台灣銘板股份有限公司

Taiwan Name Plate Co., Ltd. (TNP).

TNP ECC2 CPU Card

Security Target

Version 1.0.1

Revision History

Revision	Author	Date	Modification
V0.7	Ingram Chang	20/08/2013	Initial revision
V0.8	Ingram Chang	04/10/2013	Modification according to EOR Remove tables related to statement of compatibility
V0.9	Ingram Chang	03/11/2014	Finalization
V1.0	Ingram Chang	03/24/2014	Fix typo
V1.0.1	Ingram Chang	03/27/2014	Fix typo

Table of Content

REVISION HISTORY I
TABLE OF CONTENT II
LIST OF FIGURE IV
LIST OF TABLE IV
1 ST INTRODUCTION 1
1.1 ST Identification 1
1.2 TOE Identification 1
1.3 TOE Overview 1
1.3.1 TOE Type 1
1.3.2 Intended usage of the TOE 1
1.3.3 Security features of the TOE 2
1.3.4 TOE lifecycle 2
1.3.5 Non-TOE components 3
1.4 TOE Description 3
1.4.1 Physical Scope 3
1.4.2 Logical Scope 3
2 CONFORMANCE CLAIMS 5
2.1 COMMON CRITERIA (CC) CONFORMANCE 5
3 SECURITY PROBLEM DEFINITION 6
3.1 Assets 6
3.2 User Data 6
3.3 TSF Data 6
3.4 Threats 7
3.5 Assumptions 8
3.6 OSPs 8
4 SECURITY OBJECTIVES 10
4.1 Security Objectives for the TOE 10

4.2 Security Objectives for the TOE Environment 11
4.3 Security Objectives Rationale 11
5 SECURITY REQUIREMENTS 14
5.1 Security Functional Requirements 14
5.1.1 Authentication 14
5.1.2 Self Protection 14
5.1.3 Access and Flow Control 15
5.1.4 Audit 19
5.1.5 Cryptographic Support 20
5.1.6 Physical Security 21
5.2 Security Assurance Requirements 21
5.3 Security Requirement Rationale 22
5.4 Security Assurance Rationale 24
5.5 Dependency check 25
6 TOE SUMMARY SPECIFICATION 27
6.1 Chip Security Functions 27
6.2 Software Security Functions 27
7 GLOSSARY 29
8 REFERENCE 30

List of Figure

FIGURE 1 ECC SYSTEM ARCHITECTURE 1
FIGURE 2 LOGICAL OVERVIEW OF THE TOE 4

List of Table

TABLE 3 ACCESS CONTROL RULES 17
TABLE 4 SECURITY ASSURANCE REQUIREMENTS 22
TABLE 6 SFR DEPENDENCY CHECKING 26

1 ST INTRODUCTION

1.1 ST Identification

ECC2 CPU Card Security Target, version 1.0.1

1.2 TOE Identification

This Security Target describes the target of evaluation TNP ECC2 CPU Card, Version 1.0. The TOE consists of the following components:

- TOE Firmware: ECC2, version 1.0
- TOE Hardware: Infineon smartcard M7794 A12 (BSI-DSZ-CC-0883-2013)
- TOE Document: ECC2 CPU Card Security Guideline, Version 1.0

The product type of the TOE is smartcard.

1.3 TOE Overview

This chapter provides a brief overview of the TOE's functionality and security features.

1.3.1 TOE Type

The TOE is an anonymous Electronic Purse (EP) smartcard application running on the Infineon M7794 A12 (BSI-DSZ-CC-0883-2013) IC to the implementation of ECC EP functionalities. The Infineon M7794 A12 smartcard was certified conformant to [PP0035]. The firmware is installed on the NVM of the chip.

1.3.2 Intended usage of the TOE

The TOE is a dual interface CPU smartcard that implements EP (electronic purse) functionality for online or offline low value transaction in Easy Card Corporation (ECC) payment system. It is mainly used for the Taipei Metro System and in convenience stores. The architecture of the ECC payment system is provided below:

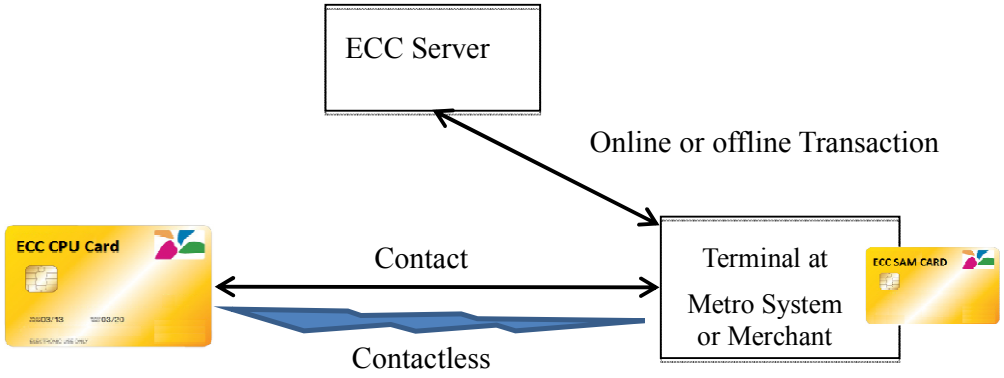

Figure 1 ECC System Architecture

The usage scenario of the EP is as below:

- Credit Purse transaction: the purse holder gives funds to the ECC Company (online transaction) or the merchant (offline transaction) who loads the EP with an equivalent amount of Electronic Money (EM),
- Debit Purse transaction: the purse holder asks the merchant for a service and transfers (offline) EM from his EP to the merchant.

