Hierarchical to:</i>	<i>FIA_UID.1</i>
<i>FIA_UID.2.1</i>	<i>The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	FIA_UID.1
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. <p><i>Application note:</i></p> <ul style="list-style-type: none"> ▪ <i>"User" means the ones associated to the packages (or applets) that act as subject of policies. In JCS, every action is always performed by an identified user interpreted here as the currently selected applet or the package that is the subject's owner. Means of identification are provided during</i>



	<p><i>the loading procedure of the package and the registration of applet instances.</i></p> <ul style="list-style-type: none"> ▪ <i>The S.JCRE defined in FMT_SMR.1 is attached to an IT security function rather than to a "user" in term of CC terminology. The JCRE doesn't identify itself with respect to the TOE, but it is a part of it.</i>
Dependencies:	No dependencies.

FIA_USB.1: USER-SUBJECT BINDING

Hierarchical to:	No other components.
FIA_USB.1.1	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: list of user security attributes] .
FIA_USB.1.2	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: rules for the initial association of attributes] .
FIA_USB.1.3	The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: rules for the changing of attributes] .
Dependencies:	FIA_ATD.1 User attribute definition

Hierarchical to:	No other components.
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 . <u>Application note:</u> <i>The user is the applet.</i> <i>The subject is the S.PACKAGE.</i>
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: none .
FIA_USB.1.3/ AID	The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: none .
Dependencies:	FIA_ATD.1 for JCS



6.1.6 FMT – Security management

FMT_MSA.1: MANAGEMENT OF SECURITY ATTRIBUTES

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FMT_MSA.1.1</i>	<i>The TSF shall enforce the [assignment: access control SFP(s), information flow control SFP(s)] to restrict the ability to [selection: change_default, query, modify, delete, [assignment: other operations]] the security attributes [assignment: list of security attributes] to [assignment: the authorized identified roles].</i>
<i>Dependencies:</i>	<i>FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1, FMT_SMF.1</i>

Hierarchical to:	No other components.
FMT_MSA.1.1	The TSF shall enforce the access control SFP(s) to restrict the ability to list of operations the security attributes list of security attributes to authorized identified roles . All is given in Table 29 – FMT_MSA.1.
Dependencies:	FDP_ACC.1 , FMT_SMR.1 , FMT_SMF.1 for JCS else FMT_SMF.1 is not supported (see §8.3.2 #7).

FMT_MSA.1				
Iteration	Access control SFP (s)	List of operations	List of security attributes	Authorized identified roles
FMT_MSA.1/ Confidentiality	▪ Confidentiality	▪ modify	▪ CTC ▪ Any application session key ▪ transaction value limit	▪ U.ISSUER
FMT_MSA.1/ Integrity	▪ Integrity	▪ modify	▪ CTC ▪ Any application session key ▪ transaction value limit	▪ U.ISSUER
FMT_MSA.1/ Life-cycle	▪ Life-cycle	▪ modify	▪ platform life-cycle ▪ Applet life-cycle state	▪ U.ISSUER
FMT_MSA.1/ Application	▪ Transaction	▪ modify	▪ any application session key ▪ transaction value limit	▪ U.ISSUER
FMT_MSA.1/ JCVM	▪ Firewall ▪ JCVM	▪ modify	▪ the Currently Active Context	▪ S.JCVM
FMT_MSA.1/ JCRE	▪ Firewall ▪ JCVM	▪ modify	▪ the Selected Applet Context	▪ S.JCRE

Table 29 – FMT_MSA.1



 FMT_MSA.2: SECURE SECURITY ATTRIBUTES

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FMT_MSA.2.1</i>	<i>The TSF shall ensure that only secure values are accepted for [assignment: list of security attributes].</i>
<i>Dependencies:</i>	<i>FDP_ACC.1 or FDP_IFC.1, FMT_MSA.1, FMT_SMR.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FMT_MSA.2.1/ JCRE	The TSF shall ensure that only secure values are accepted for security attributes defined in Firewall access control SFP and JCVM information flow control SFP.
<i>Dependencies:</i>	FDP_ACC.1 , FMT_MSA.1 , FMT_SMR.1

 FMT_MSA.3: STATIC ATTRIBUTE INITIALISATION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FMT_MSA.3.1</i>	<i>The TSF shall enforce the [assignment: access control SFP, information flow control SFP] to provide [selection, choose one of: restrictive, permissive, [assignment: other property]] default values for security attributes that are used to enforce the SFP.</i>
<i>FMT_MSA.3.2</i>	<i>The TSF shall allow the [assignment: the authorized identified roles] to specify alternative initial values to override the default values when an object or information is created.</i>
<i>Dependencies:</i>	<i>FMT_MSA.1, FMT_SMR.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FMT_MSA.3.1	The TSF shall enforce the access control SFP to provide property default values for security attributes that are used to enforce the SFP. All is given in Table 30 – FMT_MSA.3.
FMT_MSA.3.2	The TSF shall allow the authorized identified roles to specify alternative initial values to override the default values when an object or information is created.
<i>Dependencies:</i>	FMT_MSA.1 , FMT_SMR.1

FMT_MSA.3			
Iteration	Access control SFP (s)	Properties	Authorized identified roles
MFT_MSA3/ Confidentiality	▪ Confidentiality	▪ Restrictive	▪ no role
MFT_MSA3/ Integrity	▪ Integrity	▪ Restrictive	▪ no role
MFT_MSA3/ Life-cycle	▪ Life-cycle	▪ Restrictive	▪ no role
MFT_MSA3/ Application	▪ Transaction	▪ Restrictive	▪ no role
MFT_MSA3/ JCRE	▪ Firewall ▪ JCVM	▪ Restrictive	▪ not allow any role

Table 30 – FMT_MSA.3



 FMT_MTD.1: MANAGEMENT OF TSF DATA

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FMT_MTD.1.1</i>	<i>The TSF shall restrict the ability to [selection: change_default, query, modify, delete, clear, [assignment: other operations]] the [assignment: list of TSF data] to [assignment: the authorized identified roles].</i>
<i>Dependencies:</i>	<i>FMT_SMR.1, FMT_SMF.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FMT_MTD.1.1	The TSF shall restrict the ability to modify the APP ADMIN key, transaction value limit to the U.ISSUER .
<i>Dependencies:</i>	FMT_SMR.1 , FMT_SMF.1 for JCS else FMT_SMF.1 is not supported (see §8.3.2 #8).

 FMT_SMF.1: SPECIFICATION OF MANAGEMENT FUNCTIONS

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FMT_SMF.1.1</i>	<i>The TSF shall be capable of performing the following management functions: [assignment: list of management functions to be provided by the TSF].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FMT_SMF.1.1/ JCS	The TSF shall be capable of performing the following management functions: <ul style="list-style-type: none"> ▪ Modify the Currently Active Context and the Selected Applet Context.
<i>Dependencies:</i>	No dependencies.

