Common Criteria for IT Security Evaluation
Protection Profile

Transactional Smartcard Reader
Protection Profile

Profil de Protection
pour un lecteur transactionnel
de cartes à puce

Version 2.0

Issue January 2000

Registered by the French Certification Body under the reference PP/0002

Any correspondence about this document should be referred to the following organisations :

- **Cyber-COMM**

  29, rue de Berri
  F-75008 Paris

- **Service Central de la Sécurité des Systèmes d'Information**

  Information Technology Security Certification Centre
  18, rue du Docteur Zamenhof
  F-92131 Issy-Les-Moulineaux
  Telephone : (+33) 1 41 46 37 84 Fax : (+33) 1 41 46 37 01

Table of contents

Table of Figures and Tables

1. Chapter 1

PP Introduction

1.1 PP identification

Title : Transactional Smartcard Reader Protection Profile.

Version : 2.0

Author : Cyber-COMM, 29 rue de Berri - 75008 Paris - FRANCE

Evaluation Assurance Level : EAL4 augmented.

Registration : PP/0002 given by the French certification body at the protection profile registration, as certified.

Compliant with Version 2.1 of Common Criteria.

Key words : Smartcard, smartcard reader, smartcard issuer, card scheme (a payment scheme in the context of a smartcard describing data exchange and rules inside a system between commercial partners), electronic commerce, PINpad.

A glossary of terms used in the PP is given in chapter 6.1.

A product compliant with this PP may also offer additional security functional requirements, depending on the application type.

1.2 PP overview

This Protection Profile elaborated in conformance with the French IT Security Evaluation and Certification Scheme is the work of the following organisation :

- Cyber-COMM, 29 rue de Berri 75008 Paris - FRANCE

The intent of this Protection Profile is to specify functional and assurance requirements applicable to a generic « transactional smartcard reader ».

Hereafter in this document the expression « transactional smartcard reader » will be replaced by « the Device ».

A smartcard is a piece of plastic, with an electronic component embedded in it. The electronic component is a micro-computer with internal memory, able to perform controls and calculations. One or several software applications, in the micro-computer, can activate those functions to use the smartcard as a token in a card scheme, providing to the card scheme security functions such as authentication, electronic signature generation or control.

Usually the smartcard is « personalised ». It means that before the use stage of the smartcard, it is given a unique identification number, and this unique identification number can be checked without any doubt by the card scheme.

The Device is to be used with personalised smartcards, by the legitimate owners of those smartcards. The security of the smartcard is out of scope of the TOE.

The Device is supposed to be used in a private environment. That is to say that the Device is to be used by an individual, or a small group of persons (limited to a number of two or three persons), in a place under control of this individual, or group of persons. The Device is not intended to be used in a public area.

The Device is a « smartcard reader ». It means that it has a smartcard connector and is able to interact with the smartcard.

The Device is a « transactional » smartcard reader. It can execute an application software (specified and checked by a card scheme) that can process a smartcard. This application software is out of scope of the PP.

The Device could be considered as a device gathering the capacity to interact with smartcard, to securely capture an authentication information and transaction data in order to pass it to the application software to generate secured transactions.

1.2.1 Smartcard reader classification

Smartcard readers, together with applications running on it, can be classified according to their security functions. The following table sums up the different levels of smartcard readers.

READERS	Level 2 PC-SC (Simple Reader)	Level 3 = Level 2 + **keypad**	Level 4 = Level 3+**display**	Level 5 = Level 4+ **Crypto**
Existing devices	Reader PC/SC Stand alone, Integrated in UC, PCMCIA	Modem Reader, Keyboard Reader	Mouse	Standalone secure PINpad keyboard, Set-top box , Screenphones
Security functions remotely **provable**	**Authentication card**	Authentication card **Authentication cardholder**	Authentication card Authentication cardholder	Authentication card Authentication cardholder **Integrity Non-repudiation**
APPLICATIONS				
Access control	Simple (= physical key)	Strong	Strong	Strong
Identification	**OK**	OK	OK	OK
Authentication	**Card**	Card **Cardholder**	Card Cardholder	Card Cardholder **Terminal**
Loyalty	**OK**	OK	OK	OK
Micro-payments	**OK**	OK	OK	OK
e-Purse	**If Online**	If Online	If Online	**Offline**
Debit/credit	No PIN Repudiation risk	**With PIN** Repudiation risk	With PIN Repudiation risk	With PIN **Non Repudiation**
Home Banking	**Control card**	Control card **Control Cardholder**	Control card Control Cardholder Repudiation risk	Control card Control Cardholder **Non Repudiation**
Electronic Signature				**Non repudiation**
Business to Business	**Control Card**	Control Card **Control Cardholder**	Control Card Control Cardholder **Contracts Orders Receipts**	Control Card Control Cardholder **Provable Contracts Firmed Orders Doc. Credit Firmed receipts Non repudiation**

Table 1 : Smartcard reader classification

Comments :

- Level 1 is not in this table, this is a reader with no security (smartcard is not directly used).
- Level 2 is available, but provides only object authentication, though smartcard is used as a token.
- Level 3 provides cardholder authentication.
- Level 4 is not far from level 3.
- Only level 5 can provide provable non-repudiation.
- Only level 5 provides secure downloading.
- Only level 5 is remotely upgradable.
- Only level 5 is secure multi-applications.