The TOE is able to:

- Store its amount of EM which defines the balance of the EP,
- Indicate the available amount of EM,
- Debit EM via Debit Purse operations,
- Credit EM via Credit Purse transactions,
- Update parameters.

1.3.3 **Security features of the TOE**

The primary functionality of the TOE is to allow the purse holder to make EM payment in a simple, secure and fast way. The security features of the TOE are listed below:

- Integrity protection of EM during credit and debit operations,
- Integrity and confidentiality protection of cryptographic keys when used or stored
- Mutual authentication between TOE and Terminal for Credit Purse operation

  Note: Auto-load command is supported but no actual functionality.
- Mutual authentication between TOE and Terminal for Debit Purse operation
- Mutual authentication for parameters management
- Audit generation and review

1.3.4 **TOE lifecycle**

TOE life cycle

The TOE life cycle follows the life cycle described in [PP0035]. It is divided into seven distinct phases:

- Phase 1 “Embedded software development” that concerns the development of the IC
- dedicated software as well as the embedded software for EP purposes,
- Phase 2 “IC design”,
- Phase 3 “IC manufacturing”,
- Phase 4 “IC packaging”,
- Phase 5 “Composite product integration” that covers the composite product finishing process, preparation and shipping to the EP personalization line,
- Phase 6 “Personalisation”, where the EP administration data is loaded into the EP’s

memory, following the personalization instructions manual,

- Phase 7 “Usage stage” that corresponds to the operational phase of the TOE.

The TOE is delivered at the end of phases 5. All the phases before TOE delivery are evaluated according to the EAL4+ requirements. The TOE protects itself in Phase 7. In Phases 6, personalization activities are conducted according to ECC Personalization security requirement, [AGD]. Security requirements for the personalization are defined in [AGD] and are agreed upon by means of dedicated contracts with each entity performing them. The requirements are enforced by ECC Company.

1.3.5 **Non-TOE components**

The TOE requires the following IT in its environment:

- ECC SAM Card: while the terminal is performing transaction with ECC CPU Card, ECC SAM Card and ECC CPU Card will perform mutual authentication with the terminal’s assistance.
- Terminal: The terminal uses contact and/or contactless to interact with ECC CPU card to perform transaction or TOE management.
- ECC EM Server: ECC CPU card transactions are processed online or offline by the ECC EM Server.

1.4 TOE Description

1.4.1 **Physical Scope**

The physical scope of the TOE is:

- ECC EP firmware on Infineon M7794 A12 IC, version 1.0

  Note: The ECC firmware is loaded in the NVM of the chip in phase 4.
- Infineon M7794 A12 IC (BSI-DSZ-CC-0883-2013)
- [AGD] ECC2 CPU Card Security Guidance, version 1.0

The TOE’s on-chip memories are:

- User RAM size 6kbytes (of which the first 320 bytes are reserved for Infineon Technologies) Plus additional 1152 bytes of crypto RAM
- NVM with SOLID FLASH technology : 136kbytes (132kbytes for NVM、4kbytes for branch table)

1.4.2 **Logical Scope**

The following figure shows an overview about the logical structure of the TOE:

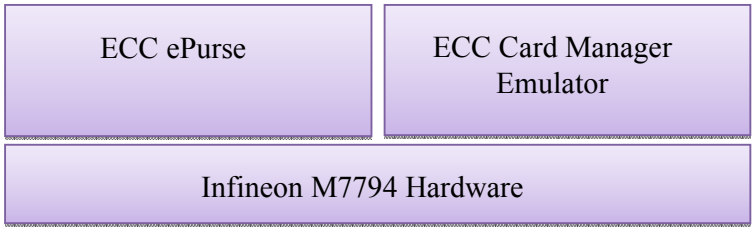

Figure 2 Logical overview of the TOE

The logical services of the TOE are:

- ECC ePurse application services:
  - Protection of EM and transactions
- ECC Card Manager Emulator services:
  - Management of card life-cycle
  - Management and control of the communication between TOE and the ECC Management server
  - Secure installation and deletion of the ECC ePurse application.

  Note: The Card Manager Emulator is not a GlobalPlatform Card Manager. It emulates part of the standard Card Manager functions for card manager
- M7794 Hardware services:
  - Protect the security assets
  - Provide secure cryptographic functionality (AES, RNG)
  - Checking environmental operating conditions

2 CONFORMANCE CLAIMS

2.1 COMMON CRITERIA (CC) CONFORMANCE

This ST conforms to:

- CC, version 3.1R4, as defined by [CCp1], [CCp2], [CCp3] and [CEM].
- CC Part 2 as CC Part 2 conformant
- CC Part 3 as CC Part 3 conformant

This ST conforms to no Protection Profile.

This ST conforms to EAL 4 augmented by ALC_DVS.2, AVA_VAN.5 and to no other packages.