 FMT_SMR.1: SECURITY ROLES

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FMT_SMR.1.1</i>	<i>The TSF shall maintain the roles [assignment: the authorized identified roles].</i>
<i>FMT_SMR.1.2</i>	<i>The TSF shall be able to associate users with roles.</i>
<i>Dependencies:</i>	<i>FIA_UID.1</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FMT_SMR.1.1	The TSF shall maintain the roles: <ul style="list-style-type: none"> ▪ U.ISSUER ▪ U.PURSEHOLDER ▪ S.JCRE ▪ S.JCVM <p>See Table 3 – Actors. Refinement:</p> <p>U.ISSUER and U.PURSEHOLDER are roles at application level. S.JCRE and S.JCVM are internal role to the platform.</p>
FMT_SMR.1.2	The TSF shall be able to associate users with roles.
<i>Dependencies:</i>	FIA_UID.1 for JCS else FIA_UID.1 is not supported (see §8.3.2 #9).



6.1.7 FPR – Privacy

FPR_UNO.1: UNOBSERVABILITY

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPR_UNO.1.1</i>	<i>The TSF shall ensure that [assignment: list of users and/or subjects] are unable to observe the operation [assignment: list of operations] on [assignment: list of objects] by [assignment: list of protected users and/or subjects].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FPR_UNO.1.1	<p>The TSF shall ensure that any users are unable to observe the operation cryptographic operations and key updates on keys by S.EP and S.CARD_MANAGER.</p> <p><u>Application note:</u></p> <p>"Observe the operation" stands for observe the processing linked to the operation in a way that could allow disclosing confidential information.</p> <p>"key updates" is only applied to APP ADMIN KEY.</p>
<i>Dependencies:</i>	<i>No dependencies.</i>



6.1.8 FPT – Protection of the TSF

FPT_FLS.1: FAILURE WITH PRESERVATION OF SECURE STATE

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_FLS.1.1</i>	<i>The TSF shall preserve a secure state when the following types of failure occur: [assignment: list of types of failure in the TSF].</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	No other components.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failure occur: <ul style="list-style-type: none"> ▪ Failure associated to potential security violations described in Table 19 – FAU_ARP.1 / FAU_SAA.1.
Dependencies:	No dependencies.

FPT_ITC.1: INTER-TSF CONFIDENTIALITY DURING TRANSMISSION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_ITC.1.1</i>	<i>The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorized disclosure during transmission.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	No other components.
FPT_ITC.1.1	The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorized disclosure during transmission. <i>Refinement:</i> <i>"Trusted IT product" stands for the Terminal.</i>
Dependencies:	No dependencies.

FPT_ITI.1: INTER-TSF DETECTION OF MODIFICATION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_ITI.1.1</i>	<i>The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric: [assignment: a defined modification metric].</i>
<i>FPT_ITI.1.2</i>	<i>The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform [assignment: action to be taken] if modifications are detected.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>



Hierarchical to:	No other components.
FPT_ITI.1.1	<p>The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another trusted IT product within the following metric:</p> <ul style="list-style-type: none"> ▪ TOKEN value associated to the Payment commands (DEBIT PURSE, CREDIT PURSE) ▪ MAC value associated to Purse Management commands (PUT DATA, WRITE LOCK) ▪ TOKEN or MAC value associated to File Management commands (READ/UPDATE/APPEND RECORD) ▪ TOKEN value associated to CHANGE KEY – APP ADMIN command <p><u>Refinement:</u></p> <p><i>"Trusted IT product" stands for the Terminal.</i></p>
FPT_ITI.1.2	<p>The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another trusted IT product and perform command abortion with emission of error status word if modifications are detected.</p> <p><u>Refinement:</u></p> <p><i>"Trusted IT product" stands for the Terminal.</i></p>
Dependencies:	No dependencies.

FPT_PHP.2: NOTIFICATION OF PHYSICAL ATTACK

<i>Hierarchical to:</i>	<i>FPT_PHP.1</i>
<i>FPT_PHP.2.1</i>	<i>The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.</i>
<i>FPT_PHP.2.2</i>	<i>The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.</i>
<i>FPT_PHP.2.3</i>	<i>For [assignment: list of TSF devices/elements for which active detection is required], the TSF shall monitor the devices and elements and notify [assignment: a designated user or role] when physical tampering with the TSF's devices or TSF's elements has occurred.</i>
<i>Dependencies:</i>	<i>FMT_MOF.1</i>

Hierarchical to:	FPT_PHP.1
FPT_PHP.2.1	The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.2.2	The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.
FPT_PHP.2.3	<p>For IC captors, the TSF shall monitor the devices and elements and notify the device when physical tampering with the TSF's devices or TSF's elements has occurred.</p> <p><u>Application note:</u></p>



	<i>The TSF shall rely on its certified IC [9] to detect physical attacks.</i>
Dependencies:	FMT_MOF.1 not satisfied (see §8.3.2 #10).

FPT_PHP.3: RESISTANCE TO PHYSICAL ATTACK

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_PHP.3.1</i>	<i>The TSF shall resist [assignment: physical tampering scenarios] to the [assignment: list of TSF devices/elements] by responding automatically such that the SFRs are always enforced.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	No other components.
FPT_PHP.3.1	The TSF shall resist the physical tampering scenarios to the list of TSF devices/elements by responding automatically such that the SFRs are always enforced.
Dependencies:	No dependencies.

FPT_PHP.3	
Physical tampering scenarios	List of TSF devices/elements
▪ Voltage supply	▪ The external voltage supply goes outside acceptable bounds
▪ Clock supply	▪ The external clock signal goes outside acceptable bounds
▪ Component	▪ The ambient temperature goes outside acceptable bounds
▪ CPU	▪ Application program abnormal runaway
▪ Component	▪ Attempts to physically probe the device
▪ RAM	▪ Attempts to gain illegal access to reserved RAM memory locations
▪ EEPROM	▪ Attempts to gain illegal access to EEPROM memory locations
▪ Reserved peripheral or registers	▪ Attempts to gain illegal access to reserved peripherals or IO register locations
▪ ROM	▪ Attempts to execute instructions to read the program memory from the non-supervisor program location
▪ RAM	▪ Attempts to move RAM stack to an illegal RAM memory location
▪ CPU	▪ Attempts to execute an AVR opcode that is not implemented
▪ EEPROM	▪ Attempts to illegally write access the device's EEPROM
▪ CPU	▪ Attempts to gain illegal access to supervisor modes
▪ Component	▪ The exposition to UV light goes outside acceptable bounds
▪ Reserved peripheral or registers	▪ Corruption
▪ AES accelerator	▪ Corruption / inhibition
▪ Crypto-processor	▪ Corruption / inhibition
▪ Unpredictable number generator	▪ Corruption / inhibition
▪ RAM	▪ Content corruption
▪ EEPROM	▪ Content corruption, transaction abortion ▪ Bad EEPROM program, erase or read sequence
▪ CPU	▪ Executable ROM code skipped or changed ▪ NVM ES code skipped or changed ▪ Stack overflow



▪ Internal clock	▪ Corruption / inhibition
------------------	---------------------------

Table 31 – FPT_PHP.3

FPT_RCV.4: FUNCTION RECOVERY

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_RCV.4.1</i>	<i>The TSF shall ensure that [assignment: list of functions and failure scenarios] have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FPT_RCV.4.1	The TSF shall ensure that Debit (DEBIT PURSE), Credit (CREDIT PURSE) and Parameters update (PUT DATA) transactions have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.
<i>Dependencies:</i>	<i>No dependencies.</i>

