SAM for EM system Protection Profile

Emission Date	:	4 February 2010
Ref./Version	:	SFPMEI-CC-PP-SAM/1.5
Number of pages	:	54 (including 6 header pages)

Table of contents

List of figures

List of tables

1 Protection Profile Introduction

1.1 Protection Profile Identification

Title:	Protection Profile – Secure Access Module for Electronic Money System
Version number:	1.5
Date:	4 February 2010
Sponsors:	BMS - SFPMEI
Technical editor:	Trusted Labs S.A.S, 5 rue du Bailliage 78000 Versailles
CC version:	3.1 revision 3

1.2 Context

The Secure Access Module Protection Profile elaborated by BMS (Billettique Monétique Services) and SFPMEI (Société Financière du Porte-Monnaie Électronique Interbancaire) is the Secure Access Module counterpart of the Intersector Electronic Purse and Purchase Device Protection Profile [PP/0101], updated to comply with Common Criteria v3.1 revision 3 and to cover the most recent state of the art in terms of security of an Electronic Money (EM) System. The Electronic Purse (EP) counterpart of [PP/0101] is addressed by the Electronic Purse Protection Profile [PP EP]. These new Protection Profiles supersede the Protection Profile [PP/0101].

1.3 Moneo System

The Moneo system is a payment system (Electronic Money system or EM system) intended for low value offline payment transactions. The global functioning of the EM system is based on three cycles:

- a first one consisting in EM creation,
- a second one consisting in payment of goods or services,
- a third one consisting in EM extinguishment.

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 EM amount. The EP that receives amounts in several transactions may aggregate them into a single EM amount. Conversely, the EM amount stored in an EP may be broken up and dispensed in several transactions. The aggregation may also be performed by the SAM.

These cycles can be summarized as follows:

- the purse holder gives funds to the EM issuer who loads the EP with an equivalent amount of EM (**load and quickload transactions**)
- the purse holder asks the merchant for a service and transfers EM from his EP to the SAM (**EM Payment transaction, corresponding to a debit operation for the EP and a credit operation for the SAM**).
- the merchant asks the EM issuer for credit on its bank account in exchange for EM (**collect transaction**)

During each transaction (load, quickload, EM payment, collect), EM circulates within a closed loop system. Only the EM issuer is authorised to create or extinguish EM. Furthermore, the EM credited on one side (EP or SAM) should be always equal to the EM debited on the other side (SAM or EP, respectively).

Figure 1 gives an overview of the Moneo system:

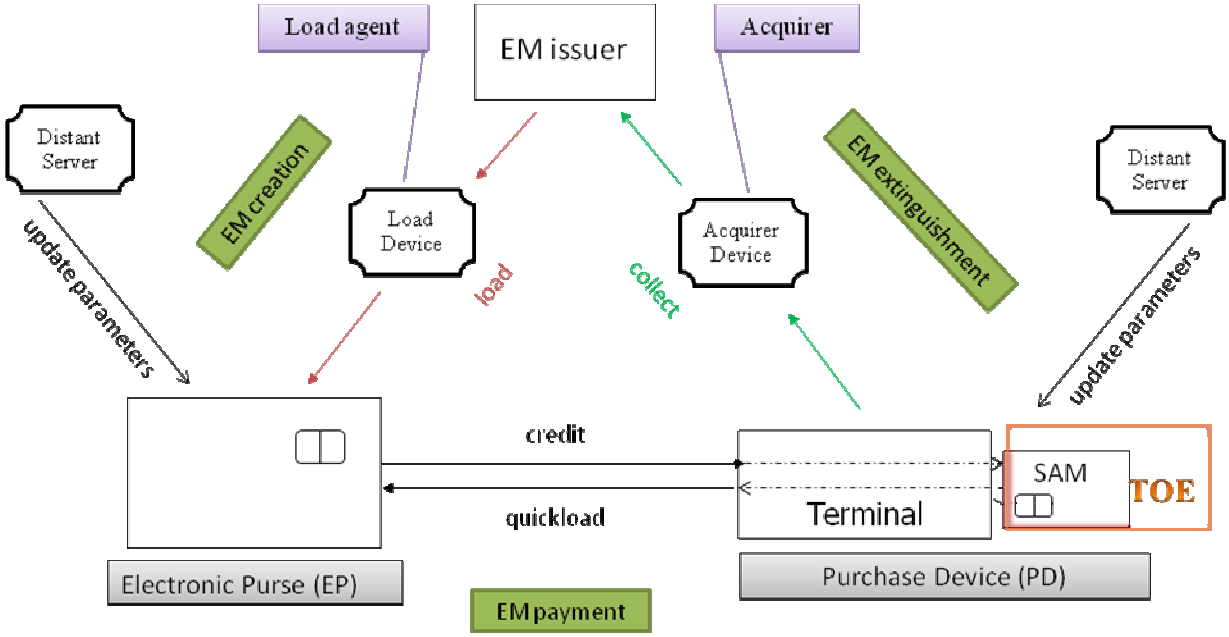

Figure 1: Moneo System

The following sections describe the devices, the actors and the operations involved in the EM system.

1.3.1 Devices

This section describes the devices that play a role in an EM system.

Electronic Purse

An Electronic Purse (EP) is a smartcard or similar device comprising a Security IC and embedded software and data. Its functionalities are similar to traditional purse functionalities with the distinction that it uses Electronic Money (EM) instead of cash money. An EP is used to facilitate payments of low value. The fully operational EP contains various updatable parameters.

Purchase device

A purchase device (PD) 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), built on a Security IC. The SAM shall provide the necessary security for the EM payment, the quickload and the collect transactions. It contains various updatable parameters.

Load device

A load device is a server. Its main functionality is to load the EM in the EP in exchange of funds via the EM issuer.

Acquirer device

An acquirer device is a server intended to handle the collect transactions.

Distant Server

A distant server has the responsibility to update the internal parameters of the EP and the SAM.

1.3.2 Actors

EM issuer

The EM issuer guarantees the EM in an EM system. To this end, the EM issuer:

- creates and dispenses EM in exchange for funds received,
- redeems collected EM and extinguishes it.

Purse holder

The purse holder is the person that is in possession of the EP and uses it for EM payment transactions. Purse holders need to protect their EP in the same way as cash.

Merchant

A merchant sells goods or services for which he accepts payment by EP. In order to handle the EM payment transactions, the merchant operates one or more purchase devices in which a SAM stores EM until collect. The merchant is responsible for the operational security of the purchase device he controls.

Load agent

A load agent (LA) is a trusted agent of an EM issuer. He executes the load transactions with the EP on behalf of the EM issuer and operates a load device for this purpose. A load agent is responsible for the operational security of its part of the EM system, and must protect the load devices he controls against unauthorised use. He is also responsible for transferring payment received from the Purse holder to the EM issuer for settlement.