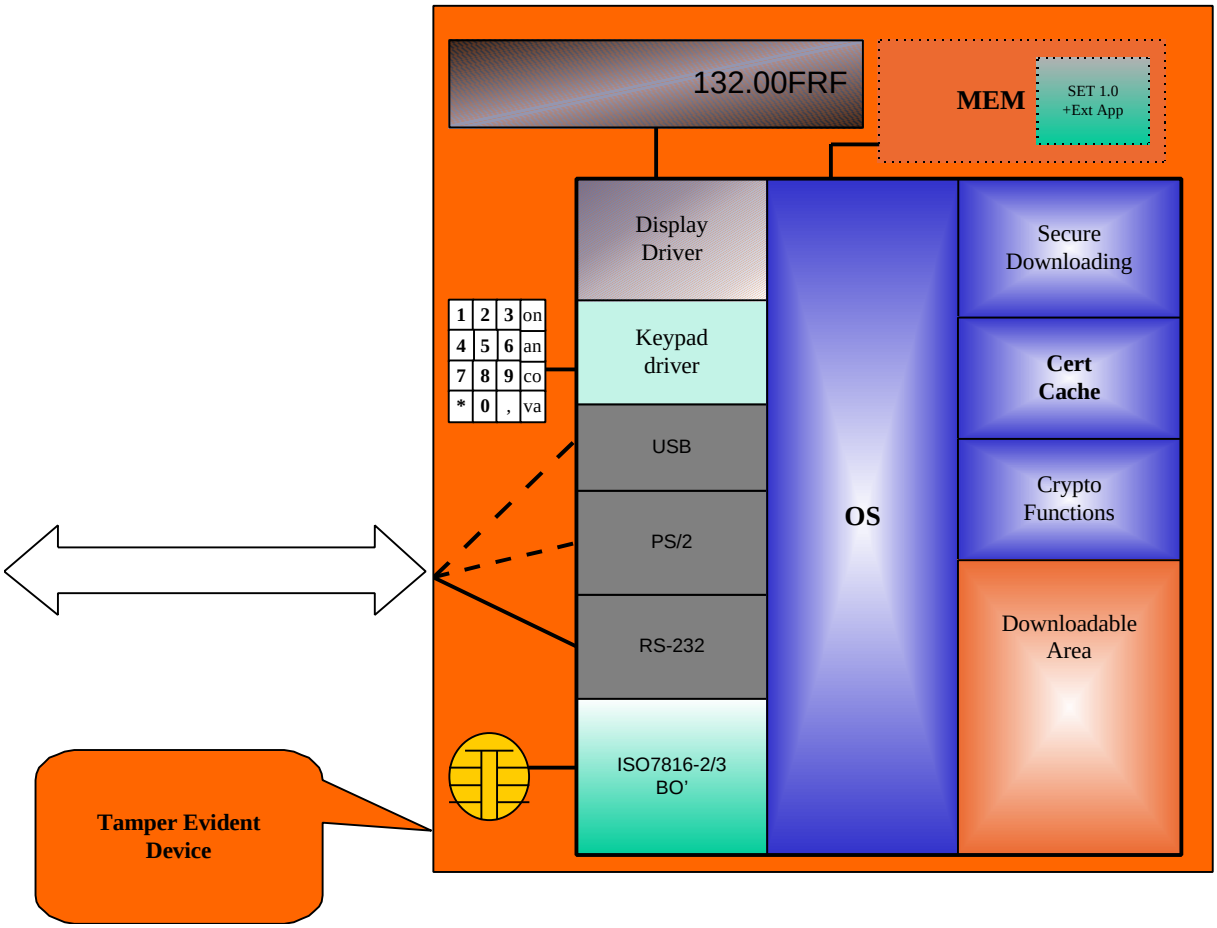

Figure 1 - An example of level 5 smartcard reader

The smartcard reader considered in this PP corresponds to the Level 5.

1.2.2 PP objectives

The main objectives of this Protection Profile are:

- to describe the Target of Evaluation (TOE) as a product and position it in its life cycle ;
- to describe the security environment of the TOE including the assets to be protected and the threats to be countered by the TOE and by the operational environment during the development production and user phases ;
- to describe the security objectives of the TOE and its supporting environment in terms of integrity and confidentiality of application data and programs, protection of the TOE and associated documentation during the development and production phases ;
- to specify the IT security requirements which includes the TOE functional requirements and the TOE IT Assurance requirements.

2. Chapter 2

TOE Description

2.1 Description of the TOE

This part of the PP describes the TOE as an aid to the understanding of its security requirements and addresses the product type, the intended usage and the general IT features of the TOE.

The **Target Of Evaluation** is a finished product that includes a smartcard reader (the device without the application), which enforces all the TOE Security Functions.

The security functions described in the present Protection Profile are thus restricted to those of the Device included in the TOE except the application. The TOE is depicted hereafter :

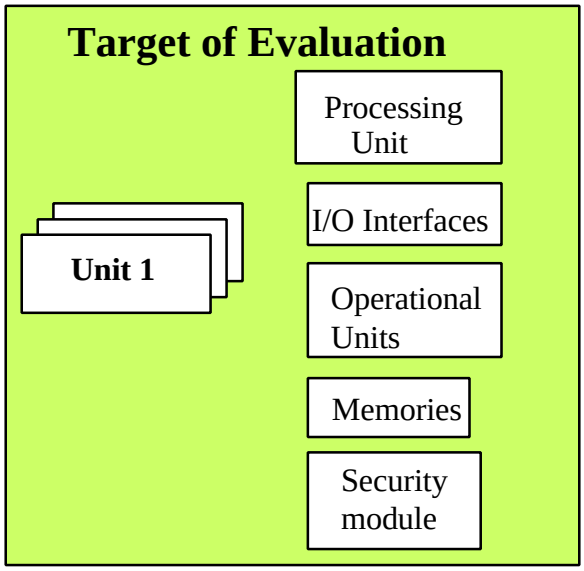

Figure 2 - TOE and scope of the PP

2.2 TOE Intended usage

In this PP, the user is the application loaded in the TOE. When other sorts of users are aimed at they will be explicitly mentioned.

A typical TOE can provide one of the following services :

- GSM device for banking transaction initiated by SIM card ;
- pay-TV device (set-top box) where the smartcard is used as a pay-per-view card ;

- PC connected device for electronic commerce over the Internet (or other open network) ;
- Webphone device ;
- Others, to be defined in the future.

2.3 Product Type

The typical Device is composed of :

- I/O interfaces (keyboard, display, printer, chip coupler, etc.) ;
- a processing unit ;
- memory components (RAM, ROM, EEPROM, etc.) ;
- one or more operational unit(s) according to the destination of the product ;
- physical and logical security barriers (shields, security module, etc.).

The security barriers are the critical parts of the Device. They contribute to the protection of cryptographic resources. This protection can be provided using a shielded envelop around the components where cryptographic resources are used, or by a single component (security module) from which the cryptographic resources are never exported.

The Device is intended to receive application software that is not part of the TOE.

The security module may include proprietary firmware, either embedded in non-volatile memories, either downloadable in conditional erasable memories, or both.

The security module is an IT component that gathers the security functions related to the smartcard processing. It has its own processing unit, memories and security barriers. It manages the cryptographic operations related to the processing of security functions.

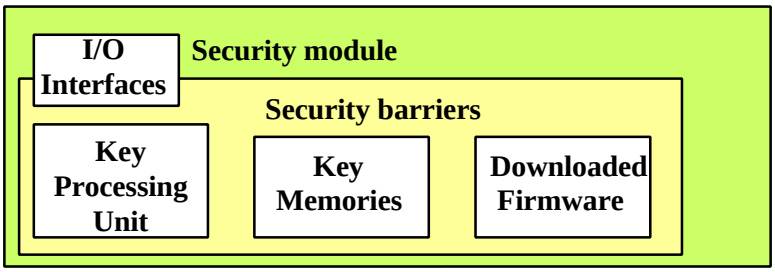

Figure 3 - Security module

In the above Figure, security barriers are any kind of physical (shields, captors, etc.) or logical protection (active responses, memory erase, etc.) fulfilling the security requirements.

2.4 Device services

The Device provides the functions for which it has been designed. Some of these functions invoke security functions that are supplied by the security module.

The Device provides services to the finished product, among which :

- secure interaction with a smartcard ;
- security services related to the processing of data transaction.

Some of these security functions are, but not limited to :

- cryptographic key management of keys used by the security functions ;
- cryptographic operation processing for application running on the device, such as encryption, decryption, signature construction and verification ;
- uniqueness of identification/authentication of the Device towards outside.

2.5 Device Life Cycle

The Device life-cycle is decomposed as described in Table 2.

- **design stage** : device (hardware and firmware) and application software design and development;
- **manufacturing stage** : manufacturing and testing ;
- **personalisation stage** : initial key loading in the device to make it a unique device (distinguished identifier), and firmware / software signature ;
- **pre-use stage** : device packaging and delivery ;
- **use stage :** firmware/software downloading, use of the device by the end user and maintenance of the device by the device maintainer;
- **post-use stage** : device end of life process.