FPT_RPL.1: REPLAY DETECTION

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_RPL.1.1</i>	<i>The TSF shall detect replay for the following entities: [assignment: list of identified entities].</i>
<i>FPT_RPL.1.2</i>	<i>The TSF shall perform [assignment: list of specific actions] when replay is detected.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

<i>Hierarchical to:</i>	<i>No other components.</i>
FPT_RPL.1.1	The TSF shall detect replay for the following entities: <ul style="list-style-type: none"> - Debit transactions (DEBIT PURSE) - Credit transactions (CREDIT PURSE) - Parameters update transactions (PUT DATA)
FPT_RPL.1.2	The TSF shall perform the abort of the transaction in the process when replay is detected.
<i>Dependencies:</i>	<i>No dependencies.</i>

FPT_TDC.1: INTER-TSF BASIC TSF DATA CONSISTENCY

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_TDC.1.1</i>	<i>The TSF shall provide the capability to consistently interpret [assignment: list of TSF data types] when shared between the TSF and another trusted IT product.</i>
<i>FPT_TDC.1.2</i>	<i>The TSF shall use [assignment: list of interpretation rules to be applied by the TSF] when interpreting the TSF data from another trusted IT product.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>



Hierarchical to:	No other components.
FPT_TDC.1.1/ JCS	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/ JCS	The TSF shall use the rules defined in [JCVM221] specification, the API token defined in the export files of reference implementation when interpreting the TSF data from another trusted IT product.
Dependencies:	No dependencies.

FPT_TST.1: TSF TESTING

<i>Hierarchical to:</i>	<i>No other components.</i>
<i>FPT_TST.1.1</i>	<i>The TSF shall run a suite of self tests [selection: during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self test should occur]] to demonstrate the correct operation of [selection: [assignment: parts of TSF], the TSF].</i>
<i>FPT_TST.1.2</i>	<i>The TSF shall provide unauthorized users with the capability to verify the integrity of [selection: [assignment: parts of TSF], TSF data].</i>
<i>FPT_TST.1.3</i>	<i>The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.</i>
<i>Dependencies:</i>	<i>No dependencies.</i>

Hierarchical to:	No other components.
FPT_TST.1.1	The TSF shall run a suite of self tests during initial start-up to demonstrate the correct operation of the TSF .
FPT_TST.1.2	The TSF shall provide unauthorized users with the capability to verify the integrity of TSF data .
FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code .
Dependencies:	No dependencies.



6.2 Security assurance requirements (SARs)

The security assurance requirement level is EAL4 augmented (EAL4+) with

- ALC_DVS.2 (Development Security – Sufficiency of security measures)
- AVA_VAN.5 (Vulnerability Analysis – Advances methodical vulnerability analysis)

Assurance component		Description	Dependencies
EAL4 +			
Development	ADV_ARC.1	Security architecture description	ADV_FSP.1 ADV_TDS.1
	ADV_FSP.4	Complete functional specification	ADV_TDS.1
	ADV_IMP.1	Implementation representation of the TSF	ADV_TDS.3 ALC_TAT.1
	ADV_TDS.3	Basic modular design	ADV_FSP.4
Guidance	AGD_OPE.1	Operational user guidance	ADV_FSP.1
	AGD_PRE.1	Preparative procedures	No dependencies
Life-cycle support	ALC_CMC.4	Production support, acceptance procedures and automation	ALC_CMS.1 ALC_DVS.1 ALC_LCD.1
	ALC_CMS.4	Problem tracking CM coverage	No dependencies
	ALC_DEL.1	Delivery procedures	No dependencies
	ALC_DVS.2	Sufficiency of security measures	No dependencies
	ALC_LCD.1	Developer defined life-cycle model	No dependencies
	ALC_TAT.1	Well-defined development tools	ADV_IMP.1
Tests	ATE_COV.2	Analysis of coverage	ADV_FSP.2 ATE_FUN.1
	ATE_DPT.1	Testing security enforcing modules	ADV_ARC.1 ADV_TDS.3 ATE_FUN.1
	ATE_FUN.1	Functional testing	ATE_COV.1
	ATE_IND.2	Independent testing – sample	ADV_FSP.2 AGD_OPE.1 AGD_PRE.1 ATE_COV.1 ATE_FUN.1
Vulnerability assessment	AVA_VAN.5	Advanced methodical vulnerability analysis	ADV_ARC.1 ADV_FSP.2 ADV_TDS.3 ADV_IMP.1 AGD_OPE.1 AGD_PRE.1

Table 32 – List of SAR

6.3 Security functional requirements (SFRs) for IT environment

NA, the security objectives for the environment are only covered by organizational measures.

6.4 Composition tasks – Security requirements part

The objective is to determine if the composite-ST (this ST) doesn't contradict the platform-ST [10] in term of security functional requirements (SFR).



IC SFR Label	IC SFR content	IC SFR additional information	RP-SFR ⁷	IP-SFR ⁸	Link to composite-product
FAU_SAS.1	The TSF shall provide the test process before TOE Delivery with the capability to store the Initialization Data and/or Pre-personalization Data and/or supplements of the Security IC Embedded Software in the not changeable configuration page area and non-volatile memory.	None	X		No link to TOE SFRs but used for the composite-product identification.
FCS_COP.1 /3DES	The TSF shall perform encryption and decryption in accordance to a specified cryptographic algorithm Triple Data Encryption Standard (3DES) with cryptographic key sizes of 2 x 56 bit or 3 x 56 bit, that meet standards define in [10]	None	X		Not used then irrelevant for the present ST.
FCS_COP.1 / AES	The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm Advanced Encryption Standard (AES) algorithm and cryptographic key sizes of 128, 192 or 256 bit that meet standards define in [10]	None		X	FCS_COP.1
FCS_COP.1 / RSA	The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm Rivest-Shamir-Adleman (RSA) and cryptographic key sizes 1024 - 4096 bits that meet standards define in [10]	None	X		Not used then irrelevant for the present ST.
FCS_CKM.1/RSA	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm rsagen1 (PKCS v2.1 RFC3447) and specified cryptographic key sizes of 1024 – 4096 bits that meet standards define in [10]	None	X		Not used then irrelevant for the present ST.
FCS_COP.1/ECDSA	The TSF shall perform signature generation and	None	X		Not used then

⁷ RP-SFR means Relevant Platform-SFR.

⁸ IP-SFR means Irrelevant Platform-SFR.