Acquirer

An acquirer is a trusted agent of the EM issuer who is responsible for collecting EM from a SAM concerning EM payment transactions. In order to handle the collect transactions the acquirer operates one or more acquirer devices. The acquirer is responsible for the operational security of its part of the EM system, and must protect the acquirer devices he controls against unauthorised use. He is also responsible for transferring payment received from the EM issuer to the merchant for settlement.

1.3.3 Transactions

The EM system IT functionalities consist of the following operations:

Load

The EP is credited with an amount of EM created by the EM issuer, via a load agent; the purse holder gives a corresponding amount of funds in turn.

Quickload

This operation is used to load EM into the EP. It is processed offline by the SAM. It may be performed when the purchase amount is greater than the balance of the purse in order to complete the EM payment.

EM payment (debit/credit operation)

The EP is debited of an EM amount while the SAM is credited with the same amount. The purse holder receives goods or services in return.

Collect

One or several amounts of the EM corresponding to a set of payment transactions stored by a SAM are delivered to the EM issuer via the acquirer device.

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 and the transaction keys.

1.4 Overview of the TOE

This section describes the Target Of Evaluation (TOE). It addresses the product type and the intended usage of the TOE.

1.4.1 TOE Type

The Target Of Evaluation (TOE) is a Secure Access Module (SAM), comprising the IC with contact interface and all the embedded software necessary to the implementation of the SAM functionalities, with the IC certified conformant to [PP-BSI-0035-2007].

This protection profile does not mandate any particular software technology; both native and virtual-machine-based implementations of SAM are acceptable, provided they comply with one of the following protection profiles:

- Embedded Software for Smart Secure Devices Protection Profile, Basic and Extended Configurations [ANSSI-CC-PP-2009/02],
- Java Card System Protection Profiles [PP-JCS-Collection]. Note that Java Card System - Open Configuration Protection Profile [PP-JCS-Open] and Java Card System - Closed Configuration Protection Profile [PP-JCS-Closed], both conformant to CC v3.1 and currently under evaluation, should be preferred to the CC v2.1 profiles defined in [PP-JCS-Collection] once their certificates will be available and whenever they match the characteristics of the TOE.

Conformity to [ANSSI-CC-PP-2009/02] can be claimed for any kind of implementation, native or interpreted (e.g. Java Card). Conformity to [PP-JCS-Collection] ([PP-JCS-Closed] or [PP-JCS-Open]) can be claimed on Java Card Secure Access Modules only.

Note that conformity can be reached through a separate evaluation (so that the SAM or its underlying platform holds a valid certificate for one of these PPs) or the SAM can be evaluated at the same time against this protection profile and against [ANSSI-CC-PP-2009/02] or [PP-JCS-Collection] ([PP-JCS-Closed] or [PP-JCS-Open]). The developer and SFPMEI/BMS shall agree on the PP configuration to be used.

Note that the IC may support other applications.

Although a Moneo SAM can be configured in various ways, all the configurations offer similar security functionalities. Hence, the term SAM denotes any of them.

1.4.2 TOE Usage

The TOE is able to:

- store its amount of EM,
- receives an amount of EM from an EP via a credit operation after debit of the EP,
- delivers stored EM to an acquirer device via a collect transaction,
- executes quickload transactions,
- update parameters.

1.4.3 Security features of the TOE

The primary functionality of the TOE is to secure collect, quickload and payment transactions. It computes the cryptograms related to those operations and maintains counters and balances related to them.

The main characteristic of the TOE is that it provides offline EM payment transactions capabilities thus requiring security mechanisms to prevent from fraud. This means:

- integrity protection of EM during collect, quickload and payment transactions,
- integrity and confidentiality protection of cryptographic keys when used or stored,
- mutual authentication between the TOE and the EP during quickload and payment transactions,
- signature of collect transactions.

1.4.4 TOE life cycle

The TOE life cycle follows the life cycle described in [PP-BSI-0035-2007]. 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 SAM purposes,
- Phase 2 "IC design",
- Phase 3 "IC manufacturing",
- Phase 4 "IC packaging" and Phase 5 "Composite product integration" that cover the composite product finishing process, preparation and shipping to the SAM personalization line
- Phase 6 "Personalisation" where the SAM administration data are loaded into the SAM's memory, following the personalization instructions manual,

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

The TOE can be delivered at the end of phases 3, 4, 5 or 6. All the phases before TOE delivery are evaluated according to the EAL4+ requirements. The TOE protects itself in Phase 7. Phases between TOE delivery and Phase 7 are handled according to standard banking best practices. Personalization activities, for example, are conducted using standard secure protocols such as SCP02. Security requirements are defined by BMS concerning theses phases and are agreed upon by means of dedicated contracts with each entity performing them. The requirements are enforced by conducting periodic security audits.

1.4.5 Protection Profile Usage

The requirements presented in this PP define the minimum security rules a Security Target (ST) of a SAM shall conform to, but are in no way exhaustive.

It remains indeed possible to add functionalities or also refer to another PP. However, any modifications to this PP are restricted by the rules defined by the conformance as set forth in Section 1.5.

1.5 Common Criteria Conformance

1.5.1 Conformance claim to CC

This Protection Profile is CC Part 2 conformant [CC2] and CC Part 3 [CC3] conformant.

1.5.2 Conformance claim to a package

The minimum assurance level for the evaluation of a SAM against this PP is EAL4 augmented with AVA_VAN.5 and ALC_DVS.2 components.

1.5.3 Conformance claim of the PP

This PP does not claim conformance to any another PP.

1.5.4 Conformance claim to the PP

The conformance to this PP, required for the Security Targets and Protection Profiles claiming conformance to it, is **strict**, as defined in CC Part 1 [CC1].

2 Security problem definition

2.1 Assets

In the following, the description of each asset provides the type of protection required for each asset.

As the main objective of the EM system is to preserve the EM flow, the assets of the SAM that shall be protected are the following ones:

2.1.1 Assets protected by the TOE

Electronic Money (EM)

Electronic Money (EM) is an electronic substitute of funds stored by the SAM. It corresponds to the total amount of payment transactions since the last collect transaction. The 3 last values of EM are recorded in the SAM.

Protection: integrity

SAM Id

The SAM Id is a unique sequence of numbers assigned to the SAM that identifies it.

Protection: integrity.

2.1.2 TSF data

Flow traceability data

Flow traceability data stands for information on the last payment and quickload transactions stored in log files.

Protection: integrity.

Keys

The secret key of the SAM, used for authentication purposes.

Protection: integrity and confidentiality.

Application note:

Keys hold information like:

- a key identifier (ID),
- a key version number,
- a counter of failed access.

If too many failures occur when using a key, this one is blocked and cannot be used anymore.

SAM application code

The application embedded in the SAM.

Protection: integrity.

Sequence counters

The sequence counters count the successive transactions:

- collect transactions,
- quickload transactions and credit operations,
- parameters update transactions.

Protection: integrity.