Main stage	Life-cycle phase	Transition event	Description
Design	Hardware and firmware design and development	Completion and testing	The designer is in charge of the design of the Device hardware and firmware so that it incorporates the intended functional and physical and logical characteristics of that Device. The firmware can be split in a kernel firmware that contains the minimum bootstrap code needed to begin a process, and some downloadable firmware.
	Application design and development	*Completion and testing*	*The application provider is in charge of the design and development of the application software intended to use the basic functionality of the Device.*
	Software updating	*Update*	*The application software administrator is responsible for the availability of an up to date software for a remote loading of the device.*
Manufacturing	Device manufacturing and testing	Completion	The Device manufacturer is responsible for producing the Device.
Personalisation	Initial key loading and testing	Loading	The Device manufacturer security administrator is in charge of loading the appropriate cryptographic keys in order to fulfil the security requirements.
	Downloadable firmware signature	Signature	If such component exists it must be integrity and authenticity protected.
	Application software signature	*Signature*	*The application provider must ensure integrity and authenticity of the application software. The security administrator is in charge of the key management and of the operation of the application software signature.*
Pre-use	Device packaging and delivery	Installation	The Device manufacturer is responsible for packaging, and shipping of the Device to its intended location for use. According to the type of the device, it can be delivered to the end user directly or via a distributor (e.g. retailer, Bank, etc.).
Use	Firmware and application software downloading	Completion	The end-user is in charge of downloading the appropriate software from appropriate servers.
	Device use	Removal	The end-user is in charge of the Device security during its use. Users of the device are applications that are activated by the end user.
	Device maintenance	Re-installation Repair, upgrade completion	The Device maintainer is in charge of the secure handling of the Device during this phase.
Post-use	Device removal from service	Destruction	The device maintainer is also particularly in charge of the correct completion of the Device destruction when appropriate.

Table 2 : Product Life-Cycle

The grey parts of the table above represent stages covered by the scope of the present PP.

The Device is integrated in the TOE during the pre-use stage.

In this table, 'personalisation' consists of giving a distinguished reference to each physical Device. This can be done by embedding a serial number or a cryptographic key in a non-volatile memory, or all other similar technique.

A transition event is the event that causes transition from one phase to the next. It is important to notice that a phase is a space of time during which the Device can be idle before a new event changes its state. For example after entering in personalisation phase and before completion of Device personalisation the Device can be stored in a room, waiting for the next personalisation session. A threat agent could take advantages of these « *idle* » times.

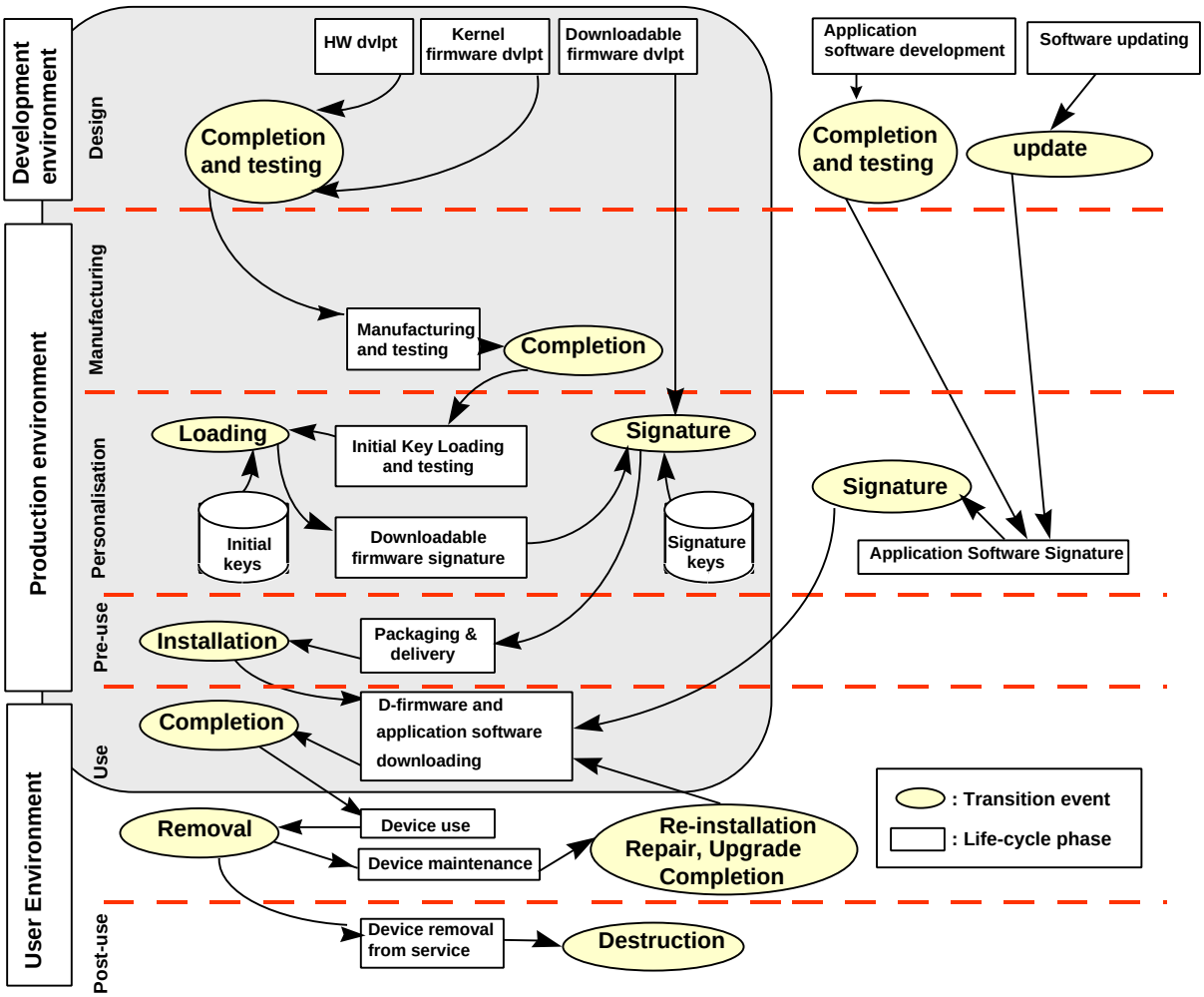

Figure 4 : Device product life-cycle

The part of the product life-cycle, covered by the present PP is limited to the grey part.

The Device runs under an operating system composed of hardware components, firmware and software. The firmware can be composed of a part embedded in the hardware (the kernel) and a part that can be changed remotely.

Applications (software) can be downloaded in the Device to provide expected functionalities of the product.

Security requirements exist to ensure mutual authentication and integrity of the Device and the elements it controls.

The following Figure shows a possible implementation of firmware and application layers and their interweaving.