IC SFR Label	IC SFR content	IC SFR additional information	RP-SFR ⁷	IP-SFR ⁸	Link to composite-product
	signature verification in accordance with a specified cryptographic algorithm ECDSA and cryptographic key sizes 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits that meet standards define in [10]				irrelevant for the present ST.
FCS_CKM.1/EC	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm Elliptic Curve EC specified in ANSI X9.62-2005 and ISO/IEC 15946-1:2002 and specified cryptographic key sizes 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits that meet standards define in [10]	None	X		Not used then irrelevant for the present ST.
FCS_COP.1/ECDH	The TSF shall perform elliptic curve Diffie-Hellman key agreement in accordance with a specified cryptographic algorithm ECDH and cryptographic key sizes 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits that meet standards define in [10]	None	X		Not used then irrelevant for the present ST.
FCS_RNG.1	The TSF shall provide a physical random number generator that implements: see [10] for details	None	X		FIA_SOS.2
FDP_ACC.1	The TSF shall enforce the Memory Access Control Policy on all subjects (software running at the defined and assigned privilege levels), all objects (data including code stored in memories) and all the operations defined in the Memory Access Control Policy, i.e. privilege levels.	Memory Access Control Policy The TOE shall control read, write, delete and execute accesses of software running at the privilege levels as defined below. Any access is controlled, regardless whether the access is on code or data or a jump on any other privilege level outside the current one. The memory model provides	X		FDP_ACC.2
FDP_ACF.1	The TSF shall enforce the Memory Access Control Policy to objects based on the following: Subject: software running at the IFX, OS1 and OS2 privilege levels required to securely operate the chip. This includes also privilege levels running interrupt routines.		X		FDP_ACF.1



IC SFR Label	IC SFR content	IC SFR additional information	RP-SFR ⁷	IP-SFR ⁸	Link to composite-product
	software running at the privilege levels containing the application software Object: data including code stored in memories Attributes: the memory area where the access is performed to and/or the operation to be performed.	distinct, independent privilege levels separated from each other in the virtual address space. These levels are referred to as the Infineon Technologies (IFX) level, operating system 1 and 2 levels (OS1, OS2), shared application level, and application 1 and 2 levels. A pseudo-level is the "current" level, which is simply the level on which code is currently being executed. The access rights are controlled by the MMU and related to the privilege level			
FMT_MSA.1	The TSF shall enforce the Memory Access Control Policy to restrict the ability to change default, modify or delete the security attributes permission control information to the software running on the privilege levels.		X		FMT_MSA.1
FMT_MSA.3	The TSF shall enforce the Memory Access Control Policy to provide well defined ¹ default values for security attributes that are used to enforce the SFP.		X		FMT_MSA.3
FMT_SMF.1	The TSF shall be capable of performing the following security management functions: access the configuration registers of the MMU.	None	X		FDP_ACC.2 FDP_ACF.1
FDP_SDI.1	The TSF shall monitor user data stored in containers controlled by the TSF for inconsistencies between stored data based on the following attributes: EDC value for ROM and RAM smart parity for the cache and ECC value for the SOLID FLASH™ NVM and verification of stored data in SOLID FLASH™ NVM.	None	X		FDP_SDI.1
FDP_SDI.2	The TSF shall monitor user data stored in containers controlled by the TSF for data integrity and one-and/or more-bit-errors on all objects, based on the following attributes: corresponding EDC value for ROM and RAM, smart parity for Cache and error correction ECC for the SOLID FLASH™ NVM.	None	X		FDP_SDI.2
FDP_IFC.1	The TSF shall enforce the Data Processing Policy on	Data Processing Policy	X		FDP_ACC.2 /



IC SFR Label	IC SFR content	IC SFR additional information	RP-SFR ⁷	IP-SFR ⁸	Link to composite-product	
	all confidential data when they are processed or transferred by the TSF or by the Security IC Embedded Software.	User Data and TSF data shall not be accessible from the TOE except when the Security IC 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 Security IC Embedded Software.			ACF.1 FDP_IFC.2 / IFF.1 FPR_UNO.1	
FDP_ITT.1	The TSF shall enforce the Data Processing Policy to prevent the disclosure of user data when it is transmitted between physically-separated parts of the TOE.		X		FDP_ACC.2 / ACF.1 FPR_UNO.1	
FPT_ITT.1	The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE. 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.		X		No direct link to any composite-product SFR	
FMT_LIM.1	The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Limited capability and availability Policy.		Limited capability and availability Policy Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.	X		FDP_ACC.2 / ACF.1 FDP_IFC.2 / IFF.1
FMT_LIM.2	The TSF shall be designed and implemented in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: Limited capability and availability Policy.		X		FDP_ACC.2 / ACF.1 FDP_IFC.2 / IFF.1	
FPT_TST.2	The TSF shall run a suite of self tests at the request of the authorised user to demonstrate the correct operation of the alarm lines and/or following environmental sensor mechanisms: [...]		None	X		FAU_ARP.1 FAU_SAA.1 FPT_TST.1
FPT_FLS.1	Failure with preservation of secure state: The TSF shall preserve a secure state when the following types of failures occur: exposure to operating		None	X		FPT_FLS.1 FPT_PHP.3



IC SFR Label	IC SFR content	IC SFR additional information	RP-SFR ⁷	IP-SFR ⁸	Link to composite-product
	conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur.				
FRU_FLT.2	Limited fault tolerance: 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).	None	X		FPT_FLS.1 FPT_PHP.3
FPT_PHP.3	The TSF shall resist physical manipulation and physical probing, to the TSF by responding automatically such that the SFRs are always enforced.	None	X		FPT_FLS.1 FPT_PHP.3

Table 33 – Composition – Security requirements part



7 TOE Summary Specification

7.1 TOE security functions

HW Security function	Description
SF_DPM	Device Phase Management
SF_PS	Protection against Snooping
SF_PMA	Protection against Modification Attacks
SF_PLA	Protection against Logical Attacks
SF_CS	Cryptographic Support
SW Security function	Description
F.SW_REACTION	Reaction management
F.SW_CRYPTO_OPERATION	Cryptographic operation management
F.SW_ACCESS_CONTROL	Access control management
F.SW_LOGGING	Logging management
F.SW_INTEGRITY	Integrity Protection
F.SW_CONFIDENTIALITY	Confidentiality Protection
F.SW_AUTHENTICATION	Authentication

Table 34 – List of Security functions

7.1.1 HW security functions

See document [\[10\]](#) for descriptions and details.

7.1.2 SW security functions

1. F.SW_REACTION

The **F.SW_REACTION** function allows to:

- Manage the policy of attacks reaction according to the security violations (FAU_ARP.1, FAU_SAA.1, FDP_SDI.2)
- Manage the physical attacks (FPT_PHP.2, FPT_PHP.3)
- Manage the automatic self-tests (FPT_TST.1)
- Manage a secure state according to the execution flows (FPT_FLS.1, FPT_RCV.4)

2. F.SW_CRYPTO_OPERATION

The **F.SW_CRYPTO_OPERATION** function allows to:

- Manage the creation/distribution and the access/deletion of cryptographic keys (FCS_CKM.1/Session, FCS_CKM.2/AES, FCS_CKM.3/AES, FCS_CKM.4)
- Manage the cryptographic operations in AES symmetric mode (FCS_COP.1/Session, FCS_COP.1/MAC, FCS_COP.1/Token)
- Manage the generation of challenges (FIA_SOS.2)
- Crypto protection (FPR_UNO.1)

3. F.SW_ACCESS_CONTROL

The **F.SW_ACCESS_CONTROL** function allows to:



- Manage the Atomicity mechanism (FDP_ACC.1/Atomicity, FDP_ROL.1/Atomicity)
- Manage the application and card life-cycles (FDP_ACC.1/Life-cycle, FDP_ACF.1/Life-cycle, FMT_MSA.1/Life-cycle, FMT_MSA.3/Life-cycle)
- Manage the Payment transactions and limits (FAU_STG.1, (FDP_IFC.1/Application, FDP_IFF.1/Application, FDP_ITC.1/Application, FMT_MSA.3/Application)
- Manage the JCVM/JCRE (FDP_ACC.2/Firewall, FDP_ACF.1/Firewall, FDP_IFC.1/JCVM, FDP_IFF.1/JCVM, FDP_ROL.1/Firewall, FIA_ATD.1/AID, FIA_UID.2/AID, FIA_USB.1/AID, FMT_MSA.1/JCVM, FMT_MSA.1/JCRE, FMT_MSA.2/JCRE, FMT_MSA.3/JCRE, FMT_SMF.1/JCS, FMT_SMR.1)
- Replay protection (FPT_ITC.1, FPT_ITI.1, FPT_RPL.1)
- Manage the protection of residual information (FDP_RIP.1/JCS, FDP_RIP/Key) and the exchange of data (FDP_UCT.1, FDP_UIT.1)