State of the SAM

The state of the SAM stores information about the SAM internal states during its usage phase.

Protection: integrity.

Application note:

The behaviour of the SAM 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).

Static counters

Static counters cover the configuration (parameters) of the SAM:

- the maximum quickload amount,
- the maximum number of quickload transactions between two collect transactions,
- the maximum accumulated quickload amount between two collect transactions.

Protection: integrity.

2.2 Users

The users of the TOE are the entities who interact with the SAM through its physical or logical external interfaces.

2.3 Threats

The threats described hereafter apply to the usage phase of the TOE. The threat agent or "attacker" is a hostile user that is physically and logically outside the TOE. The threat agent wishes to abuse the assets of the TOE by functional, physical or logical attacks or any combination of them.

2.3.1 COUNTERFEITING

The creation of fake transactions by falsification of assets in order to create or lose EM.

T.COUNTERFEITING_COLLECT

Counterfeiting of a collect transaction in order to credit the merchant with a financial counterpart greater or lesser than the EM collected from the SAM; it leads to unauthorized EM creation or loss.

Related asset: EM.

T.COUNTERFEITING_CREDIT

Counterfeiting of a payment transaction in order to credit the SAM with an EM amount greater or lesser than the EM amount specified in the transactions; it leads to unauthorized EM creation or loss.

Related asset: EM.

T.COUNTERFEITING_UPDATE

Counterfeiting of a parameters update transaction in order to change the keys values.

Related assets: Keys.

2.3.2 DISCLOSURE

Unauthorised disclosure of assets.

T.DISCLOSURE_KEYS

Unauthorised access to the secret keys.

Related assets: Keys.

2.3.3 LOSS OF INTEGRITY

Unauthorized modification of assets.

T.INTEG_CODE

Unauthorized modification of the TOE code: an attacker modifies the code in order to bypass the security policy of the SAM.

Related asset: SAM application code.

T.INTEG_EM

Unauthorized modification of stored EM: an attacker modifies the amount of EM stored in the SAM in order to increase or decrease the amount.

Related asset: EM.

T.INTEG_FLOW_TRA_DATA

Unauthorised modification of stored flow traceability data: an attacker modifies the log of the last transactions in order to hide potentially malicious operations performed on the SAM.

Related asset: Flow traceability data.

T.INTEG_KEYS

Unauthorized modification of stored keys: an attacker modifies the value of the secret keys and associated attributes stored in the SAM in order to input a known key.

Related assets: Keys.

T.INTEG_SAM_ID

Unauthorized modification of stored SAM Id: an attacker modifies the value of the SAM Id stored in the SAM in order to input another one.

Related asset: SAM Id.

T.INTEG_SEQ_COUNT

Unauthorized modification of stored sequence counters: an attacker modifies the value of sequence counters in order to force the SAM to accept counterfeited or replayed transactions.

Related assets: Sequence counters.

T.INTEG_SM

Unauthorized modification of the State Machine: an attacker modifies or deletes information that defines the current state of the SAM in order, for instance, to bypass a secure state.

Related asset: State of the SAM.

T.INTEG_STATIC_COUNT

Unauthorized modification of stored static counters: an attacker modifies the value of static counters which define the configuration of the SAM in order to bypass controls or limitations enforced by the EM system.

Related assets: Static counters.

2.3.4 REPLAY

Replay of a previous transaction or the last transaction. Such a replay can be performed immediately after the first sending of the transaction or later.

T.REPLAY_CREDIT

Replay of a payment transaction: a SAM is credited several times via a previous complete sequence of credit operations; it leads to unauthorized EM creation.

Related asset: EM.

T.REPLAY_UPDATE

Replay of a parameters update transaction: a SAM is updated several times via a previous complete sequence of parameters update operations; it leads to fraudulent changes of keys stored in the SAM.

Related assets: keys.

2.3.5 STEALING

Stealing of the TOE.

T.STEALING

An attacker steals the SAM of the legitimate merchant in order to use it for his own purpose.

2.3.6 FAILURE

A failure occurring during a transaction leads to a non-secure state inducing vulnerabilities in the SAM.

T.FAIL_BYPASS

The attacker may disrupt the SAM during a transaction in order, for instance, to bypass a control.

Related assets:EM, Static counters, Sequence counters, State of the SAM, SAM application code.

Application note:

This attack is typically realised by a fault injection method.

2.4 Organisational Security Policies

The following organisational security policies are mandatory for the TOE:

OSP.AGGREGATE

During a payment transaction, the SAM is able to aggregate the credit amounts (associated to EMs) to its global overall amount. The result is a new total with a value equivalent to the sum of all the amounts.

OSP.DEBIT_BEFORE_CREDIT

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

OSP.MANAGEMENT_OF_SECRETS

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 organisational 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 unauthorised disclosure and modification and allow to detect potential security violations.

OSP.MERCHANT_BEHAV

The merchant shall keep the SAM and shall not lend it, especially to untrusted persons.

2.5 Assumptions

The following assumption concerns the TOE operational environment, after the TOE delivery.

A.PROTECTION AFTER TOE DELIVERY

The TOE is assumed to be protected by the environment after delivery (may range from Phase 3 to 6) and before entering the final usage phase (Phase 7 of the life cycle). 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.

Note: 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.

3 Security Objectives

3.1 Security Objectives for the TOE

The main security objective for the TOE is to ensure EM flow preservation. The TOE shall implement state of art technology to achieve the following derived security objectives.

O.AGGREGATE

During a payment transaction, the SAM shall aggregate the amounts of the credited EM to the amount stored in the SAM. The result is the total amount of payment transactions since the last collect transaction.

O.AUTH

The SAM shall enforce mutual authentication with external devices (EP and distant server to perform update) prior any transaction.

O.CONF_DATA

The SAM shall prevent unauthorized disclosure of confidential TSF data: the keys.

O.EM

The SAM shall prevent unauthorized creation or loss of EM.

O.INTEG_DATA

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

O.LIMIT

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

O.OPERATE

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

O.RECORD

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

O.REPLAY

The SAM shall detect and reject replayed transactions.

O.TAMPER

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

3.2 Security objectives for the Operational 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 unauthorised 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 merchant the rules dealing with the use of the SAM. Especially, he must inform the user that he must keep its TOE in a trusted place to protect it from stealing.

3.3 Security Objectives Rationale

3.3.1 Threats

3.3.1.1 COUNTERFEITING

T.COUNTERFEITING_COLLECT This threat is countered by:

- O.CONF_DATA and O.INTEG_DATA that prevent the unauthorized disclosure or modification of data,
- O.EM which ensures EM flow preservation so that fraudulent creation of EM in the SAM using a collect transaction 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.

T.COUNTERFEITING_CREDIT This threat 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 of EM in the SAM using a debit transaction 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.

T.COUNTERFEITING_UPDATE This threat 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 SAM using a parameter update transaction 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.