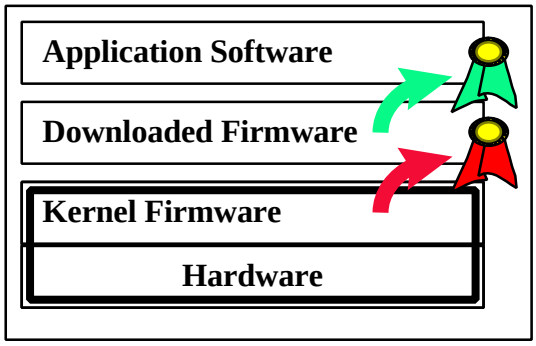

Figure 5 : Layered Device Design

2.6 Device Environment

Considering the Device, three types of environments are defined :

- Development environment ;
- Production environment ;
- User environment.

2.6.1 Roles

During its life cycle, the security of the Device is managed by various roles :

- The **designer** : he is in charge of the Device security during the design phase (hardware and firmware : kernel and downloaded parts).
- The **manufacturer** : he is in charge of the Device production. In many cases the designer and the manufacturer are from the same company but can be operated by separate business units. The manufacturer is in charge of the personalisation of the Device (hardware and software when appropriate).
- The designer/manufacturer **security administrator** : he is in charge of
  - the key management of keys he is in charge of,
  - the secure loading of initial keys in the Device,
  - the operation of firmware signature.

- The **application provider** : he is in charge of the design and development of the application software that will be run in the Device. He is in charge of the integrity and authenticity of the application software before signature.
- The **security administrator** during the personalisation phase : he is in charge of
  - the key management of keys he is in charge of,
  - the secure loading of a unique-per-device key in the Device,
  - the operation of application software signature.
- The **application software administrator** : he is responsible for the availability of an up to date software for a remote loading of the device.
- The **distributor** : according to the type of TOE, the TOE can be delivered to the end user directly or via a distributor (e.g. retailer, Bank, etc.).
- The **user** : this is an application that is activated by an end user.
- The **end user** is defined as a human entity external to the TOE but who interacts with the TOE through an application interface.
- The **device maintainer** is in charge of the secure handling of the Device during the maintenance phase. He is also in charge of the correct completion of the Device destruction when appropriate.

The application provider role and the application software administrator role are out of the scope of the PP.

2.6.2 Development Environment

The development begins with the Device specification. All parties in contact with sensitive information are required to abide by Non-Disclosure Agreements.

2.6.3 Production Environment

The following stages are achieved in the production environment :

- the manufacturing stage during which the Device is built and tested ;
- the personalisation stage during which initial key loading is achieved in order to make the Device distinguishable and able to prove its identity ;
- the packaging and delivery phase during which the Device is inserted in the TOE and the finished product is shipped to its intended recipient.

2.6.4 User Environment

The Device is used in a wide range of applications to assure reading of smartcards and processing of sensitive information in a secure way. Typical user (=application) environments are home-banking, electronic commerce, TV set-top boxes, mobile phones, etc.

The user (=application) environment therefore covers a wide spectrum of very different intended use, thus making it difficult to avoid and monitor any abuse of the TOE.

2.7 General IT features of the TOE

In its smartcard reader application, the TOE IT functionalities consist of:

- interacting with other finished product units : units outside the TOE request the TOE to provide services linked with the use of a smartcard ;
- interacting with a smartcard : the application requests the smartcard to provide internal smartcard resources (data, process) ;
- generating secret or private cryptographic keys ;
- distributing secret or private cryptographic keys ;
- deleting secret or private cryptographic keys ;
- storing secret or private cryptographic keys ;
- providing security services to an application :
  - running arithmetical functions ;
  - running cryptographic operations (encryption, digital signature, hashing) ;
  - processing data received from outside the TOE ;
  - formatting and securing transactions for outside the TOE.

3. Chapter 3

Security environment

This section describes the security aspects of the environment in which the TOE is intended to be used and addresses the description of the assets to be protected, the threats, the organisational security policies and the secure usage assumptions.

3.1 Assets requiring protection

Assets are security relevant elements of the TOE that include the following categories :

1. **Internal TOE assets**
   The following assets are typical to the TOE and are considered as TSF data. They are loaded in the TOE at some phases of its life cycle, that are in the development environment or in the manufacturing environment. Assets are presented under logical and physical aspects :

   - the cryptographic resources (cryptographic functions) provided by the security module and the key processing unit where they are operated ;

     ⇒ these assets require integrity

   - the cryptographic keys and the key memories of the security module where they are stored ;

     ⇒ these assets require integrity and confidentiality

   - the connections between the processing unit and the key memories;

     ⇒ these assets require confidentiality

   - the firmware (Kernel and downloaded parts) ;

     ⇒ these assets require integrity and authenticity

   Cryptographic resources and cryptographic keys are downloaded in the same way than firmware. In the following, their downloading is considered as part of firmware downloading.

   In the context of the maintenance stage, the firmware before downloading is considered as an external TOE asset (user data).

2. **External TOE assets**

   The following assets belong to the application downloaded in the device. They are considered as user data and only appear after the application downloading, that is in the user environment, and stay out of the scope of the TOE even if application data and keys are resources occupying memories within the scope of the TOE.

- an **application software** running on the TOE ;
- **application data and keys**

  $\Rightarrow$ these assets require integrity and authenticity

Application data and keys are downloaded in the same way than application software. In the following, their downloading is considered as part of application software downloading.

3.2 Assumptions

This section describes the assumptions that must be satisfied by the TOE environment.

3.2.1 Assumptions upon the development environment

A_DESIGN.01 : the designer issues and maintains a written procedure describing the security rules, and applies it in the development environment.

A_DESIGN.02 : the designer and its security administrator ensure protection of cryptographic keys involved in the design stage, and especially during the firmware signature phase.

3.2.2 Assumptions upon the production environment

A_MANUF.01 : the manufacturer issues and maintains a written procedure describing the security rules, and applies it in the production environment.

A_MANUF.02 : the manufacturer and its security administrator ensures protection of cryptographic keys involved in the personalisation stage, and especially during the firmware and application software signature phase.

A_MANUF.03 : the manufacturer ensures the security of the TOE during packaging and delivery phases up to its intended use location.

3.2.3 Assumptions upon the user environment

A_RESP : users are informed of their responsibility when using the TOE.

A_PRIVATE : the TOE is intended to be used in a private environment.

A_APPLI.01 : the TOE is intended to be used with an application software complying with a smartcard scheme.

A_APPLI.02 : security requirements at the application level exist to ensure mutual authentication and integrity between the firmware and the applications.

3.3 Threat agents

A threat agent to the TOE can be :

- **an end user :** an end user is a person who has received a product in an authorised way and who wants to alter transaction data of an application or to counterfeit the normal processing of an application.
- **a designer/manufacturer security administrator :** this is an operator entitled to process key management, secure loading of initial keys in the device and the operation of firmware signature.
- **an application provider** in charge of the design and development of the application software who wants to alter it in a fraudulent way. The application provider is considered as a threat agent because he has the capability of inserting a trojan horse into an application he is developing. As a user, this application may try to access to internal assets of the TOE or to external assets belonging to other applications.
- the **application software administrator** : he is responsible of the availability of an up to date software for a remote loading of the device.
- **a security administrator (during the personalisation phase):** this is an operator entitled to process key management in order to personalise the TOE. He/she wants to modify the TOE cryptographic keys.
- **an aggressor :** this is a person who has not received a product in an authorised way or otherwise gains illicit access to the TOE, and who wants directly or through an application :
  - to replace at least one of the internal TOE assets by fake ones ;
  - to alter the TOE to use it in an unauthorised manner ;
  - to tamper the TOE in order to obtain application data or keys.