4. F.SW_LOGGING

The **F.SW_LOGGING** function allows to:

- Manage the logging of Payment transactions (FAU_GEN.1, FAU_SAR.1, FAU_STG.1)
- Logging protection (FPT_ITI.1)

5. F.SW_INTEGRITY

The **F.SW_INTEGRITY** function allows to:

- Manage the Integrity access control (FMT_MSA.1/Integrity, FMT_MSA.3/Integrity, FMT_MTD.1)
- Manage the integrity of assets (FDP_ACC.2/Integrity, FDP_ACF.1/Integrity, FDP_ITC.1/Application, FDP_RIP.1/JCS, FDP_RIP.1/Key, FDP_UIT.1, FMT_MTD.1)
- Integrity protection (FDP_SDI.2, FPT_ITI.1)

6. F.SW_CONFIDENTIALITY

The **F.SW_CONFIDENTIALITY** function allows to:

- Manage the Confidentiality access control (FMT_MSA.1/Confidentiality, FMT_MSA.3/Confidentiality)
- Manage the confidentiality of assets (FDP_ACC.2/Confidentiality, FDP_ACF.1/ Confidentiality, FDP_ITC.1/Application, FDP_RIP.1/JCS, FDP_RIP.1/Key, FDP_UCT.1, FPR_UNO.1)
- Confidentiality protection (FPT_ITC.1)

7. F.SW_AUTHENTICATION

The **F.SW_AUTHENTICATION** function allows to:

- Manage the distribution of cryptographic keys (FCS_CKM.3/AES)
- Manage the CPU authentication by the Terminal (FCO_NRO.2/CPU authentication, FDP_IFC.1/Application, FIA_UAU.1/CPU authentication, FIA_UAU.4/CPU authentication, FIA_UAU.6/CPU authentication)
- Manage the SAM authentication by the CPU (FCO_NRO.2/SAM authentication, FDP_IFC.1/Application, FIA_UAU.1/SAM authentication, FIA_UAU.4/SAM authentication, FIA_UAU.6/SAM authentication)
- Authentication (FDT_ITC.1, FPT_ITI.1)

7.2 Assurance measures

Assurance measure	Description
CPU.ASE	This assurance is given by the ST document (this present document) that describes the TOE to evaluate.
→ ST is provided to Easy Card Corporation.	



Assurance measure	Description
	<p>→ The ST document is given to security evaluation laboratory (accredited by FSC) for analysis.</p>
CPU.ADV	<p>This assurance is given by the Gemalto development system (Development life-cycle). More precisely document that ensure that HW documents (IC datasheet, IC guidance and recommendation for the OS developer, OS countermeasures, etc...) are taken into account during OS development.</p> <p>→ The documents related to development and composition of SW with HW, are given to security evaluation laboratory (accredited by FSC) for analysis.</p>
CPU.ADV_IMP	<p>This assurance is given by source codes.</p> <p>→ The security evaluation laboratory (accredited by FSC) performs codes review.</p>
CPU.AGD → Transaction Flow, Functional specifications and Personalization specifications are provided to Easy Card Corporation.	<p>This assurance is given by specification of all commands: description of APDU, Status Word, etc...</p> <p>→ The specifications are given to security evaluation laboratory (accredited by FSC) for analysis.</p>
CPU.ALC	<p>This assurance is given by the Gemalto development system (Development life-cycle, Configuration management system, Secure development environment, masking process between Gemalto and IC manufacturer, etc...).</p> <p>→ The proof is given to security evaluation laboratory (accredited by FSC) that mask, source codes and samples are identical. The security evaluation laboratory (accredited by FSC) performs audit on development system environment.</p>
CPU.ATE → Test results are provided to Easy Card Corporation.	<p>This assurance is given by the Gemalto development system (Development life-cycle).</p> <p>→ The documents related to tests are given to security evaluation laboratory (accredited by FSC) for analysis.</p>
CPU.AVA → Samples are provided to Easy Card Corporation.	<p>This assurance is given by production of samples corresponding to the TOE to evaluate.</p> <p>→ The security evaluation laboratory (accredited by FSC) performs physical testing (Side channel and Fault perturbation).</p>

Table 35 – List of Assurance measures



8 Annex

8.1 Acronyms

AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
ATR	Answer To Reset
EAL	Evaluation Assurance Level
ECC	Easy Card Corporation
EEPROM	Electrically Erasable Programmable Read-Only Memory
HW	Hardware
ICC	Integrated Circuit Card
ISO	International Organization for Standardization
OS	Operating System
PP	Protection Profile
PPS	Protocol and Parameter Selection
PSN	Purse Serial Number
ROM	Read-Only Memory
ST	Security Target
SW	Software
TOE	Target Of Evaluation

8.2 Security objectives rationale

8.2.1 Assets and Protection – Coverage

Assets vs Security	Integrity	Confidentiality	Correct execution
D.EM	X		
D.EP_IVDATA	X		
D.EP_CODE	X	X	X
D.KEYS	X	X	
D.EP_STATE	X		
D.LOG_DATA	X		
D.COUNTERS	X		

Table 36 – Assets and Protection – Coverage



8.2.2 Threats/OSP/Assumptions and Assets – Coverage

Threats/OSP/Assumptions vs Assets	D.EM	D.EP_IVDATA	D.EP_CODE	D.KEYS	D.EP_STATE	D.LOG_DATA	D.COUNTERS
T.COUNTERFEITING_DEBIT	X						
T.COUNTERFEITING_CREDIT	X						
T.COUNTERFEITING_AUTOLOAD	X						
T.COUNTERFEITING_AUTH							X
T.COUNTERFEITING_UPDATE				X			X
T.DISCLOSURE_KEYS				X			
T.INTEG_EM	X						
T.INTEG_EP_IVDATA		X					
T.INTEG_CODE			X				
T.INTEG_KEYS				X			
T.INTEG_EP_STATE					X		
T.INTEG_LOG_DATA						X	
T.INTEG_COUNTERS							X
T.REPLAY_DEBIT	X						
T.REPLAY_CREDIT	X						
T.REPLAY_AUTOLOAD	X						
T.REPLAY_UPDATE				X			X
T.Separation	X	X	X	X	X	X	X
OSP.SECRET_MNGT	X	X	X	X	X	X	X
OSP.DEBIT_BEFORE_CREDIT	X						
OSP.PURSE_BEHAVIOR	X						
A.PHYSICAL	X	X	X	X	X	X	X
A.PROTECTION_AFTER_TOE_DELIVERY	X	X	X	X	X	X	X