3.3.1.2 DISCLOSURE

T.DISCLOSURE_KEYS This threat is countered by:

- O.CONF_DATA which prevent 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 asset.

3.3.1.3 LOSS OF INTEGRITY

T.INTEG_CODE This threat 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 This threat 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 SAM using a collect transaction or credit operation 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.INTEG_FLOW_TRA_DATA This threat 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.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.TAMPER which ensures security information and especially security assets of the TOE cannot be physically tampered.

T.INTEG_KEYS This threat 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_SAM_ID This threat 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_SEQ_COUNT This threat 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_SM This threat 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_STATIC_COUNT This threat 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.

3.3.1.4 REPLAY

T.REPLAY_CREDIT This threat is countered by:

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

T.REPLAY_UPDATE This threat is countered by:

- O.REPLAY which ensures the SAM will operate in a continuous secure state in case of replayed parameters update transaction; the replayed transaction will be detected and rejected by the SAM.

3.3.1.5 STEALING

T.STEALING This threat is countered by:

- O.LIMIT which avoids
  - the storage of important amount of EM coming from quickload transaction in the SAM,
  - the execution of too much quickload transactions which would not be actually processed afterward,
- OE.TOE-USAGE which ensures the TOE issuer provides to the user the rules to securely use its TOE.

3.3.1.6 FAILURE

T.FAIL_BYPASS This threat is countered by:

- O.TAMPER which ensures the TOE cannot be disrupted using physical interfaces.

3.3.2 Organisational Security Policies

OSP.AGGREGATE This OSP is directly covered by O.AGGREGATE.

OSP.DEBIT_BEFORE_CREDIT This OSP is directly covered by OE.DEBIT_BEFORE_CREDIT.

OSP.MANAGEMENT_OF_SECRETS OE.MANAGEMENT_OF_SECRETS directly covers the organisational security policy.

OSP.MERCHANT_BEHAV This OSP is covered by:

- OE.TOE-USAGE which ensures that the merchant is aware of the security rules related to the TOE.

3.3.3 Assumptions

A.PROTECTION AFTER TOE DELIVERY OE.PROTECTION_AFTER_TOE_DELIVERY directly covers the assumption.

3.3.4 SPD and Security Objectives

Threats	Security Objectives	Rationale
T.COUNTERFEITING_COLLECT	O.EM, O.RECORD, O.CONF_DATA, O.INTEG_DATA	Section 3.3.1
T.COUNTERFEITING_CREDIT	O.EM, O.RECORD, O.AUTH, O.CONF_DATA, O.INTEG_DATA	Section 3.3.1
T.COUNTERFEITING_UPDATE	O.EM, O.RECORD, O.AUTH, O.CONF_DATA, O.INTEG_DATA	Section 3.3.1
T.DISCLOSURE_KEYS	O.CONF_DATA, O.TAMPER	Section 3.3.1
T.INTEG_CODE	O.INTEG_DATA, O.TAMPER, O.OPERATE	Section 3.3.1
T.INTEG_EM	O.TAMPER, O.INTEG_DATA, O.EM, O.OPERATE	Section 3.3.1
T.INTEG_FLOW_TRA_DATA	O.INTEG_DATA, O.TAMPER, O.OPERATE, O.RECORD	Section 3.3.1
T.INTEG_KEYS	O.INTEG_DATA, O.TAMPER, O.OPERATE	Section 3.3.1
T.INTEG_SAM_ID	O.INTEG_DATA, O.TAMPER, O.OPERATE	Section 3.3.1
T.INTEG_SEQ_COUNT	O.INTEG_DATA, O.TAMPER, O.OPERATE	Section 3.3.1
T.INTEG_SM	O.INTEG_DATA, O.TAMPER, O.OPERATE	Section 3.3.1
T.INTEG_STATIC_COUNT	O.INTEG_DATA, O.TAMPER, O.OPERATE	Section 3.3.1
T.REPLAY_CREDIT	O.EM, O.REPLAY	Section 3.3.1
T.REPLAY_UPDATE	O.REPLAY	Section 3.3.1
T.STEALING	OE.TOE-USAGE, O.LIMIT	Section 3.3.1
T.FAIL_BYPASS	O.TAMPER	Section 3.3.1

Table 1 Threats and Security Objectives - Coverage

Security Objectives	Threats
O.AGGREGATE	
O.AUTH	T.COUNTERFEITING_CREDIT, T.COUNTERFEITING_UPDATE
O.CONF_DATA	T.COUNTERFEITING_COLLECT, T.COUNTERFEITING_CREDIT, T.COUNTERFEITING_UPDATE, T.DISCLOSURE_KEYS
O.EM	T.COUNTERFEITING_COLLECT, T.COUNTERFEITING_CREDIT, T.COUNTERFEITING_UPDATE, T.INTEG_EM, T.REPLAY_CREDIT
O.INTEG_DATA	T.COUNTERFEITING_COLLECT, T.COUNTERFEITING_CREDIT, T.COUNTERFEITING_UPDATE, T.INTEG_CODE, T.INTEG_EM, T.INTEG_FLOW_TRA_DATA, T.INTEG_KEYS, T.INTEG_SAM_ID, T.INTEG_SEQ_COUNT, T.INTEG_SM, T.INTEG_STATIC_COUNT
O.LIMIT	T.STEALING
O.OPERATE	T.INTEG_CODE, T.INTEG_EM, T.INTEG_FLOW_TRA_DATA, T.INTEG_KEYS, T.INTEG_SAM_ID, T.INTEG_SEQ_COUNT, T.INTEG_SM, T.INTEG_STATIC_COUNT
O.RECORD	T.COUNTERFEITING_COLLECT, T.COUNTERFEITING_CREDIT, T.COUNTERFEITING_UPDATE, T.INTEG_FLOW_TRA_DATA
O.REPLAY	T.REPLAY_CREDIT, T.REPLAY_UPDATE
O.TAMPER	T.DISCLOSURE_KEYS, T.INTEG_CODE, T.INTEG_EM, T.INTEG_FLOW_TRA_DATA, T.INTEG_KEYS, T.INTEG_SAM_ID, T.INTEG_SEQ_COUNT, T.INTEG_SM, T.INTEG_STATIC_COUNT, T.FAIL_BYPASS

Security Objectives	Threats
OE.DEBIT_BEFORE_CREDIT	
OE.MANAGEMENT_OF_SECRETS	
OE.PROTECTION_AFTER_TOE_DELIVERY	
OE.TOE-USAGE	T.STEALING

Table 2 Security Objectives and Threats - Coverage

Organisational Security Policies	Security Objectives	Rationale
OSP.AGGREGATE	O.AGGREGATE	Section 3.3.2
OSP.DEBIT_BEFORE_CREDIT	OE.DEBIT_BEFORE_CREDIT	Section 3.3.2
OSP.MANAGEMENT_OF_SECRETS	OE.MANAGEMENT_OF_SECRETS	Section 3.3.2
OSP.MERCHANT_BEHAV	OE.TOE-USAGE	Section 3.3.2