3.4 Description of the threats

The TOE as defined in chapter 2 is required to counter the threats described hereafter; a threat agent wishes to abuse the assets, either by functional attacks or by environmental manipulations, or by specific hardware manipulations or by any other type of attacks.

From ISO/FDIS 13491-1 :1997 : Banking - Secure cryptographic devices (retail) Part 1 : Concepts, requirements and evaluation methods

Attacks scenarios :

- **Penetration** : this is an active attack which involves the physical perforation or unauthorised opening of the device to ascertain sensitive data contained within it, for example, cryptographic keys. Therefore, penetration is an attack on the physical characteristics of the device.

- **Monitoring** : this is a passive attack which may involve the monitoring of electromagnetic radiation for the purposes of discovering sensitive information contained within the device; or visually, aurally, or electronically monitoring secret data being entered into the device. Therefore, monitoring is an attack on the physical characteristics of the device.
- **Manipulation** : this is the unauthorised sending to the device of a sequence of inputs so as to cause the disclosure of sensitive information or to obtain a service in an unauthorised manner, for example, causing the device to enter its "test mode" in order that sensitive information could be disclosed or the device integrity manipulated. Manipulation is an attack on the logical characteristics of the device.
- **Modification** : this is the unauthorised modification or alteration of the logical or physical characteristics of the device, for example, inserting a PIN-disclosing "bug" in a PIN pad between the point of PIN entry and the point of PIN encryption. Note that modification may involve penetration but for the purpose of altering the device rather than disclosing information contained within the device. The unauthorised replacement of a cryptographic key contained within a device is a form of modification. Modification is an attack on either the physical or logical characteristics of the device.
- **Substitution** : this is the unauthorised replacement of one device with another. The replacement device might be a look-alike "counterfeit" or emulating device having all or some of the correct logical characteristics plus some unauthorised functions, such as a PIN-disclosing bug. The replacement device might be a once-legitimate device that had been subject to unauthorised modifications and then substituted for another legitimate device. Removal is a form of substitution which may be carried out in order to perform a penetration or modification attack in an environment better suited to such attacks, or as a first step in a substitution attack, the device may be taken out of its operating environment. Substitution can be seen as a special case of modification in which the adversary does not actually modify the target device but instead replaces it with a modified substitute. Substitution is an attack on the physical and logical characteristics of the device.

3.4.1 Threats to internal TOE assets

T_INTERN.01 : an aggressor may replace the TOE by a similar equipment where some components have been modified (cloning).

T_INTERN.02 : one (or more) cryptographic resource(s) of the TOE is (are) altered by a manufacturer security administrator or a security administrator or an aggressor before or during the personalisation phase where proper monitoring, administrative oversight, and/or procedures are not in place.

T_INTERN.03 : one (or more) cryptographic resource(s) of the TOE is (are) altered during the use stage by an aggressor who may turn the

TOE in an insecure state due to conditions that occur internal to the function.

T_INTERN.04 : one (or more) cryptographic key(s) is (are) altered by a security administrator or an aggressor who misuse his/her privilege.

T_INTERN.05 : one (or more) cryptographic key(s) is (are) disclosed by a security administrator or an aggressor who misuse his/her privilege.

T_INTERN.06 : the internal connections of the TOE where cryptographic keys circulate can be accessed by an aggressor.

T_INTERN.07 : the firmware is altered by an aggressor in order to bypass security controls.

3.4.2 Threats to external TOE assets

T_EXTERN.01 : a threat agent may alter or replace an application software in a manner that is not consistent with the security policy.

T_EXTERN.02 : external assets may be exposed to an environment where proper physical and/or procedural controls are not locally in place, and allow one of the following threat agents : an application provider, an application software administrator or an aggressor to access it.

3.5 Organisational security policies

This section describes the security policies with which the TOE must comply :

P_PRODUCT.01 : a user of the TOE cannot break the integrity nor the confidentiality of the assets belonging to another user.

P_PRODUCT.02 : the security module must take control on the processor (internal or external to the device) of the TOE in order to control access to the external interfaces of the TOE.

P_IDENT : the TOE shall be able to provide with a unique identifier to an appropriate verifier and give an evidence of its identity.

4. Chapter 4

Security objectives

The security objectives of the TOE mainly cover the following aspects :

- the TOE must only operate upright and authentic resources ;
- the TOE must ensure integrity of downloaded application and related data ;
- the TOE must protect internal data ;
- the TOE must be provable towards outside of the TOE.

4.1 Security objectives for the TOE

O_TOE.01 : the TOE must ensure the confidentiality of cryptographic keys it manages during their storage and use.

O_TOE.02 : the TOE must ensure the integrity of cryptographic resources and keys it manages during their storage and use.

O_TOE.03 : the TOE must ensure the protection of the connections between processing unit and memories of the security module.

O_TOE.04 : the TOE must ensure authentication of firmware and/or application downloaded in the TOE.

O_TOE.05 : the TOE must ensure the integrity of external TOE assets (including application data and keys) during their storage and use in the TOE, and whatever the form of the data (electronically stored or displayed on a screen).

O_TOE.06 : the TOE must provide a self protection against tampering.

O_TOE.07 : the TOE must ensure the continued correct operation of its security functions.

O_TOE.08 : the TOE must be able to generate the evidence of its distinguish identity to an appropriate verifier according to an appropriate security policy.

O_TOE.09 : the TOE must be protected against internal technical failures.

4.2 Security objectives for the environment

Security objectives related to TOE environments :

O_ENV.01 : end-users must be informed of their responsibilities when using the TOE.

O_ENV.02 : the TOE must not be diverted from its intended usage.

O_ENV.03 : management and use of the TOE must not endanger assets managed by the TOE.

O_ENV.04 : at each stage of its life cycle the entity in charge of the TOE must issue and maintain a writing procedure to apply during the stage.

O_ENV.05 : in development and production environment, whenever cryptographic keys are used the entity in charge of the TOE must ensure protection of cryptographic keys.

O_ENV.06 : in production environment, the modification of cryptographic keys must require authentication of the security administrator, and generate a secure audit trail of the modification.

5. Chapter 5

IT Security requirements

5.1 TOE security functional requirements

A minimum strength of functions claim consistent with the TOE Security Objectives is SOF-Medium.

5.1.1 Class FCS : Cryptographic Support

5.1.1.1 Cryptographic key management (FCS_CKM)

FCS_CKM.1 Cryptographic key generation

FCS_CKM.1.1 : The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

FCS_CKM.2 Cryptographic key distribution

FCS_CKM.2.1 : The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: *cryptographic key distribution method*] that meets the following: [assignment: *list of standards*].

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 : The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

5.1.1.2 Cryptographic operation (FCS_COP)

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: [assignment: *list of standards*].

list of cryptographic operations :

- peer authentication

- symmetric key generation
- message digest calculation
- MAC calculation
- internal key protection
- digital signature generation and verification
- encipherment and decipherment of data
- key distribution

5.1.2 Class FDP : User Data Protection

5.1.2.1 Data authentication (FDP_DAU)

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 [assignment: *list of objects or information types*].