This is a composite evaluation based on the hardware platform Infineon M7794 A12 chip certificate and evaluation results:

- Chip vendor: Infineon Technologies AG
- Scheme: BSI
- Certification number: BSI-DSZ-CC-0883-2013
- PP conformance: Security IC Platform Protection Profile, Version 1.0, 15 June 2007, BSI-CC-PP-0035-2007
- Common Criteria version: 3.1
- Assurance: Common Criteria Part 3 conformant EAL 5 augmented by ALC_DVS.2, ATE_DPT.2, and AVA_VAN.5

Note: the TOE is similar to that described in Moneo Electronic Purse Protection Profile (Moneo – Electronic Purse Protection Profile, Ref SFPMEI-CC-PP-EP, Version 1.5, BMS/SFPMEI, February 4th 2010). However the application environment is different. Therefore the ST author used this PP as a reference for SFRs but the ST does not fully comply to this PP.

3 Security Problem Definition

3.1 Assets

The security assets and the type of protection required are listed below:

3.2 User Data

Electronic Money (EM): the amount (balance of the EP, loaded by credit transactions) stored in the Electronic Purse (EP).

Protection: integrity

EP identification and validity data (IV_DATA): the Primary Account Number, which is a unique sequence of numbers assigned to the EP used for identification purposes, and validity data that allows to detect EP end-of-life.

Protection: integrity.

3.3 TSF Data

EP application (EP_APP): the application code embedded in the EP

Protection: integrity

Transaction Log (LOG): the last transactions stored in log files.

Protection: integrity

Keys (KEYS): the EP secret keys used for authentication purposes.

Protection: integrity and confidentiality.

CTC: the sequence counters count the successive transactions.

Protection: integrity.

State of the EP (EP_STAT): the state of the EP stores information about the EP internal states during its usage phase.

Protection: integrity.

Application note:

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).

3.4 Threats

The threat agent of the following threats is any attacker.

T.COUNTERFEITING_DEBIT

Counterfeiting of a debit operation in order to debit the EP with an EM greater or lesser than the actual transaction that leads to unauthorized EM creation or loss. (Asset: **EM**)

T.COUNTERFEITING_CREDIT

Counterfeiting of a credit transaction in order to credit the EP without any financial counterpart that leads to unauthorized EM creation or loss. (Asset: **EM**)

T.COUNTERFEITING_UPDATE

Counterfeiting of a parameters update transaction in order to change the keys values or the static counters values. (Asset: **KEY**, **CTC**).

T.DISCLOSURE_KEYS

Unauthorized access to the secret keys. (Asset: **KEYS**)

T.INTEG_CODE

Unauthorized modification of the TOE code: an attacker modifies the code in order to bypass the security policy of the EP. (Asset: **EP_APP**)

T.INTEG_LOG

Unauthorized modification of transaction log: an attacker modifies the value of the last transaction log stored in the EP in order to input a known key. (Asset: **LOG**)

T.INTEG_KEYS

Unauthorized modification of stored keys: an attacker modifies the value of the secret keys and associated attributes stored in the EP that leads to unauthorized EM modification. (Asset: **KEY**)

T.INTEG_EM

Unauthorized modification of stored EM: An attacker modifies the amount of EM stored in the EP in order to increase or decrease the amount. (Asset: **EM**)

T.INTEG_IV_DATA

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. (Asset: **IV_DATA**)

T.INTEG_COUNTER

Unauthorized modification of stored static counter: an attacker modifies the value of sequence counters in order to force the EP accepting counterfeited or replayed transactions. (Asset: **CTC**)

T.INTEG_EP_STAT

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. (Asset: **EP_STAT**)

T.REPLAY_DEBIT

Replay of a debit: an EP is debited several times via a previous complete sequence of debit operations; it leads to unauthorized EM loss. (Asset: **EM**)

T.REPLAY_CREDIT

Replay of a credit transaction: an EP is loaded several times via a previous complete sequence of credit operations; it leads to unauthorized EM creation. (Asset: **EM**)

T.REPLAY_UPDATE

Replay of a parameters update transaction: an EP is updated several times via a previous complete sequence of parameters update operations; it leads to fraudulent changes of parameters stored in the EP (keys and static counters). (Asset: **KEYS, CTC**).

3.5 Assumptions

A.PROTECTION_AFTER_TOE_DELIVERY

It is assumed that the persons manipulating the TOE in the operational environment follow the TOE guides. And he is 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.

3.6 OSPs

OSP.MANAGE_SECRET

Management of secret data (e.g. generation, storage, distribution, destruction, loading into the TOE of cryptographic private keys, symmetric keys, user authentication data) performed outside the product on behalf of the TOE Manufacturer shall comply with security organizational policies that enforce integrity and confidentiality of these data. 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.

OSP.DEBIT_BEFORE_CREDIT

Debit from the EP always precedes credit of the SAM during a payment transaction.

OSP.SECURE_USAGE

The Purse holder shall keep the EP as a real purse with coins and bank notes and she/he shall not loan it, especially to untrusted persons.

4 Security Objectives

4.1 Security Objectives for the TOE

O.AUTH

The EP shall enforce mutual authentication with external devices prior any transaction.

O.EM

The EP shall prevent unauthorized creation or loss of EM.

O.CONF_DATA

The EP shall prevent unauthorized disclosure of confidential the Keys.

O.INTEG_DATA

The EP shall prevent unauthorized modification of user and TSF data.

O.LIMIT

The EP behavior shall be limited by maximum values defined in the static counters.