Table 3 OSPs and Security Objectives - Coverage

Security Objectives	Organisational Security Policies
O.AGGREGATE	OSP.AGGREGATE
O.AUTH	
O.CONF_DATA	
O.EM	
O.INTEG_DATA	
O.LIMIT	
O.OPERATE	
O.RECORD	
O.REPLAY	
O.TAMPER	
OE.DEBIT_BEFORE_CREDIT	OSP.DEBIT_BEFORE_CREDIT
OE.MANAGEMENT_OF_SECRETS	OSP.MANAGEMENT_OF_SECRETS
OE.PROTECTION_AFTER_TOE_DELIVERY	
OE.TOE-USAGE	OSP.MERCHANT_BEHAV

Table 4 Security Objectives and OSPs - Coverage

Assumptions	Security objectives for the Operational Environment	Rationale
A.PROTECTION AFTER TOE DELIVERY	OE.PROTECTION_AFTER_TOE_DELIVERY	Section 3.3.3

Table 5 Assumptions and Security Objectives for the Operational Environment - Coverage

Security objectives for the Operational Environment	Assumptions
OE.DEBIT_BEFORE_CREDIT	
OE.MANAGEMENT_OF_SECRETS	
OE.PROTECTION_AFTER_TOE_DELIVERY	A.PROTECTION AFTER TOE DELIVERY
OE.TOE-USAGE	

Table 6 Security Objectives for the Operational Environment and Assumptions - Coverage

4 Security Functional Requirements

4.1 Security Functional Requirements

The Security Target author shall instantiate all the operations in the following Security Functional Requirements according to Moneo Secure Access Module specifications.

4.1.1 Authentication

FDP_DAU.1 Basic Data Authentication

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of **EM**.

Refinement:

The EM validity is only relevant for the collect transactions.

FDP_DAU.1.2 The TSF shall provide **the acquirer device** with the ability to verify evidence of the validity of the indicated information.

FIA_UAU.1/Distant_server Timing of authentication

FIA_UAU.1.1/Distant_server The TSF shall allow

- **to transfer information from the SAM to the distant server,**
- **to stop parameters update transactions,**
- **the distant server to authenticate the SAM,**
- **[assignment: list of TSF-mediated actions]**

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

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

Refinement:

The "user" stands for the distant server.

FIA_UAU.1/EP Timing of authentication

FIA_UAU.1.1/EP The TSF shall allow

- **the interaction between the EP and the SAM in order to exchange the information necessary for performing quickload transactions (including key index and versions to be used for authentication),**

- **to stop debit operation of the EP and quickload transactions,**
- **to authenticate the EP**
- **[assignment: list of TSF-mediated actions]**

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

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

Refinement:

The "user" stands for the Electronic Purse.

FIA_UAU.6/EP Re-authenticating

FIA_UAU.6.1/EP The TSF shall re-authenticate the user under the conditions

- **beginning of a payment transaction,**
- **beginning of a quickload transaction**.

Refinement:

The "user" stands for the Electronic Purse.

FIA_UAU.6/Distant_server Re-authenticating

FIA_UAU.6.1/Distant_server The TSF shall re-authenticate the user under the conditions

- **beginning of a parameters update transaction**.

Refinement:

The "user" stands for the distant server.

FIA_UAU.3 Unforgeable authentication

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.

Refinement:

The "user" stands for the Electronic Purse, the acquirer device or the distant server.

FIA_UAU.4 Single-use authentication mechanisms

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

- **the EP authentication mechanism,**
- **the distant server authentication mechanism,**
- **[assignment: identified authentication mechanism(s)]**.

4.1.2 Storage integrity

FDP_SDI.1 Stored data integrity monitoring

FDP_SDI.1.1 The TSF shall monitor user data stored in containers controlled by the TSF for **any integrity errors** on all objects, based on the following attributes: **[assignment: user data attributes]**.

4.1.3 Security properties on communications

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 trusted IT product from unauthorised disclosure during transmission.

Refinement:

The "trusted IT product" stands for the distant server. The "transmission" occurs during the parameters update transaction and the "TSF data" stands for SAM parameters being updated (transaction keys).

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 trusted IT product within the following metric: **[assignment: a defined modification metric]**.

FPT_ITI.1.2 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.

Refinement:

The "trusted IT products" stands for the distant server. The "transmission" occurs during the parameters update transaction and the "TSF data" stands for SAM parameters being updated (transaction keys).

FPT_RPL.1 Replay detection

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

- **credit transactions,**
- **quickload transactions,**
- **parameters update transactions**.

FPT_RPL.1.2 The TSF shall perform

- **the abort of the transaction in process,**
- **[other specific actions]**

when replay is detected.

4.1.4 Access control security policy

FDP_ACC.2 Complete access control

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

Refinement:

The "objects" under this security access control policy refer to the user and TSF data. The "subjects" are the entities that can access these data through specific "operations".

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: **[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]**.

FDP_ACF.1.2 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]**.

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: **[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]**.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]**.

Refinement:

Access control rules shall enforce confidentiality and integrity of objects, depending on their characteristics.

FMT_MSA.1/Assets Management of security attributes

FMT_MSA.1.1/Assets The TSF shall enforce the **Assets Security policy** 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 authorised identified roles]**.

FMT_MSA.3/Assets Static attribute initialisation

FMT_MSA.3.1/Assets The TSF shall enforce the **Assets Security policy** to provide **[selection: choose one of: restrictive, permissive, [assignment: other property]]** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Assets The TSF shall allow the **[assignment: the authorised identified roles]** to specify alternative initial values to override the default values when an object or information is created.

4.1.5 Flow control security policy

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the **Transaction policy** on **[assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]**.

Refinement:

The "operations" under this security flow control policy refer to the debit, load, quickload, collect and parameters update transactions. The "subjects" are the entities that play an active role in the execution of those transactions and the "information" is the data transmitted during the transactions.

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: **[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 rules]**.

FDP_IFF.1.4 The TSF shall explicitly authorise an information flow based on the following rules: **[assignment: rules, based on security attributes, that explicitly authorise information flows]**.

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **[assignment: rules, based on security attributes, that explicitly deny information flows]**.

Application note:

Rules shall typically address the mutual authentication between the SAM and the EP or the Acquirer device, the conditions on the amounts that can be loaded on or debited from the EP depending on various maximum limit amounts, and the number of transactions of a given a type allowed by the SAM. The instantiation of this requirement shall comply with the rules defined in the Moneo specifications.

FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1 The 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: **[assignment: additional importation control rules]**.

Application note:

"User data" stands for user and TSF data imported during collect transactions, quickload transactions or credit operations processed by the SAM.

FMT_MSA.1/Transaction Management of security attributes

FMT_MSA.1.1/Transaction 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 authorised identified roles]**.