FDP_DAU.1.2 : The TSF shall provide [assignment: *list of subjects*] with the ability to verify evidence of the validity of the indicated information.

5.1.3 Class FIA : Identification and authentication

5.1.3.1 Authentication failures (FIA_AFL)

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 : The TSF shall detect when [assignment: *number*] unsuccessful authentication attempts occur related to [assignment: *list of authentication events*].

number = 1 attempt

list of authentication events = digital signature failure

FIA_AFL.1.2 : When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [assignment: *list of actions*].

list of actions = send an error message and stop

5.1.3.2 User authentication (FIA_UAU)

FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 : The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Refinement 1 : authenticated = to verify the digital signature of the user loaded in the TOE.

Refinement 2 : user = application or downloaded firmware.

5.1.3.3 User identification (FIA_UID)

FIA_UID.2 User identification before any action

FIA_UID.2.1 : The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Refinement : user = an application.

5.1.4 Class FMT : Security management

5.1.4.1 Class FMT : Actions to be taken for management

The following actions could be considered for the management functions in FMT.

Function	Actions	Function	Actions	Function	Actions
FCS_CKM.1	a)	FIA_UAU.2	a)	FPT_ITT.1	a), b)
FCS_CKM.2	a)	FIA_UID.2	a)	FPT_PHP.3	a)
FCS_CKM.4	a)	FMT_MOF.1	a)	FPT_RCV.2	a), b)
FCS_COP.1	NM	FMT_MTD.1	a)	FPT_TST.1	a)
FDP_DAU.1	a)	FMT_SMR.1	NA		
FIA_AFL.1	a), b)	FPT_FLS.1	NM		

(a,b) : refers to the respective management defined in part 2 of CC V2.1
NM : No management activity
NA : Not applicable

Table 3 : Management activity versus functional requirements

5.1.4.2 Management of functions in TSF (FMT_MOF)

FMT_MOF.1 Management of security functions behaviour

FMT_MOF.1.1 : The TSF shall restrict the ability to [selection: *determine the behaviour of, disable, enable, modify the behaviour of*] the functions [assignment: *list of functions*] to [assignment: *the authorised identified roles*].

5.1.4.3 Management of TSF data (FMT_MTD)

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 : The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear,* [assignment: *other operations*]] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].

5.1.4.4 Security management roles (FMT_SMR)

FMT_SMR.1 Security roles

FMT_SMR.1.1 : The TSF shall maintain the roles [assignment: *the authorised identified roles*].

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

Refinement : users = security administrators.

5.1.5 Class FPT : Protection of the TOE Security Functions

5.1.5.1 Fail secure (FPT_FLS)

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 : The TSF shall preserve a secure state when the following types of failures occur: [assignment: *list of types of failures in the TSF*].

5.1.5.2 Internal TOE TSF data transfer (FPT_ITT)

FPT_ITT.1 Basic internal TSF data transfer protection

FPT_ITT.1.1 : The TSF shall protect TSF data from [selection: *disclosure, modification*] when it is transmitted between separate parts of the TOE.

5.1.5.3 TSF Physical protection (FPT_PHP)

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 TSP is not violated.

5.1.5.4 Trusted recovery (FPT_RCV)

FPT_RCV.2 Automated recovery

FPT_RCV.2.1 :When automated recovery from a failure or service discontinuity is not possible, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

FPT_RCV.2.2 :For [assignment: *list of failures/service discontinuities*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

5.1.5.5 TSF self test (FPT_TST)

FPT_TST.1 TSF testing

FPT_TST.1.1 : The TSF shall run a suite of self tests [selection: *during initial start-up, periodically during normal operation, at the request*

of the authorised user, at the conditions [assignment: *conditions under which self test should occur*]] to demonstrate the correct operation of the TSF.

FPT_TST.1.2 : The TSF shall provide authorised users with the capability to verify the integrity of TSF data.

FPT_TST.1.3 : The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

Refinement : users = device maintainer, security administrator.

5.2 TOE security assurance requirements

The targeted evaluation assurance level is EAL4 augmented, summarised in the Table 6.5 of Common Criteria Part 3.

The following specific augmentation components are added to EAL4 :

Ref.	Component	Name
1	ADV_IMP.2	Implementation of the TSF
2	AVA_VLA.3	Moderately resistant

Table 4 : Components added to EAL4

ADV_IMP.2 Implementation of the TSF

The developer shall provide the implementation representation for the entire TOE security functions.

AVA_VLA.3 Moderately resistant

The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

The developer shall document the disposition of identified vulnerabilities.

6. Chapter 6

Application notes

6.1 Definitions

In this Protection Profile, the following definitions apply :

PINpad : a PIN entry device complying with ISO 9564-1.

Smartcard : A smart card is a credit-card-sized plastic card that contains a general-purpose microprocessor (typically an 8-bit microcontroller such as a Motorola 6805 or an Intel 8051). The microprocessor is underneath a gold contact pad located on one side of the card. It has a non volatile memory and a processing unit embedded within it.

7. Chapter 7

Rationale

7.1 Introduction

This chapter presents the evidence used in the PP evaluation. This evidence supports the claims that the PP is a complete and cohesive set of requirements and that a conformant TOE would provide an effective set of IT security countermeasures within its security environment.

7.2 Security objectives rationale

This section demonstrates that the stated security objectives address all the security environment aspects identified.

Ref.	A / T / P	Security objectives	Written evidence
01	A_RESP	O_ENV.01, O_ENV.03	Direct matching. Using the TOE without endangering the assets managed by the TOE means that users are practically aware of the right way of using the TOE.
02	A_PRIVATE	O_ENV.02	The objective contributes to keep the TOE in a private environment.
03	A_APPLI.01	O_ENV.02	The destination of the TOE is to be used in conjunction with a smartcard.
04	A_APPLI.02	O_ENV.03	The TOE authenticates downloading of applications and protects access to cryptographic resources by maintaining integrity controls.
05	A_DESIGN.01	O_ENV.03, O_ENV.04	Management of the TOE will not endanger the assets it manages if in development environment, a written procedure and security rules are maintained. Direct matching.
06	A_DESIGN.02	O_ENV.03, O_ENV.05	Management of the TOE will not endanger the cryptographic keys it manages if during the firmware signature phase, a written procedure and security rules are maintained. Direct matching.
07	A_MANUF.01	O_ENV.03, O_ENV.04	Management of the TOE will not endanger the assets it manages if in production environment, a written procedure and security rules are maintained. Direct matching.