Table 37 – Threats/OSP/Assumptions and Assets – Coverage

8.2.3 Threats/OSP/Assumptions and Security Objectives – Coverage

Security objectives vs Threats/OSP/Assumptions	T.COUNTERFEITING_DEBIT	T.COUNTERFEITING_CREDIT	T.COUNTERFEITING_AUTOLOAD	T.COUNTERFEITING_AUTH	T.COUNTERFEITING_UPDATE	T.DISCLOSURE_KEYS	T.INTEG_EM	T.INTEG_EP_IVDATA	T.INTEG_CODE	T.INTEG_KEYS	T.INTEG_EP_STATE	T.INTEG_LOG_DATA	T.INTEG_COUNTERS	T.REPLAY_DEBIT	T.REPLAY_CREDIT	T.REPLAY_AUTOLOAD	T.REPLAY_UPDATE	T.Separation	OSP.SECRET_MNGT	OSP.DEBIT_BEFORE_CREDIT	OSP.PURSE_BEHAVIOR	A.PHYSICAL	A.PROTECTION_AFTER_TOE_DELIVERY
O.AUTH	X	X	X	X	X																		
O.EM	X	X	X				X							X	X	X							
O.CONF_DATA						X																	
O.INTEG_DATA							X	X	X	X	X	X	X										
O.OPERATE							X	X	X	X	X	X	X										
O.REPLAY														X	X	X	X						
O.TAMPER						X	X	X	X	X	X	X	X										
O.RECORD	X	X	X									X											
O.Atomicity	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					
O.Separation						X	X	X	X	X	X	X	X					X					
OE.SECRET_MNGT																			X				
OE.DEBIT_BEFORE_CREDIT																				X			
OE.PURSE_BEHAVIOR																					X		





OE.PHYSICAL																							X		
OE.PROTECTION_AFTER_TOE_DELIVERY																									X

Table 38 – Threats/OSP/Assumptions and Security Objectives – Coverage

Rationale:

T.COURTEFEITING_DEBIT/CREDIT/AUTOLOAD: These threats are countered by,

- O.AUTH that requires the authentication of both the TOE and the external device before performing any Debit Purse or Credit Purse transactions
- O.EM which ensure EM flow preservation so that fraudulent creation of EM in the EP using Debit or Credit operations is not possible
- O.RECORD which ensures that the TOE records necessary events and data (flow traceability data) in order to be presented again as an element of evidence of the real transaction
- O.Atomicity which provide a means to perform memory operations atomically

T.COURTEFEITING_AUTH/UPDATE: These threats are countered by,

- O.AUTH that requires the authentication of both the TOE and the external device before performing any transaction
- O.Atomicity which provide a means to perform memory operations atomically

T.DISCLOSURE_KEYS: This threat is countered by,

- O.CONFID_DATA which ensures the confidentiality of D.KEYS
- O.TAMPER which ensures security information and especially D.KEYS cannot be physically tampered
- O.Atomicity which provide a means to perform memory operations atomically
- O.Separation which provide a means to control the access to services and resources

T.INTEG_EM: This threat is countered by,

- O.EM which ensure EM flow preservation so that fraudulent creation of EM in the EP is not possible
- O.INTEG_DATA which ensures the integrity of D.EM
- O.OPERATE which ensure the correct operations of the related transactions
- O.TAMPER which ensures security information and especially assets cannot be physically tampered
- O.Atomicity which provide a means to perform memory operations atomically
- O.Separation which provide a means to control the access to services and resources

T.INTEG_EP_IVDATA /CODE/KEYS/ EP_STATE/COUNTERS: This threat is countered by,

- O.INTEG_DATA which ensures the integrity of D.EP_IV_DATA, D.EP_CODE, D.KEYS, D.EP_STATE and D.COUNTERS
- O.OPERATE which ensure the correct operations of the related transactions
- O.TAMPER which ensures security information and especially assets cannot be physically tampered





- O.Atomicity which provide a means to perform memory operations atomically
- O.Separation which provide a means to control the access to services and resources

T.INTEG_LOG_DATA: This threat is countered by,

- O.INTEG_DATA which ensures the integrity of D.LOG_DATA
- O.OPERATE which ensure the correct operations of the related transactions
- O.TAMPER which ensures security information and especially assets cannot be physically tampered
- O.RECORD which ensures that the TOE records necessary events and data (flow traceability data) in order to be presented again as an element of evidence of the real transaction
- O.Atomicity which provide a means to perform memory operations atomically
- O.Separation which provide a means to control the access to services and resources

T.REPLAY_DEBIT/CREDIT/AUTOLOAD: This threat is countered by,

- O.EM which ensure EM flow preservation so that fraudulent creation of EM in the EP using Debit or Credit operations is not possible
- O.REPLAY which ensure the EP will operate in a continuous secure state in case of replayed Debit or Credit operations will be detected and rejected by the EP
- O.Atomicity which provide a means to perform memory operations atomically

T.REPLAY_UPDATE: This threat is countered by,

- O.REPLAY which ensure the EP will operate in a continuous secure state in case of replayed parameters update operation will be detected and rejected by the EP
- O.Atomicity which provide a means to perform memory operations atomically

T.Separation: This threat is countered by,

- O.Atomicity which provide a means to perform memory operations atomically
- O.Separation which provide a means to control the access to services and resources

OSP.SECRET_MNGT:

- OE.SECRET_MNGT directly covers the OSP.

OSP.DEBIT_BEFORE_CREDIT:

- OE.DEBIT_BEFORE_CREDIT directly covers the OSP.

OSP.PURSE_BEHAVIOR:





- OE.PURSE_BEHAVIOR directly covers the OSP.

A.PHYSICAL:

- OE.PHYSICAL directly covers the Assumption.

A.PROTECTION_AFTER_TOE_DELIVERY:

- OE.PROTECTION_AFTER_TOE_DELIVERY directly covers the Assumption.



8.3 Security requirements rationale

8.3.1 Security Objectives and SFR – Coverage

SFRs vs Security objectives		O.AUTH	O.EM	O.CONF_DATA	O.INTEG_DATA	O.OPERATE	O.REPLAY	O.TAMPER	O.RECORD	O.Atomicity	O.Separation
FAU	FAU_ARP.1							X			
	FAU_GEN.1								X		
	FAU_SAA.1							X			
	FAU_SAR.1								X		
	FAU_STG.1		X		X	X		X	X		X
FCO	FCO_NRO.2/CPU authentication		X								
	FCO_NRO.2/SAM authentication		X								
FCS	FCS_CKM.1/Session	X					X				
	FCS_CKM.2/AES	X									X
	FCS_CKM.3/AES	X									X
	FCS_CKM.4	X	X	X			X				
	FCS_COP.1/Session	X					X				
	FCS_COP.1/MAC	X	X	X	X		X				





SFRs vs Security objectives		O.AUTH	O.EM	O.CONF_DATA	O.INTEG_DATA	O.OPERATE	O.REPLAY	O.TAMPER	O.RECORD	O.Atomicity	O.Separation
	FCS_COP.1/Token	X	X	X	X						
FDP	FDP_ACC.1/Atomicity									X	
	FDP_ACC.1/Life-cycle					X					
	FDP_ACC.2/Confidentiality			X							
	FDP_ACC.2/Integrity		X		X						
	FDP_ACC.2/Firewall										X
	FDP_ACF.1/Life-cycle					X					
	FDP_ACF.1/Firewall										X
	FDP_ACF.1/Confidentiality			X							
	FDP_ACF.1/Integrity		X		X						
	FDP_IFC.1/JCVM										X
	FDP_IFC.1/Application		X								
	FDP_IFF.1/JCVM										X
	FDP_IFF.1/Application		X								
	FDP_ITC.1/Application		X								
FDP_RIP.1/JCS											X