FMT_MSA.3/Transaction Static attribute initialisation

FMT_MSA.3.1/Transaction 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.

FMT_MSA.3.2/Transaction The TSF shall allow the **[assignment: the authorised identified roles]** to specify alternative initial values to override the default values when an object or information is created.

4.1.6 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 operations,**
- **last quickload transactions,**
- **[assignment: other specifically defined auditable events]**.

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, **[assignment: other audit relevant information]**.

Application note:

Date and time of the event are determined by the terminal hosting the SAM.

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide **[assignment: authorised users]** with the capability to read **[assignment: list of audit information]** 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 unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to **detect** unauthorised modifications to the stored audit records in the audit trail.

4.1.7 Fail safe

FPT_RCV.4 Function recovery

FPT_RCV.4.1 The TSF shall ensure that **credit, collect, quickload 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.

4.1.8 Cryptography and random generation

FCS_COP.1 Cryptographic operation

FCS_COP.1.1 The TSF shall perform **[assignment: list of cryptographic operations]** in accordance with a specified cryptographic algorithm **[assignment: cryptographic algorithm]** and cryptographic key sizes **[assignment: cryptographic key sizes]** that meet the following: **[M-CRYPTO]**.

Refinement:

The "list of cryptographic operations" shall include at least all the operations that support the mutual authentication of SAM and external IT product especially the EP and the distant server).

FIA_SOS.2 TSF Generation of secrets

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet **[assignment: random numbers quality metric]**.

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for **[assignment: list of TSF functions]**.

Refinement:

"Secrets" stands for random values.

Application note:

The "quality metric" shall meet national schemes requirements (e.g. [CRYPTO] in France).

4.1.9 Platform and IC protection

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 **[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.

Application note:

The TSF shall rely on its Integrated Circuit certified against [PP-BSI-0035-2007] to detecty physical attacks. The security target author shall explain in the TOE Summary Specification how the IC and the embedded software cooperate to implement this requirement.

FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1 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.

Application note:

The TSF shall rely on its Integrated Circuit certified against [PP-BSI-0035-2007] to resist to physical tampering scenarios. The Security Target author shall explain in the TOE Summary Specification how the IC and the embedded software cooperate to implement this requirement.

4.2 Security Assurance Requirements

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

4.3 Security Requirements Rationale

4.3.1 Objectives

4.3.1.1 Security Objectives for the TOE

O.AGGREGATE This objective is covered by:

- FDP_IFF.1, FDP_IFC.1, FDP_ITC.1, FMT_MSA.3/Transaction and FMT_MSA.1/Transaction which ensure the TOE is able to aggregate the EM amounts to its global overall amount (BAL respectively BAL') when the TOE processes a load, respectively a quickload transaction.

O.AUTH This objective is covered by:

- FIA_UAU.1/EP, FIA_UAU.1/Acquirer_device, FIA_UAU.1/Distant_server which require the authentication of the corresponding external device to the TOE to perform 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/EP, FIA_UAU.6/Distant_server which require the re-authentication of the corresponding external device to the TOE each time a transaction with an external device needs to be (re)initiated,
- FDP_DAU.1 which requires the authentication of the TOE before performing a transaction with the EP, the acquirer device or the distant server,
- FIA_SOS.2 which ensures the TOE can generate random value to perform authentication processes.

O.CONF_DATA This objective 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.3/Assets and FMT_MSA.1/Assets which ensures that security assets cannot be retrieved without passing by a secure access control,
- FPT_PHP.2 and FPT_PHP.3 that address 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.

O.EM This objective is directly covered by:

- FDP_IFC.1, FDP_IFF.1, FDP_ITC.1, FMT_MSA.3/Transaction and FMT_MSA.1/Transaction which enforce an information flow control policy applicable to 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 load, quickload, 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 which ensures that efficient cryptographic means are used to protect EM on the TOE and during transactions.

This specific objective is also indirectly covered by all the SFRs related to the other objectives. Indeed, these objectives ensure the correct and secure operation of the TOE which guarantees that it is not possible to create or loss EM (i.e. modify) from outside or even inside the TOE.

O.INTEG_DATA This objective 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.3/Assets and FMT_MSA.1/Assets 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 unauthorised deletion and detects any attack against this security asset.

O.LIMIT This objective is covered by:

- FDP_IFF.1, FDP_IFC.1, FMT_MSA.3/Transaction and FMT_MSA.1/Transaction 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 quickload transaction.

O.OPERATE This objective 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. Indeed, such an unstable state could lead to incorrect behaviour of the TOE.

O.RECORD This objective 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.

O.REPLAY This objective is covered by:

- FPT_RPL.1 which ensures that load, quickload, 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.

O.TAMPER This objective is covered by:

- FPT_PHP.2 which requires detection of the physical tampering,
- FPT_PHP.3 which requires the protection against physical tampering.

4.3.2 Rationale tables of Security Objectives and SFRs

Security Objectives	Security Functional Requirements	Rationale
O.AGGREGATE	FDP_ITC.1, FDP_IFC.1, FDP_IFF.1, FMT_MSA.3/Transaction, FMT_MSA.1/Transaction	Section 4.3.1
O.AUTH	FIA_UAU.1/EP, FIA_UAU.6/EP, FIA_UAU.1/Distant_server, FIA_UAU.6/Distant_server, FIA_UAU.3, FIA_UAU.4, FDP_DAU.1, FIA_SOS.2	Section 4.3.1
O.CONF_DATA	FPT_ITC.1, FDP_ACC.2, FDP_ACF.1, FPT_RCV.4, FPT_PHP.2, FPT_PHP.3, FMT_MSA.3/Assets, FMT_MSA.1/Assets	Section 4.3.1
O.EM	FPT_RPL.1, FDP_ITC.1, FDP_IFC.1, FDP_IFF.1, FPT_RCV.4, FCS_COP.1, FMT_MSA.3/Transaction, FMT_MSA.1/Transaction	Section 4.3.1
O.INTEG_DATA	FDP_SDI.1, FPT_ITI.1, FDP_ACC.2, FDP_ACF.1, FAU_STG.1, FPT_RCV.4, FPT_PHP.2, FPT_PHP.3, FMT_MSA.3/Assets, FMT_MSA.1/Assets	Section 4.3.1
O.LIMIT	FDP_IFC.1, FDP_IFF.1, FMT_MSA.3/Transaction, FMT_MSA.1/Transaction	Section 4.3.1
O.OPERATE	FPT_RCV.4	Section 4.3.1
O.RECORD	FAU_GEN.1, FAU_SAR.1	Section 4.3.1
O.REPLAY	FPT_RPL.1, FIA_SOS.2	Section 4.3.1
O.TAMPER	FPT_PHP.2, FPT_PHP.3	Section 4.3.1