08	A_MANUF.02	O_ENV.03, O_ENV.05, 0_ENV.06	Management of the TOE does not endanger the cryptographic keys it manages if during the personalisation stage, a written procedure and security rules are maintained. Direct matching. Authentication of the security administrator and generation of a secure audit trail cover this assumption.
09	A_MANUF.03	O_ENV.03	Management of the TOE does not endanger the assets it manages if during packaging and delivery phases, a written procedure and security rules are maintained.
10	T_INTERN.01	O_TOE.02, O_TOE.04, O_TOE.05, O_TOE.06, O_TOE.08, O_ENV.04, O_ENV.05, O_ENV.06	Checking integrity and authenticity of the TOE components guarantees that they have not been replaced by modified versions of these components. Integrity checking of application keys during their storage and use in the TOE do not allow their replacement, without control, by other keys (O_TOE.05). Self protection of the TOE against tempering do not allow replacement of physical components nor the altering, without visible notification, of existing ones. Verification of the distinguished identity of the TOE guarantees that any cloning of the device will be detected. The TOE components are protected by writing procedures that are applied during the TOE life-cycle. This guarantees that security administrators cannot access them without control. Cryptographic keys are protected by the entities in charge of the TOE in production environment whenever used or modified.
11	T_INTERN.02	O_TOE.02, O_ENV.04	Checking integrity of the TOE cryptographic resources guarantees that they cannot be modified without detection. Cryptographic resources of the TOE are protected by writing procedures that are applied during the TOE life-cycle. This guarantees that security administrators cannot access them without control (O_ENV.04).
12	T_INTERN.03	O_TOE.02, O_TOE.07, O_TOE.09, O_ENV.04	Checking integrity of the TOE cryptographic resources guarantees that they cannot be modified without detection. Continued correct operation guaranty and protection against internal technical failures provided by the TOE protects its security functions from being altered during their use. Cryptographic resources of the TOE are protected by writing procedures that are applied during the TOE life-cycle. This guarantees that security administrators

			cannot access them without control.
13	T_INTERN.04	O_TOE.02, O_ENV.05, O_ENV.06	Checking integrity of the TOE cryptographic resources guarantees that they cannot be modified without detection during their storage. Direct matching since cryptographic keys are protected by the entities in charge of the TOE in production environment whenever used or modified.
14	T_INTERN.05	O_TOE.01, O_ENV.05, O_ENV.06	Confidentiality of cryptographic keys is provided by the TOE during their storage and use. Direct matching since cryptographic keys are protected by the entities in charge of the TOE in production environment whenever used or modified.
15	T_INTERN.06	O_TOE.01, O_TOE.02, O_TOE.03, O_TOE.06	Confidentiality of cryptographic keys is provided by the TOE during their storage and use. This means that they are necessarily protected during internal communications. Checking integrity of the TOE cryptographic resources guarantees that they cannot be modified during internal communications without detection. Protection of the connections between processing units and memories which is provided by the TOE protects cryptographic keys during their conveying through internal connections. Self protection against tampering guarantees that any attempt to intercept connections between processing unit and memories will fail or at least be detected. This protects cryptographic keys which are conveyed through internal connections.
16	T_INTERN.07	O_TOE.04, O_TOE.06, O_TOE.07	Authentication of the firmware contributes to protect it against alteration. Hence its replacement cannot be done without detection. Self protection against tampering guarantees that any attempt to alter the firmware will fail. Continued correct operation guaranty provided by the TOE protects the behaviour of firmware security functions from being altered during their use.
17	T_EXTERN.01	O_TOE.05	Direct matching.
18	T_EXTERN.02	O_TOE.03, O_TOE.05,	Protection of the connections between processing units and memories which is provided by the TOE protects external assets during their conveying through internal connections. Integrity checking of application keys during their storage and use in the TOE do not allow their replacement, or altering without control.

		O_TOE.06,	Self protection against tampering guarantees that any attempt to alter application software running on the TOE or associated data and keys will fail.
		O_ENV.03,	Management operations are performed in a such secured way that it doesn't introduce risks of undetectable alteration or replacement of application software running on the TOE or associated data and keys.
		O_ENV.04	External assets of the TOE are protected by writing procedures that are applied during the TOE life-cycle. This guarantees that security administrators cannot access them without control.
19	P_PRODUCT.01	O_TOE.03,	Protection of the connections between processing units and memories which is provided by the TOE protects external assets during their conveying through internal connections.
		O_TOE.05	Direct matching.
20	P_PRODUCT.02	O_TOE_01,	Guaranty of confidentiality of cryptographic keys in the TOE during their storage and use means that they are necessarily protected when manipulated within the internal processor of the security module.
		O_TOE.02,	Guaranty of integrity of the TOE cryptographic resources during internal communications without detection requires that a control is performed which avoids their altering by an attacker who takes control of the external interfaces of the TOE.
		O_TOE.05,	The TOE maintains integrity of external TOE assets during their storage and use whatever be the form of the data.
		O_TOE.06	Self protection against tampering requires that any attempt to alter application software running on the TOE or associated data and keys from external interfaces must be detected. The TOE should ensure that by controlling processors within it.
21	P_IDENT	O_TOE.08	Direct matching.

Table 5 : Matching Assumptions/Threats/Policies - Security objectives

	O_TOE.01	O_TOE.02	O_TOE.03	O_TOE.04	O_TOE.05	O_TOE.06	O_TOE.07	O_TOE.08	O_TOE.09
T_INTERN.01		✓		✓	✓	✓		✓	
T_INTERN.02		✓							
T_INTERN.03		✓					✓		✓
T_INTERN.04		✓							
T_INTERN.05	✓								
T_INTERN.06	✓	✓	✓			✓			
T_INTERN.07				✓		✓	✓		
T_EXTERN.01					✓				
T_EXTERN.02			✓		✓	✓			
P_PRODUCT.01			✓		✓				
P_PRODUCT.02	✓	✓			✓	✓			
P_IDENT								✓	

Table 6 : Cross reference : Threats/Policies - Security objectives for the TOE

	O_ENV.01	O_ENV.02	O_ENV.03	O_ENV.04	O_ENV.05	O_ENV.06
A_RESP	✓		✓			
A_PRIVATE		✓				
A_APPLI.01		✓				
A_APPLI.02			✓			
A_DESIGN.01			✓	✓		
A_DESIGN.02			✓		✓	
A_MANUF.01			✓	✓		
A_MANUF.02			✓		✓	✓
A_MANUF.03			✓			
T_INTERN.01				✓	✓	✓
T_INTERN.02				✓		
T_INTERN.03				✓		
T_INTERN.04					✓	✓
T_INTERN.05					✓	✓
T_INTERN.06						
T_INTERN.07						
T_EXTERN.01						
T_EXTERN.02			✓	✓		
P_PRODUCT.01						
P_PRODUCT.02						
P_IDENT						

Table 7 : Cross reference : Assumptions/Threats/Policies - Security objectives for the environment