O.OPERATE

The EP shall ensure the continued correct operation of its security functions especially in case of abnormal transactions and unexpected interruption.

O.RECORD

The EP shall record the last transactions to support effective security management.

O.REPLAY

The EP shall detect and reject replayed transactions.

O.TAMPER

The EP shall prevent physical tampering of its security critical parts.

4.2 Security Objectives for the TOE Environment

OE.DEBIT_BEFORE_CREDIT

Debit must always precede credit during EM payment transaction.

OE.MANAGEMENT_OF_SECRETS

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

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 product guides (user and administrator guidance of the product, installation documentation and personalization guide).

OE.TOE_USAGE

The EM issuer shall communicate to the Purse holder the rules dealing with the use of the EP. Especially it must inform the user that he must keep EP the same way he does for a real purse,

The Purse holder shall enforce these rules.

OE.TIME

The terminal shall provide reliable time stamp to the TOE

4.3 Security Objectives Rationale

T.COUNTERFEITING_DEBIT is countered by:

- O.CONF_DATA and O.INTEG_DATA that prevent the unauthorized disclosure or modification of data,
- O.AUTH that requires the authentication of both the TOE and the external device before performing any transaction,
- O.EM which ensures EM flow preservation so that fraudulent creation or loss of EM in the EP using a debit operation 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. The time stamp for the log is provided by OE.TIME

T.COUNTERFEITING_CREDIT is countered by:

- O.CONF_DATA and O.INTEG_DATA that prevent the unauthorized disclosure or modification of data,
- O.AUTH that requires the authentication of both the TOE and the external device before performing any transaction,

- O.EM which ensures EM flow preservation so that fraudulent creation or loss of EM in the EP using a debit operation 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. The time stamp for the log is provided by OE.TIME
- OE.TOE_USAGE which ensures the TOE issuer provides to the user the rules to securely use its TOE

T.COUNTERFEITING_UPDATE is countered by:

- O.CONF_DATA and O.INTEG_DATA that prevent the unauthorized disclosure or modification of data,
- O.AUTH that requires the authentication of both the TOE and the external device before performing any transaction.

T.DISCLOSURE_KEYS is countered by:

- O.CONF_DATA which prevents from illegal disclosure of the security assets of the TOE,
- O.TAMPER which ensures security information and especially security assets of the TOE cannot be physically tampered in order to allow the disclosure of the assets.
- O.LIMIT which prevents the security assets from disclosure by limiting the usage time of the assets.

T.INTEG_CODE/T.INTEG_IV_DATA/T.INTEG_COUNTER/T.INTEG_SM are countered by:

- O.INTEG_DATA which prevents from illegal modification of the security assets of the TOE,
- O.OPERATE which ensures the correct operation of the related transaction,
- O.TAMPER which ensures security information and especially security assets of the TOE cannot be physically tampered.

T.INTEG_LOG is countered by:

- O.INTEG_DATA which prevents from illegal modification of the security assets of the TOE,
- O.RECORD which ensures that the TOE records necessary events and data in order to be presented again as an element of evidence of the real transaction,
- O.OPERATE which ensures the correct operation of the related transaction,
- O.TAMPER which ensures security information and especially security assets of the TOE cannot be physically tampered.

T.INTEG_KEYS is countered by:

- O.INTEG_DATA which prevents from illegal modification of the security assets of the TOE,
- O.OPERATE which ensures the correct operation of the related transaction,
- O.TAMPER which ensures security information and especially security assets of the TOE cannot be physically tampered.

T.INTEG_EM is countered by:

- O.INTEG_DATA which prevents from illegal modification of the security assets of the TOE,
- O.EM which ensures EM flow preservation so that fraudulent creation of EM in the EP is not possible,
- O.OPERATE which ensures the correct operation of the related transaction,
- O.TAMPER which ensures security information and especially security assets of the TOE cannot be physically tampered.

T.REPLAY_DEBIT/CREDIT are countered by:

- O.EM which ensures EM flow preservation so that fraudulent loss of EM in the EP using a debit operation is not possible,
- O.REPLAY which ensures the EP will operate in a continuous secure state in case of replayed debit operation; the replayed transaction will be detected and rejected by the EP.

T.REPLAY_UPDATE is countered by O.REPLAY which ensures the EP will operate in a continuous secure state in case of replayed parameters update transaction; the replayed transaction will be detected and rejected by the EP.

A.PROTECTION_AFTER_TOE_DELIVERY is directly covered by OE.PROTECTION_AFTER_TOE_DELIVERY

OSP.MANAGE_SECRETS is directly covered by OE.MANAGEMENT_OF_SECRETS

OSP.DEBIT_BEFORE_CREDIT is directly covered by OE.DEBIT_BEFORE_CREDIT.

OSP.SECURE_USAGE is covered by OE.TOE_USAGE

5 Security Requirements

The following notational conventions are used in the requirements. Assignment operations are indicated in **bold**. Selection operations are indicated in ***italic bold***. Refinements are indicated in **underlined bold**. In general refinements were applied to clarify requirements and/or make them more readable. Iterations were indicating by adding three letters to the component name.

5.1 Security Functional Requirements

5.1.1 Authentication

FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow

- **INITIAL PROCESSING command for EP authentication**
- **READ PURSE command**

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

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

FIA_UAU.3 Unforgeable authentication

FIA_UAU.3.1 The TSF shall ***prevent*** use of authentication data that has been forged by any **Terminal** of the TSF.