Table 7 Security Objectives and SFRs - Coverage

Security Functional Requirements	Security Objectives
FDP_DAU.1	O.AUTH
FIA_UAU.1/Distant_server	O.AUTH
FIA_UAU.1/EP	O.AUTH
FIA_UAU.6/EP	O.AUTH
FIA_UAU.6/Distant_server	O.AUTH
FIA_UAU.3	O.AUTH
FIA_UAU.4	O.AUTH
FDP_SDI.1	O.INTEG_DATA
FPT_ITC.1	O.CONF_DATA
FPT_ITI.1	O.INTEG_DATA
FPT_RPL.1	O.EM, O.REPLAY
FDP_ACC.2	O.CONF_DATA, O.INTEG_DATA
FDP_ACF.1	O.CONF_DATA, O.INTEG_DATA
FMT_MSA.1/Assets	O.CONF_DATA, O.INTEG_DATA
FMT_MSA.3/Assets	O.CONF_DATA, O.INTEG_DATA
FDP_IFC.1	O.AGGREGATE, O.EM, O.LIMIT
FDP_IFF.1	O.AGGREGATE, O.EM, O.LIMIT
FDP_ITC.1	O.AGGREGATE, O.EM
FMT_MSA.1/Transaction	O.AGGREGATE, O.EM, O.LIMIT
FMT_MSA.3/Transaction	O.AGGREGATE, O.EM, O.LIMIT
FAU_GEN.1	O.RECORD
FAU_SAR.1	O.RECORD
FAU_STG.1	O.INTEG_DATA
FPT_RCV.4	O.CONF_DATA, O.EM, O.INTEG_DATA, O.OPERATE
FCS_COP.1	O.EM
FIA_SOS.2	O.AUTH, O.REPLAY
FPT_PHP.2	O.CONF_DATA, O.INTEG_DATA, O.TAMPER
FPT_PHP.3	O.CONF_DATA, O.INTEG_DATA, O.TAMPER

Table 8 SFRs and Security Objectives

4.3.3 Dependencies

4.3.3.1 SFRs dependencies

Requirements	CC Dependencies	Satisfied Dependencies
FDP_DAU.1	No dependencies	
FIA_UAU.1/Distant_server	(FIA_UID.1)	
FIA_UAU.1/EP	(FIA_UID.1)	
FIA_UAU.6/EP	No dependencies	
FIA_UAU.6/Distant_server	No dependencies	
FIA_UAU.3	No dependencies	
FIA_UAU.4	No dependencies	
FDP_SDI.1	No dependencies	
FPT_ITC.1	No dependencies	
FPT_ITI.1	No dependencies	
FPT_RPL.1	No dependencies	
FDP_ACC.2	(FDP_ACF.1)	FDP_ACF.1
FDP_ACF.1	(FDP_ACC.1) and (FMT_MSA.3)	FDP_ACC.2
FMT_MSA.1/Assets	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_ACC.2
FMT_MSA.3/Assets	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/Assets
FDP_IFC.1	(FDP_IFF.1)	FDP_IFF.1
FDP_IFF.1	(FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1, FMT_MSA.3/Transaction
FDP_ITC.1	(FDP_ACC.1 or FDP_IFC.1) and (FMT_MSA.3)	FDP_IFC.1, FMT_MSA.3/Transaction
FMT_MSA.1/Transaction	(FDP_ACC.1 or FDP_IFC.1) and (FMT_SMF.1) and (FMT_SMR.1)	FDP_IFC.1
FMT_MSA.3/Transaction	(FMT_MSA.1) and (FMT_SMR.1)	FMT_MSA.1/Transaction
FAU_GEN.1	(FPT_STM.1)	
FAU_SAR.1	(FAU_GEN.1)	FAU_GEN.1
FAU_STG.1	(FAU_GEN.1)	FAU_GEN.1
FPT_RCV.4	No dependencies	
FCS_COP.1	(FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2) and (FCS_CKM.4)	
FIA_SOS.2	No dependencies	

Requirements	CC Dependencies	Satisfied Dependencies
FPT_PHP.2	(FMT_MOF.1)	
FPT_PHP.3	No dependencies	

Table 9 SFRs dependencies

Rationale for the exclusion of dependencies

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

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

The dependency FMT_MSA.3 of FDP_ACF.1 is unsupported. The dependency with FMT_MSA.3 is not relevant: security attributes are defined during development and manufacturing of the TOE and could not be modified during operational use.

The dependency FMT_SMF.1 of FMT_MSA.1/Assets is unsupported. This PP does not mandate any specific security management function. The ST author shall add FMT_SMF if necessary (for instance, if the standard CC operations for the management of security attributes are not enough or appropriate).

The dependency FMT_SMR.1 of FMT_MSA.1/Assets is unsupported. This PP does not mandate any specific security role. The ST author shall add FMT_SMR if necessary. Otherwise FMT_MSA.3.2 shall be filled in with "none".

The dependency FMT_SMR.1 of FMT_MSA.3/Assets is unsupported. This PP does not mandate any specific security role. The ST author shall add FMT_SMR if necessary. Otherwise FMT_MSA.3.2 shall be filled in with "none".

The dependency FMT_SMF.1 of FMT_MSA.1/Transaction is unsupported. This PP does not mandate any specific security management function. The ST author shall add FMT_SMF if necessary (for instance, if the standard CC operations for the management of security attributes are not enough or appropriate).

The dependency FMT_SMR.1 of FMT_MSA.1/Transaction is unsupported. This PP does not mandate any specific security role. The ST author shall add FMT_SMR if necessary. Otherwise FMT_MSA.3.2 shall be filled in with "none".

The dependency FMT_SMR.1 of FMT_MSA.3/Transaction is unsupported. This PP does not mandate any specific security role. The ST author shall add FMT_SMR if necessary. Otherwise FMT_MSA.3.2 shall be filled in with "none".

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 of no use for the TOE objectives.

The dependency FCS_CKM.1 or FDP_ITC.1 or FDP_ITC.2 of FCS_COP.1 is unsupported. The dependency with FCS_CKM.1 is not satisfied because key generation is not in the scope of the TOE. The dependency with FDP_ITC.1 or FDP_ITC.2 is not satisfied because this PP does not mandate any specific key importation method. The ST

author is allowed to extend the scope of the TOE and add requirements on these issues. Nevertheless, keys are particular TSF data that shall be protected against disclosure and modification, as specified in FPT_ITC.1 and FPT_ITI.1.

The dependency FCS_CKM.4 of FCS_COP.1 is unsupported. The dependency with FCS_CKM.4 "Cryptographic key destruction" is not relevant: destruction of the keys is out of the scope of the TOE.

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

4.3.3.2 SARs dependencies

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

Requirements	CC Dependencies	Satisfied Dependencies
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 10 SARs dependencies

4.3.4 Rationale for the Security Assurance Requirements

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

4.3.5 ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE. This assurance component is a higher hierarchical component to EAL4 (only ALC_DVS.1 is included in EAL4). Due to the nature of the TOE, there is a need to justify the sufficiency of these procedures to protect their confidentiality and integrity. ALC_DVS.2 has no dependencies.