SFRs vs Security objectives		O.AUTH	O.EM	O.CONF_DATA	O.INTEG_DATA	O.OPERATE	O.REPLAY	O.TAMPER	O.RECORD	O.Atomicity	O.Separation
	FDP_RIP.1/Key			X							X
	FDP_ROL.1/Firewall										X
	FDP_ROL.1/Atomicity									X	
	FDP_SDI.2		X		X						
	FDP_UCT.1			X							
	FDP_UIT.1		X		X						
FIA	FIA_ATD.1/AID										X
	FIA_SOS.2	X					X				
	FIA_UAU.1/CPU authentication	X									
	FIA_UAU.1/SAM authentication	X									
	FIA_UAU.3	X									
	FIA_UAU.4/CPU authentication	X									
	FIA_UAU.4/SAM authentication	X									
	FIA_UAU.6/CPU authentication	X									
	FIA_UAU.6/SAM authentication	X									
	FIA_UID.2/AID										X





SFRs vs Security objectives		O.AUTH	O.EM	O.CONF_DATA	O.INTEG_DATA	O.OPERATE	O.REPLAY	O.TAMPER	O.RECORD	O.Atomicity	O.Separation
	FIA_USB.1/AID										X
FMT	FMT_MSA.1/Confidentiality			X							
	FMT_MSA.1/Integrity		X		X						
	FMT_MSA.1/Life-cycle					X					
	FMT_MSA.1/Application		X								
	FMT_MSA.1/JCVM										X
	FMT_MSA.1/JCRE										X
	FMT_MSA.2/JCRE										X
	FMT_MSA.3/Confidentiality			X							
	FMT_MSA.3/Integrity		X		X						
	FMT_MSA.3/Life-cycle					X					
	FMT_MSA.3/Application		X								
	FMT_MSA.3/JCRE										X
	FMT_MTD.1					X					
	FMT_SMF.1/JCS										X
FMT_SMR.1		X	X	X	X	X	X	X	X	X	X
FPR	FPR_UNO.1			X				X			





SFRs vs Security objectives		O.AUTH	O.EM	O.CONF_DATA	O.INTEG_DATA	O.OPERATE	O.REPLAY	O.TAMPER	O.RECORD	O.Atomicity	O.Separation
FPT	FPT_FLS.1					X					
	FPT_ITC.1			X							
	FPT_ITI.1		X		X						
	FPT_PHP.2		X	X	X			X			
	FPT_PHP.3		X	X	X			X			
	FPT_RCV.4		X	X	X	X					
	FPT_RPL.1		X				X				
	FPT_TDC.1/JCS										X
	FPT_TST.1		X		X	X			X		

Table 39 – Security Objectives and SFR – Coverage

Rationale:

O.AUTH is covered by:





- FCS_COP.1, FCS_CKM.1, FCS_CKM.2, FCS_CKM.3 and FCS_CKM.4 that specify the cryptographic operations and the key destruction mechanism
- FIA_SOS.2 that provides a random number generation mechanism
- FIA_UAU.1/CPU authentication and SAM authentication that requires the authentication of the corresponding each time a transaction needs to be initiated
- FIA_UAU.3 that prevents against use of authentication data
- FIA_UAU.4/CPU authentication and SAM authentication that prevents against reuse of authentication data
- FIA_UAU.6/CPU authentication and SAM authentication that requires the re-authentication of the corresponding each time a transaction needs to be (re)initiated
- FMT_SMR.1 that states the authorized role

O.EM is covered by:

- FAU_STG.1 that protects the audit record stored in the TOE against unauthorized detection and detects any attack against the asset
- FCO_NRO.2/CPU authentication and SAM authentication that ensure proof of validity of loaded EM
- FCS_COP.1/MAC/Token and FCS_CKM.4 that specify the characteristics of cryptographic operations and key destruction mechanism the Application access control policy
- FDP_IFC.1/IFF.1 and FMT_MSA.1/MSA.3 and FDP_ITC.1 for Application that define the security policy enforcing access control
- FDP_ACC.2/ACF.1 and FMT_MSA.1/MSA.3 for Integrity that define the security policy enforcing access control of integer data
- FDP_SDI.2 that specifies the data that is monitored and the response of the TOE, Data belongs to the security policy enforcing access control of integer data
- FDP_UIT.1 and FPT_ITI.1 that provide minimum requirements for protected communication of integer data where access control policy for integrity relies on
- FMT_SMR.1 that states the authorized role
- FPT_PHP2 and FPT_PHP.3 which requires detection and protection against physical tampering
- FPT_RCV.4 that ensures the TOE cannot be modified and put in a inconsistent state which could alter the integrity of the assets
- FPT_RPL.1 that ensures that transactions are protected against replay: the TSF can detect and react
- FPT_TST.1 that contributes to the integrity protection of data

O.CONF_DATA is covered by:

- FDP_ACC.2/ACF.1 and FMT_MSA.1/MSA.3 for Confidentiality that define the security policy enforcing access control of confidential data
- FCS_COP.1/MAC/Token and FCS_CKM.4 that specify the characteristics of cryptographic operations and key destruction mechanism the access control policy for confidentiality relies on
- FDP_RIP.1/Key that ensures that no residual confidential data is available
- FMT_SMR.1 that states the authorized role
- FPR_UNO.1 that enforces the unobservability of confidential data processing
- FPT_PHP2 and FPT_PHP.3 which requires detection and protection against physical tampering
- FPT_RCV.4 that ensures the TOE cannot be modified and put in a inconsistent state which could alter the integrity of the assets



O.INTEG_DATA is covered by:

- FAU_STG.1 that protects the audit record stored in the TOE against unauthorized detection and detects any attack against the asset
- FDP_ACC.2/ACF.1 and FMT_MSA.1/MSA.3 for Integrity that define the security policy enforcing access control of integer data
- FCS_COP.1/MAC/Token that specify the characteristics of cryptographic operations the access control policy for integrity relies on
- FDP_SDI.2 that specifies the data that is monitored and the response of the TOE, Data belongs to the security policy enforcing access control of integer data
- FDP_UIT.1 and FPT_ITI.1 that provide minimum requirements for protected communication of integer data where access control policy for integrity relies on
- FMT_SMR.1 that states the authorized role
- FPT_PHP2 and FPT_PHP.3 which requires detection and protection against physical tampering
- FPT_RCV.4 that ensures the TOE cannot be modified and put in a inconsistent state which could alter the integrity of the assets
- FPT_TST.1 that contributes to the integrity protection of data as well as code

O.OPERATE is covered by:

- FAU_STG.1 that protects the audit record stored in the TOE against unauthorized detection and detects any attack against the asset
- FDP_ACC.1/ACF.1 and FMT_MSA.1/MSA.3 for Lifecycle that define the security policy enforcing access control
- FMT_MTD.1 that requires controlled access to TSF data allowed only to authorized roles
- FMT_SMR.1 that states the authorized role
- FPT_FLS.1 that requires that failure do not impact the security of the TOE
- FPT_RCV.4 that ensures the TOE cannot be modified and put in a inconsistent state which could alter the integrity of the assets
- FPT_TST.1 that contributes to the integrity protection of data