FIA_UAU.3.2 The TSF shall ***prevent*** use of authentication data that has been copied from any other **Terminal** of the TSF.

FIA_UAU.4 Single-use authentication mechanisms

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to:

- **the Terminal authentication mechanism with SAM,**
- **the Terminal authentication mechanism with remote server.**

FIA_UAU.6 Re-authenticating

FIA_UAU.6.1The TSF shall re-authenticate the **Terminal** under the conditions

- **beginning of a Credit transaction.**
- **beginning of a Debit transaction.**
- **beginning of a update parameter transaction.**

5.1.2 Self Protection

FDP_SDI.1 Stored data integrity monitoring

FDP_SDI.1.1The TSF shall monitor user data stored in containers controlled by the TSF for **any**

integrity errors on all objects, based on the following attributes: **checksum and duplicated user data**.

FPT_ITC.1 Inter-TSF confidentiality during transmission

FPT_ITC.1.1 The TSF shall protect all TSF data transmitted from the TSF to another **Terminal** from unauthorized disclosure during transmission.

Application note:

The "transmission" occurs during the parameters update transaction and the "TSF data" stands for EP parameters being updated.

FPT_ITI.1 Inter-TSF detection of modification

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and another **Terminal** within the following metric:

- **TOKEN value associated to the Payment commands (DEBIT PURSE, CREDIT PURSE)**
- **MAC value associated to Purse Management commands (PUT DATA, WRITE LOCK)**
- **TOKEN or/and MAC value associated to File Management commands (READ/UPDATE/APPEND RECORD)**
- **TOKEN value associated to CHANGE KEY command**

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and another **Terminal** and perform **command termination and return error status word** if modifications are detected.

FPT_RPL.1 Replay detection

FPT_RPL.1.1 The TSF shall detect replay for the following entities:

- **Debit transactions,**
- **Credit transactions,**
- **Parameters update transactions.**

FPT_RPL.1.2 The TSF shall perform **the abort of the transaction in process** when replay is detected.

FPT_RCV.4 Function recovery

FPT_RCV.4.1 The TSF shall ensure that **debit, credit and parameters update transactions** have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

5.1.3 Access and Flow Control

FDP_ACC.2 Complete access control

FDP_ACC.2.1 The TSF shall enforce the **Assets Security policy** on

subject: Command Interpreter,

objects: EM, KEYS, CTC, LOG, EP_STAT, Purse Attribute, Files

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.

FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the **Assets Security policy** to objects based on the following:

subject: Command Interpreter

objects: EM, KEYS, CTC, LOG, Purse Attribute, Files

security attribute: EP_STAT

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

Subject operation of Command Interpreter	**Object**	**Security Attribute State of EP_STAT**	**Access Control**
READ RECORD **UPDATE RECORD** **APPEND RECORD**	**Files**	**Started、Active、Debit、Credit、Auto-load**	**Operations are allowed when correct authentication token or/and MAC is provided.**
DEBIT PURSE	**EM, LOG**	**Active**	**Operation is allowed when correct authentication token is provided.**
CREDIT PURSE	**EM, LOG**	**Active**	**Operation is allowed when correct authentication token is provided.**
INITIATE PROCESSING	**CTC**	**Started、Active、Debit、Credit**	**The number of this operation is limited by the specified maximum CTC value.**
CHANGE APP ADMIN KEY	**APP ADMIN KEY**	**Active**	**Operation is allowed when correct authentication token is provided.**
WRITE LOCK	**Purse Attributes**	**Active**	**Operation is allowed when correct MAC is provided.**
READ RECORD	**LOG**	**Started**	**Free to read**

READ PURSE	**EM, CTC**	**Ready**	**Free to read**
PUT DATA	**Purse Attributes**	**Active**	**Operation is allowed when correct MAC is provided.**
GET DATA	**File ID**	**Ready、Started、Active、Debit、Credit**	**Free to read (read only)**
All	**KEYS**	**All**	**Read Forbidden**

Table 1 Access Control Rules

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

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **None**.

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **Assets Security policy** to restrict the ability to ***modify*** the security attributes: **APP ADMIN KEY** to **Valid Token Holder**.

FMT_MSA.3/Asset Static attribute initialization

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

FMT_MSA.3.2/Asset The TSF shall allow the **Valid Token Holder** to specify alternative initial values to override the default values when **APP ADMIN KEY** is created.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: **modifying the APP ADMIN KEY**

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1The TSF shall enforce the **Transaction policy** on

- **subject: EP, SAM**
- **information: EM, Authentication Token, MAC**
- **operation: READ RECORD, UPDATE RECORD, APPEND RECORD, DEBIT PURSE, CREDIT PURSE, INITIATE PROCESSING**

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the **Transaction policy** based on the following types of subject and information security attributes:

subjects: EP, SAM

information : EM transaction, Management command, TOKEN, MAC

security attributes: EP_STAT

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:

- **EP permits an EM transaction when its authentication TOKEN is correct.**
- **EP permits an EM transaction when its transaction amount does not exceed the maximum transaction amount.**
- **EP permits an EM transaction when the balance of EP does not exceed maximum balance and not below minimum balance.**
- **EP permits a Purse Management command when its MAC is correct.**
- **EP permits a File Management command when its authentication TOKEN and/or MAC is correct.**
- **EP permits a transaction or management operation when the command sequence tracked by EP_STAT is correct.**

FDP_IFF.1.3 The TSF shall enforce the **None.**

FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: **None**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **When the following security attribute are incorrect the information are rejected by the subject.**

- **Authentication TOKEN or MAC**
- **Purse attributes (maximum balance, maximum transaction amount)**
- **State machine state**

FMT_MSA.3/Transaction Static attribute initialization

FMT_MSA.3.1/Transaction The TSF shall enforce the **Transaction policy** to provide ***restrictive*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Transaction The TSF shall allow the **None** to specify alternative initial values to override the default values when an object or information is created.

FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1The TSF shall enforce the **Transaction policy** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled

under the SFP from outside the TOE: **None**.

Application note:

"User data" stands for user and TSF data entering the EP during credit, debit or EP management transactions.

5.1.4 **Audit**

FAU_GEN.1 Audit data generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

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:

- **last credit transaction**
- **last debit transaction,**

Refinement:

The audit functions are active all the time, hence item a) Start-up and shutdown of the TOE is not relevant.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

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:**

- **Transaction sequence number**
- **Transaction date**
- **Transaction amount**
- **Electronic value**
- **Device ID**
- **Purse balance**

Refinement:

Date and time of events are determined by the terminal.

Application note:

For each type auditable event, the Security Target author shall define the maximum number of information stored. Date and time of the event are determined by the terminal.

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide **all users** with the capability to read

- **Transaction sequence number**
- **Transaction date**

- **Transaction amount**
- **Electronic value**
- **Device ID**
- **Purse balance**

from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

FAU_STG.1 Protected audit trail storage

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.

5.1.5 Cryptographic Support

FCS_COP.1/SessionKey Cryptographic operation

FCS_COP.1.1/SessionKey The TSF shall perform **session key generation** in accordance with a specified cryptographic algorithm **AES** and cryptographic key sizes **128 bits** that meet the following: **AES standard**.

FCS_COP.1/TOKEN Cryptographic operation

FCS_COP.1.1/TOKEN The TSF shall perform **TOKEN calculation** in accordance with a specified cryptographic algorithm **AES** and cryptographic key sizes **128 bits** that meet the following: **AES standard**.

FCS_COP.1/MAC Cryptographic operation

FCS_COP.1.1/MAC The TSF shall perform **MAC calculation** in accordance with a specified cryptographic algorithm **AES** and cryptographic key sizes **128 bits** that meet the following: **AES standard**.

FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 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 standard**

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The TSF shall destroy all cryptographic keys in accordance with a specified cryptographic key destruction method **overwrite the keys with arbitrary value** that meets the following: **none**.

FIA_SOS.2 TSF Generation of secrets

FIA_SOS.2.1The TSF shall provide a mechanism to generate **random values** that meet **AIS31 class P2**.

FIA_SOS.2.2The TSF shall be able to enforce the use of TSF generated secrets for **8-bytes challenge generation (INITIAL PROCESSING).**

5.1.6 **Physical Security**

FPT_PHP.2 Notification of physical attack

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 For **the IC**, the TSF shall monitor the devices and elements and notify **the EP** when physical tampering with the TSF's devices or TSF's elements has occurred.

Application note:

The TSF shall rely on its Integrated Circuit certified against [PP0035] to detect physical attacks.

FPT_PHP.3 Resistance to physical attack

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

Application note:

The TSF shall rely on its Integrated Circuit certified against [PP0035] to resist to physical tampering scenarios.

5.2 Security Assurance Requirements

The security assurance requirement level is EAL4 augmented with ALC_DVS.2 and AVA_VAN.5.

Security assurance requirements	Titles
Class ADV: Development	
ADV_ARC.1	Security Architecture
ADV_FSP.4	Functional specification
ADV_IMP.1	Implementation representation
ADV_TDS.3	TOE design
Class AGD: Guidance documents	
AGD_OPE.1	Operational user guidance

AGD_PRE.1	Preparative guidance
Class ALC: Life-cycle support	
ALC_CMC.4	CM capabilities
ALC_CMS.4	CM scope
ALC_DEL.1	Delivery
ALC_DVS.2	**Development security**
ALC_LCD.1	Life-cycle definition
ALC_TAT.1	Tools and techniques
Class ASE: Security Target evaluation	
ASE_CCL.1	Conformance claims
ASE_ECD.1	Extended components definition
ASE_INT.1	ST introduction
ASE_OBJ.2	Security objectives
ASE_REQ.2	Derived security requirements
ASE_SPD.1	Security problem definition
ASE_TSS.1	TOE summary specification
Class ATE: Tests	
ATE_COV.2	Coverage
ATE_DPT.1	Depth
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing
Class AVA: Vulnerability analysis	
AVA_VAN.5	**Vulnerability analysis**

Table 2 Security Assurance Requirements

5.3 Security Requirement Rationale

O.AUTH is covered by:

- FIA_UAU.1, which requires the TOE authenticate the external device for a transaction,

- FIA_UAU.3, which prevents against use of forged authentication data,
- FIA_UAU.4 which prevents against reuse of authentication data,
- FIA_UAU.6, which requires the TOE re-authenticate the external device for each transaction,
- FIA_SOS.2 which requires the TOE to use random challenge numbers for authentication,
- FCS_COP.1/MAC/Token, FDP_ITC.1 and FCS_CKM.4 which require the cryptographic operations, key loading and the key destruction mechanism for authentication.