4.3.6 AVA_VAN.5 Advanced methodical vulnerability analysis

The TOE must resist to high attack potential. This is due to the fact that the SAM can be placed in hostile environments, including electronic laboratories. This robustness level is achieved by the assurance requirement AVA_VAN.5. Independent vulnerability analysis is based on highly detailed technical information. The attacker is assumed to be thoroughly familiar with the specific implementation of the TOE. The attacker is presumed to have a high level of technical knowledge. AVA_VAN.5 has dependencies with ADV_ARC.1, ADV_FSP.4, ADV_TDS.3, ADV_IMP.1, AGD_PRE.1, AGD_OPE.1 and ATE_DPT.1. All these dependencies are satisfied by EAL4.

5 Notice

This document has been generated with TL SET version 2.3.6 (for CC3). For more information about the security editor tool of Trusted Labs visit our website at www.trusted-labs.com.

Annexe A Definitions and acronyms

A.1 Definitions

This section provides definitions about terms frequently used in this document. The definition of the Common Criteria related terms is specified in [CC1], § 4.

Acquirer device	An acquirer device is a server intended to handle the collect transactions.
Balance	The balance is the amount of EM stored in the EP (in a specific currency). It is increased by load and quickload transactions, and decreased by debit transactions: - BAL : The balance of the EP, loaded by load transactions. - BAL': The balance of the EP, loaded by quickload transactions.
BMS	The Billettique Monétique Services company (BMS) groups together the leading: - banking and financial establishments marketing Moneo such as BNP-Paribas, Banque Populaire, Caisse d'Épargne, Crédit Agricole, CIC, HSBC, LCL, Crédit Mutuel, La Banque Postale, Société Générale, - Carriers such as SNCF or RATP - Operators such as France Telecom. BMS is responsible for the design, marketing development and operation of the electronic purse on a multi-application smart card, capitalizing on the technological performance and achievements of French monetics and intermodal tele-cash dispensing.
Collect	One or several amounts of the EM corresponding to a set of payment transactions stored by a SAM are delivered to the EM issuer via acquirer device.
Credit (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 credit transaction when it's only related to the TOE.
Electronic Money	Electronic Money is the counter part of the fund received by the EM issuer and stored in the EP or the SAM. For the SAM, It corresponds to the total amount of payment transactions since the last collect transaction.

Acquirer device	An acquirer device is a server intended to handle the collect transactions.
EM issuer	The EM issuer guarantees the EM in an EM system. To this end, the EM issuer creates and dispenses EM in exchange for funds received, redeems collected EM and extinguishes it.
Electronic Purse	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 Electronic Money (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.
Integrated Circuit	Integrated Circuit is an electronic component designed to perform processing and/or memory functions.
Load device	A load device is a server. Its main functionality is to load the EM in the EP in exchange of funds via the EM issuer.
Load	The EP is credited with an amount of EM created by the EM issuer, via a load agent; the purse holder gives a corresponding amount of funds in turn.
Merchant	A merchant sells goods or services for which he accepts payment by EP. In order to handle the EM payment transactions, the merchant operates one or more purchase devices in which a SAM stores EM until collect. The merchant is responsible for the operational security of the purchase device he controls.
Purchase device	A 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), built on an integrated circuit module. The SAM shall provide the necessary security for the EM payment, the quickload and the collect transactions. It contains various parameters that could be updated.
Purse holder	The purse holder is the person that is in possession of the EP and uses it for EM payment transactions. Purse holders need to protect their EP as if it is cash.
Quickload	This operation is used to load EM into the EP. It is processed offline. 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.

Acquirer device	An acquirer device is a server intended to handle the collect transactions.
SAM	Secure Access Module, typically in an ID0 plug-in size form factor, placed in a Purchase Device. The SAM identifies the EP, checks the authenticity of the EP, encrypts and protects the exchanges with the EP and the Acquirer Device. It acts as an EM payment transactions certifier.
SFPMEI	Société Financière du Porte-Monnaie Électronique Interbancaire is the EM institution in France. It is approved by the authorities whose objective is to insure consumers of the deposits value made on an electronic purse, regardless of the companies responsible for its operation. The SFPMEI is an EM issuer.

A.2 Acronyms

CC	Common Criteria
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information
EAL	Evaluation Assurance Level
EM	Electronic Money
EP	Electronic Purse
IC	Integrated Circuit
OS	Operating System
OSP	Organisational Security Policy
PP	Protection Profile
SAM	Secure Access Module
SF	Security Function
TOE	Target Of Evaluation

Annexe B References

[CC1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model. Version 3.1, revision 3, July 2009. CCMB-2009-07-001.
[CC2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements. Version 3.1, revision 3, July 2009. CCMB-2009-07-002.
[CC3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements. Version 3.1, revision 3, July 2009. CCMB-2009-07-003.
[CEM]	Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. Version 3.1, revision 3, July 2009. CCMB-2009-07-004.
[CRYPTO]	Mécanismes cryptographiques. Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques. Version 1.11, 24 Octobre 2008, ANSSI. (This version or later applicable one.) Gestion des clés cryptographiques. Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques. Version 1.10, 24 Octobre 2008. ANSSI. (This version or later applicable one.)
[M-CRYPTO]	"Electronic Purse Moneo - Mécanismes cryptographiques", version 2.4 (or later applicable one), SFPMEI.
[ANSSI-CC-PP-2009/02]	Embedded Software for Smart Secure Devices Protection Profile, Basic and Extended configurations, v1.0, 27 November 2009, ANSSI.
[PP-BSI-0035-2007]	Security IC Platform Protection Profile, version 1.0, June 2007, BSI.
[PP-JCS-Collection]	JavaCard System Minimal Configuration Protection Profile version 1.0b, Sun Microsystems, Inc. PP/0303
	JavaCard System Standard 2.1.1 Configuration Protection Profilenversion 1.0b, Sun Microsystems, Inc. PP/0304
	JavaCard System Standard 2.2 Configuration Protection Profile version 1.0b, Sun Microsystems, Inc. PP/0305
	JavaCard System Defensive Configuration Protection Profile version 1.0b, Sun Microsystems, Inc. PP/0306
[PP-JCS-Closed]	Java Card System - Closed configuration Protection Profile, Sun Microsystems, Inc. *Under evaluation at the publication date of this PP.*
[PP-JCS-Open]	Java Card System - Open configuration Protection Profile, Sun Microsystems, Inc. *Under evaluation at the publication date of this PP.*

[PP-JCS]	Java Card System Protection Profiles, Closed configuration and Open configuration, SUN Microsystems.
[PP/0101]	Intersector Electronic Purse and Purchase Device (version without Last Purchase Cancellation), version 1.3, March 2001, SFPMEI.
[PP EP]	Electronic Purse Protection Profile, version 1.5, 4 February 2010, SFPMEI.

Index