O.REPLAY is covered by:

- FCS_CKM.1/Session, FCS_COP.1/Session/MAC and FCS_CKM.4 that specify the characteristics of cryptographic operations and the key destruction mechanism
- FIA_SOS.2 that provides a random number generation mechanism
- FMT_SMR.1 that states the authorized role
- FPT_RPL.1 that ensures that transactions are protected against replay: the TSF can detect and react

O.TAMPER is covered by:

- FAU_SAA.1 that specifies the events that are audited for detection of potential security violations made available by the IC and FAU_ARP.1 that response to potential security violations
- FAU_STG.1 that protects the audit record stored in the TOE against unauthorized detection and detects any attack against the asset
- FMT_SMR.1 that states the authorized role



- FPR_UNO.1 that enforces the unobservability of confidential data processing
- FPT_PHP2 and FPT_PHP.3 which requires detection and protection against physical tampering

O.RECORD is covered by:

- FAU_GEN.1 that requires the generation of an audit record
- FAU_SAR.1 that allows the capability to read the audit record
- FAU_STG.1 that protects the audit record stored in the TOE against unauthorized detection and detects any attack against the asset
- FMT_SMR.1 that states the authorized role
- FPT_TST.1 that contributes to the protection of data

O.Atomicity is covered by:

- FDP_ACC.1/Atomicity that defines the memory operations for which rollback operation is allowed
- FDP_ROL.1/Atomicity that states the rules for rollback that leaves the TOE in a secure state
- FMT_SMR.1 that states the authorized role

O.Separation is ensured by Javacard system firewall (Firewall, JCS, JCVM, AID, JCRE).

- FAU_STG.1 that protects the audit record stored in the TOE against unauthorized detection and detects any attack against the asset
- FCS_CKM.2 and FCS_CKM.3 that specify the cryptographic operations
- FDP_ACC.2/ACF.1 for Firewall and FDP_IFCC.1/IFF.1 for JCVM define the security policy enforcing access control
- FDP_RIP.1 that protect the residual information
- FDP_ROL.1/Firewall that preserve the integrity of data for rollback
- FIA_ATD.1, FIA_UID.2 and FIA_USB.1 for AID that ensure firewalling
- FMT_MSA.3/JCRE and FMT_SMF.1/JCS that manage firewalling and FMT_SMR.1 that states the authorized role



8.3.2 SFR dependencies – Coverage

All SFR dependencies are covered except:

#1/ The dependency FPT_STM.1 of FAU_GEN.1 is unsupported. The dependency with FPT_STM.1 is not relevant to the TOE: correctness of time is not use for the TOE objectives.

#2/ The dependency FIA_UID.1 of FCO_NRO.2 is unsupported. The dependency with FIA_UID.1 is not relevant to the TOE: there is no identification to the TOE. It is implicitly performed when the physical device is presented.

#3/ The dependency FDP_ACF.1/Atomicity of FDP_ACC.1/Atomicity is unsupported. The dependency with FDP_ACF.1/Atomicity is not relevant to the TOE. The FDP_ACC requirement serves as a framework for the definition of the operations that can be rolled back. There is no need to require FDP_ACF since there is neither mandatory attribute nor rule.

#4/ The dependency FTP_ITC.1 or FTP_TRP.1 of FDP_UCT.1 is unsupported. The requirement FDP_UCT.1 allows stating the property that confidential data shall enter and leave the TOE according to the Confidentiality Access Control SFP. To mandate any TOE capacity to establish trusted paths or trusted channels is not relevant to the TOE.

#5/ The dependency FTP_ITC.1 or FTP_TRP.1 of FDP_UIT.1 is unsupported. The requirement FDP_UIT.1 allows stating the property that integer data shall enter and leave the TOE according to the Integrity Access Control SFP. To mandate any TOE capacity to establish trusted paths or trusted channels is not relevant to the TOE.

#6/ The dependency FIA_UID.1 of FIA_UAU.1/SAM is unsupported. The dependency with FIA_UID.1 is not relevant to the TOE: there is no identification to the TOE. It is always the TOE which identifies itself to other devices.

#7/ The dependency FMT_SMF.1 of FMT_MSA.1 is unsupported. The dependency with FMT_SMF.1 is not relevant to the TOE. To mandate any particular security management functions is not relevant to the TOE. Standard CC operations for the management of security attributes are sufficient for the present ST.

#8/ The dependency FMT_SMF.1 of FMT_MTD.1 is unsupported. The dependency with FMT_SMF.1 is not relevant to the TOE. To mandate any particular security management functions is not relevant to the TOE.

#9/ The dependency FIA_UID.1 of FMT_SMR.1 is unsupported. The dependency with FIA_UID.1 is not relevant to the TOE: there is no identification to the TOE. It is implicitly performed when the physical device is presented.

#10/ The dependency FMT_MOF.1 of FPT_PHP.2 is unsupported. The dependency with FMT_MOF.1 is not relevant to the TOE: during operational use of the TOE, the behavior security functions could not be changed.

8.3.3 SAR – Rationale

All SARs are drawn from the CC V3.1 R3 [1][2][3] and CEM [4].

Augmentations are in line with the requirement high resistance against state-of-the-art attacks of an e-purse smartcard (in line with the Moneo Electronic Purse Protection Profile). Augmentation results from the selection of ALC_DVS.2 and AVA_VAN.5.



8.3.4 SAR dependencies – Coverage

All SAR dependencies are covered.

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



AVA_VAN.5	(ADV_ARC.1) and (ADV_FSP.4) and (ADV_IMP.1) and (ADV_TDS.3) and (AGD_OPE.1) and (AGD_PRE.1) and (ATE_DPT.1)	ADV_ARC.1, ADV_FSP.4, ADV_IMP.1, ADV_TDS.3, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1
-----------	---	---

Table 40 – SAR dependencies – Coverage

8.4 TOE summary specification rationale



8.4.1 SAR and Assurances Measures – Coverage

SARs vs Assurance Measures		CPU_ASE	CPU_ADV	CPU_ADV_IMP	CPU_AGD	CPU_ALC	CPU_ATE	CPU_AVA
ASE	ASE_CCL.1	X						
	ASE_ECD.1	X						
	ASE_INT.1	X						
	ASE_OBJ.2	X						
	ASE_REQ.2	X						
	ASE_SPD.1	X						
	ASE_TSS.1	X						
ADV	ADV_ARC.1		X					
	ADV_FSP.4		X					
	ADV_IMP.1			X				
	ADV_TDS.3		X					
AGD	AGD_OPE.1				X			
	AGD_PRE.1				X			
ALC	ALC_CMC.4					X		
	ALC_CMS.4					X		
	ALC_DEL.1					X		
	ALC_DVS.2					X		
	ALC_LCD.1					X		
	ALC_TAT.1					X		
ATE	ATE_COV.2						X	
	ATE_DPT.2						X	
	ATE_FUN.1						X	
	ATE_IND.2						X	
AVA	AVA_VAN.5							X

Table 41 – SAR and Assurances Measures – Coverage

- FIN DU DOCUMENT -