O.EM is directly covered by:

- FDP_IFC.1, FDP_IFF.1, FDP_ITC.1 and FMT_MSA.3/Transaction which enforce an information flow control policy on user data and TSF data,
- FPT_RCV.4 which ensures that a transaction is performed completely or is aborted and that a secure state is preserved,
- FPT_RPL.1 which ensures that credit, debit and parameters update transactions are protected against replay; the TSF can detect it and react by aborting the transaction in process,
- FCS_COP.1/SessionKey, FCS_CKM.1 and FCS_CKM.4 that specify the characteristics of cryptographic operations and key destruction mechanism the Transaction control policy

O.CONF_DATA is covered by:

- FPT_ITC.1 which ensures the confidentiality of the TSF data during transmission for the case of the parameters update transactions,
- FDP_ACC.2, FDP_ACF.1, FMT_MSA.1 and FMT_MSA.3/Asset which ensure that security assets are protected by a secure access control,
- FPT_PHP.2 and FPT_PHP.3 which ensure physical protection of confidential data,
- FPT_RCV.4 which ensures that the TOE data cannot be disclosed and that it is impossible to put the TOE in an inconsistent and unstable state allowing to retrieve the security assets.
- FCS_COP.1/SessionKey, FCS_CKM.1 and FCS_CKM.4 that specify the characteristics of cryptographic operations, key generation and key destruction mechanism for data encryption

O.INTEG_DATA is covered by:

- FDP_SDI.1 which ensures that user data stored in the TOE are monitored against any integrity error,
- FPT_ITI.1 which ensures the integrity of the TSF data during transmission for the case of the parameters update transactions,
- FDP_ACC.2, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3/Asset and FMT_SMF.1 which ensure that security assets cannot be modified without passing by a secure access control,
- FPT_PHP.2 and FPT_PHP.3 that address physical protection of integer data,
- FPT_RCV.4 which ensures that the TOE data cannot be modified and put in an inconsistent and unstable state which could alter the integrity of the TOE security assets,

- FAU_STG.1 which protects the audit record stored in the TOE against unauthorized deletion and detects any attack against this security asset.

O.LIMIT is covered by:

- FDP_IFF.1, FDP_IFC.1, FDP_ITC.1, FMT_MSA.1 and FMT_MSA.3/Asset which define the access control policy within the TOE for the protection of the security assets of the TOE, in particular the rules to apply for the case of any transaction. This includes the EM limited by the value of a maximum amount when the TOE processes a credit transaction.

O.OPERATE is covered by:

- FPT_RCV.4 which ensures that the TOE cannot enter in an unstable and inconsistent state, even due to a failure during a transaction.

O.RECORD is covered by:

- FAU_GEN.1 which requires the generation of an audit record of the last performed transactions,
- FAU_SAR.1 which allows the capability to read this audit record.
- FAU_STG.1 that protects the audit record stored in the TOE against unauthorized detection and detects any attack against the asset

O.REPLAY is covered by:

- FPT_RPL.1 which ensures that credit, debit and parameters update transactions are protected against replay; the TSF can detect it and react by aborting the transaction in process,
- FIA_SOS.2 which ensures the TOE can generate random value to enforce the protection against replay attacks.
- FCS_CKM.1, FCS_COP.1/SessionKey and FCS_CKM.4 that specify the characteristics of cryptographic operations and the key destruction mechanism

O.TAMPER 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
- FPT_PHP.2 and FPT_PHP.3 which requires detection and protection against physical tampering

5.4 Security Assurance Rationale

All SARs are drawn from the [CCp3]. Augmentations are in line with the requirement high resistance against state-of-the-art attacks of an e-purse smartcard. Augmentation results from the selection of ALC_DVS.2 and AVA_VAN.5.

5.5 Dependency check

SFR	Dependency
FIA_UAU.1	FIA_UID.1: FIA_UID.1 is not relevant to the TOE: there is no identification to the TOE
FIA_UAU.3	-
FIA_UAU.4	-
FIA_UAU.6	-
FDP_SDI.1	-
FPT_ITC.1	-
FPT_ITI.1	-
FPT_RPL.1	-
FPT_RCV.4	-
FDP_ACC.2	FDP_ACF.1: met
FDP_ACF.1	FDP_ACC.1: met FMT_MSA.3: Met by FMT_MSA.3/Asset
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1]: met by FDP_ACC.1 FMT_SMR.1: FMT_SMR.1 is not relevant to the TOE: there is no role identification in the TOE. FMT_SMF.1: met
FMT_MSA.3/Asset	FMT_MSA.1: met by FMT_MSA.1 FMT_SMR.1: FMT_SMR.1 is not relevant to the TOE: there is no role identification in the TOE.
FMT_SMF.1	-
FDP_IFC.1	FDP_IFF.1: met
FDP_IFF.1	FDP_IFC.1: met FMT_MSA.3: met by FMT_MSA.3/Transaction
FMT_MSA.3/Transaction	FMT_MSA.1: not necessary to be met since the security attributes are fixed and cannot be changed
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1]: Met by FDP_IFC.1 FMT_MSA.3: met by FMT_MSA.3/Transaction

FAU_GEN.1	FPT_STM.1: The time stamp is obtained from OE.TIME
FAU_SAR.1	FAU_GEN.1: met
FAU_STG.1	FAU_GEN.1: met
FCS_COP.1/SessionKey	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]: met by FCS_CKM.1 FCS_CKM.4: met
FCS_COP.1/TOKEN	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]: met by FDP_ITC.1 FCS_CKM.4: met
FCS_COP.1/MAC	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]: met by FDP_ITC.1 FCS_CKM.4: met
FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1]: met by FCS_COP.1/SessionKey FCS_CKM.4: met
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]: met by FDP_ITC.1/key and FCS_CKM.1
FIA_SOS.2	-
FPT_PHP.2	FMT_MOF.1: not met because during operational use of the TOE, the behavior of security functions could not be changed.
FPT_PHP.3	-

Table 3 SFR dependency checking

6 TOE Summary Specification

6.1 Chip Security Functions

SF_DPM: Device Phase Management

The life cycle of the chip is split up into several phases. Different operation modes help to protect the TOE during each phase of its lifecycle.

SF_PS: Protection against Snooping

The chip uses various means to protect from snooping of memories and busses and prevents single stepping.

SF_PMA: Protection against Modifying Attacks

This chip implements protection against modifying attacks of memories, alarm lines and sensors.

SF_PLA: Protection against Logical Attacks

Memory access of the chip is controlled by a Memory Management Unit (MMU), which implements different privilege levels. The MMU decides, whether access to a physical memory location is allowed based on the access rights of the privilege levels.

SF_CS: Cryptographic Support

The chip is equipped with an asymmetric and a symmetric hardware accelerator and also software modules to support several symmetric and asymmetric cryptographic operations. It further provides random numbers to meet FCS_RNG.1.

SF_PHY: Protection against Physical Manipulation

The function protects the chip against manipulation of the IC hardware, the IC Dedicated Software in ROM, the Security IC Embedded Software in ROM and EEPROM, the application data in EEPROM and RAM including TSF data in the security rows. It also protects User Data or TSF data against disclosure by physical probing when stored or while being processed by the TOE.

6.2 Software Security Functions

SF_PHY_SW: Software Managed Physical Protection

- The EP receives a hardware interrupt and halts the chip when a physical attack is detected by the chip (FPT_PHP.2).
- Physical protection (FPT_PHP.3)

SF_CRYPTO: Cryptographic Operation

- Manage the creation, loading and deletion of cryptographic keys (FCS_CKM.1, FDP_ITC.1, FCS_CKM.4)
- Manage the cryptographic operations in AES symmetric mode (FCS_COP.1/SessionKey, FCS_COP.1/MAC, FCS_COP.1/Token)
- Manage the generation of challenges (FIA_SOS.2)

SF_ACCESS_CONTROL: Access Control

- Manage the EP Access Control (FDP_ACC.2, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3/Asset, FMT_SMF.1)
- Manage the Payment transactions (FDP_IFC.1, FDP_IFF.1, FMT_MSA.3/Transaction, FDP_ITC.1)
- Replay protection (FPT_ITC.1, FPT_ITI.1, FPT_RPL.1)

SF_LOG: Audit

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

SF_SELF_PROTECTION: Self Protection

- Ensure the integrity of assets (FDP_ACC.2, FDP_ACF.1, FDP_ITC.1, FDP_SDI.1, FPT_ITI.1)
- Ensure the confidentiality of assets (FDP_ACC.2, FDP_ACF.1, FDP_ITC.1, FPT_ITC.1)
- Ensure a secure state according to the execution flows (FPT_RCV.4)
- Perform secure authentication (FIA_UAU.1, FIA_UAU.3, FIA_UAU.4, FIA_UAU.6, FDT_ITC.1, FPT_ITI.1)

7 Glossary

ADMIN	Administrator
AES	Advanced Enctyption Standard
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
APP	Application
ATR	Answer To Reset
CTC	Card Transaction Counter
EAL	Evaluation Assurance Level
ECC	Easy Card Corporation
ECC2	2nd Generation ECC card / the TOE
EEPROM	Electrically Erasable Programmable Read-Only Memory
EM	Electronic Money
EP	Electronic Purse
HW	Hardware
ICC	Integrated Circuit Card
ID	Identification
ISO	International Organization for Standardization
NVM	Non-Volatile-Memory
OS	Operating System
PP	Protection Profile
PPS	Protocol and Parameter Selection
PSN	Purse Serial Number
RNG	Random Number Generator
ROM	Read-Only Memory
SAM	Secure Access Module
ST	Security Target
SW	Software
TOE	Target Of Evaluation

8 Reference

[AGD] ECC2 CPU Card Security Guidance, Version 1.0

[CCp1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2012, Version 3.1 Revision 4, CCMB- 2012-09-001

[CCp2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated September 2012, Version 3.1 Revision 4, CCMB- 2012-09-002

[CCp3] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated September 2012, Version 3.1 Revision 4, CCMB- 2012-09-003

[CCDB] Composite Product Evaluation for Smartcards and similar devices, dated September 2007, Version1.0, Revision 1, CCDB-2007-09-001

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

[ECC CPU] Functional Specification – CPU Card

[ECC SAM] Functional Specification – SAM Card

[GP] GlobalPlatform - Card Spec v2.1.1

[PP0035] Security IC Platform Protection Profile