7.3 Security requirements rationale

7.3.1 Security functional requirement rationale

Ref.	Security objectives	Security requirements	Written evidence
01	O_TOE.01	**FCS_CKM.2,**	Cryptographic key distribution in accordance with standard based algorithms and key sizes guarantees that management of cryptographic keys protects their confidentiality.
		FMT_MTD.1, FMT_SMR.1,	Management of TSF data and security roles ensure protection of keys during their management.
		FIA_UID.2,	User identification before any action guarantees a control over actions related with cryptographic keys.
		FPT_ITT.1,	Basic internal TSF data transfer protection, guarantees that cryptographic keys are protected when transmitted between separate parts of the TOE.
		FPT_PHP.3	Resistance to physical attacks contribute to protect confidentiality of cryptographic keys.
02	O_TOE.02	**FCS_CKM.2,**	Cryptographic key distribution in accordance with standard based algorithms and key sizes guarantees that management of cryptographic keys protects their integrity.
		FMT_MTD.1, FMT_SMR.1,	Management of TSF data and security roles ensure protection of cryptographic resources during their management.
		FIA_UID.2,	User identification before any action guarantees a control over actions related with cryptographic keys.
		FPT_ITT.1,	Basic internal TSF data transfer contributes to protect against alteration of cryptographic resources.
		FPT_PHP.3	Resistance to physical attacks contribute to protect integrity of cryptographic resources and keys during their management and use.
03	O_TOE.03	**FPT_ITT.1,**	Protection of basic internal TSF data transfer means protection of connections between processing unit and memories.
		FPT_PHP.3	Resistance to physical attacks contribute to protect internal connections within the TOE.
04	O_TOE.04	**FIA_AFL.1,**	Basic data authentication achieves firmware authentication.
		FCS_COP.1,	Cryptographic operations contribute to provide authentication.
		FCS_CKM.1,	Generation of cryptographic keys in accordance with a specified standard-based cryptographic key generation

			algorithm and specified cryptographic key sizes provides with a guaranty on the security of the firmware authentication mechanism.
		FCS_CKM.2,	Distribution of cryptographic keys in accordance with a standard-based key distribution algorithm provides with a guaranty on the confidentiality of the firmware authentication key.
		FCS_CKM.4,	Destruction of cryptographic keys in accordance with a standard-based key destruction algorithm provides with a guaranty on the non-reusing of the firmware authentication key (used for the signature of the firmware).
		FIA_UAU.2	Authentication of the users guarantees that the user identity could not be usurped. Any attempt of downloading a fake firmware or application will be detected during the verification of its signature. The signature of the firmware/application authenticates directly the firmware designer/manufacturer or the application designer.
05	O_TOE.05	**FCS_COP.1,**	Cryptographic operations contribute to provide authentication.
		FCS_CKM.1,	Generation of cryptographic keys in accordance with a specified standard-based cryptographic key generation algorithm and specified cryptographic key sizes provides with a guaranty on the authentication of the applications and the confidentiality of the associated data and keys.
		FCS_CKM.2,	Distribution of cryptographic keys in accordance with a standard-based key distribution algorithm provides with a guaranty on the authentication of the applications and the confidentiality of the associated data and keys.
		FCS_CKM.4,	Destruction of cryptographic keys in accordance with a standard-based key destruction algorithm provides with a guaranty on the non-reusing of the firmware authentication key.
		FDP_DAU.1,	Basic Data Authentication provides with a guarantee of authenticity of the application software, data and keys running onto the TOE.
		FIA_UID.2,	User identification before any action guarantees a control over actions which may have an impact on the application software which is running onto the TOE and related data and keys.
		FPT_PHP.3	Resistance to physical attacks contribute to protect application software, data and keys within the TOE.
06	O_TOE.06	**FPT_PHP.3**	Resistance to physical attacks contribute to protect the TOE.

07	O_TOE.07	**FMT_MOF.1,** **FPT_FLS.1,** **FPT_PHP.3,** **FPT_RCV.2**	Management of security functions behaviour requires that a mechanism is implemented in order to guarantee the continued correct operation of the TOE. Preservation of a secure state in case of failures ensures as a result the continued correct operations of the TOE security functions. Resistance to physical attacks protects the behaviour of the TOE security functions and provides as a consequence their continued correct operation. Automated recovery provides with a means of rebuilding, in case of failure, the original state before service discontinuity. This allows to restart security functions operations and obtain a correct behaviour.
08	O_TOE.08	**FCS_COP.1**	Evidence of distinguish identity is provided using cryptographic operations. It consists for the almost in managing cryptographic keys used for the generation of evidence (signature mechanism).
09	O_TOE.09	**FPT_FLS.1,** **FPT_RCV.2,** **FPT_TST.1**	Failure with preservation of secure state, automated recovery and TSF testing ensure the TOE protection against internal technical failures.

Table 8 : Matching security objectives - security requirements

	O_TOE.01	O_TOE.02	O_TOE.03	O_TOE.04	O_TOE.05	O_TOE.06	O_TOE.07	O_TOE.08	O_TOE.09
FCS_CKM.1				✓	✓				
FCS_CKM.2	✓	✓		✓	✓				
FCS_CKM.4				✓	✓				
FCS_COP.1				✓	✓			✓	
FDP_DAU.1					✓				
FIA_AFL.1				✓					
FIA_UAU.2				✓					
FIA_UID.2	✓	✓			✓				
FMT_MOF.1							✓		
FMT_MTD.1	✓	✓							
FMT_SMR.1	✓	✓							
FPT_FLS.1							✓		✓
FPT_ITT.1	✓	✓	✓						
FPT_PHP.3	✓	✓	✓		✓	✓	✓		
FPT_RCV.2							✓		✓
FPT_TST.1									✓

Table 9 : Cross reference : security objectives - security requirements

7.3.2 Security functional requirement dependencies

This section demonstrates that the dependencies between security requirements components included in this PP are satisfied.

Ref.	Component	Dependency	Line reference
1	FCS_CKM.1	[FCS_CKM2 or FCS_COP.1], FCS_CKM.4, **FMT_MSA.2**	2 or 4,3
2	FCS_CKM.2	[FDP_ITC.1 or FCS_CKM.1], FCS_CKM.4, **FMT_MSA.2**	1,3
3	FCS_CKM.4	[FDP_ITC.1 or FCS_CKM.1], **FMT_MSA.2**	1
4	FCS_COP.1	[FDP_ITC.1 or FCS_CKM.1], FCS_CKM.4, **FMT_MSA.2**	1,3
5	FDP_DAU.1	None	
6	FIA_AFL.1	FIA_UAU.1	Role played by FIA_UAU.2
7	FIA_UAU.2	**FIA_UID.1**	
8	FIA_UID.2	None	
9	FMT_MOF.1	FMT_SMR.1	11
10	FMT_MTD.1	FMT_SMR.1	11
11	FMT_SMR.1	FIA_UID.1	Role played by FIA_UID.2
12	FPT_FLS.1	ADV_SPM.1	Included in EAL4
13	FPT_ITT.1	None	
14	FPT_PHP.3	None	
15	FPT_RCV.2	FPT_TST.1, AGD_ADM.1, ADV_SPM.1	16, included in EAL4
16	FPT_TST.1	**FPT_AMT.1**	

Table 10 : Functional dependencies analysis

Rationale for exclusion of the following component dependencies :

FMT_MSA.2

In this PP the user is an application loaded in the TOE. This application is signed (static signature) by the issuer of the application. This PP considers that the signature is not a security attribute that has to be handled. Consequently the requirements related to Management of security attribute, consistency are considered out of scope.

FPT_AMT.1

In this PP the abstract machine testing is not appropriate. The transactional smartcard reader does not rely on an abstract machine.

FIA_UAU.2

In this PP the user, which is an application or a downloadable firmware, is loaded in the TOE. Before its downloading the user is signed (static signature). The verification of the signature by the firmware during the downloading process corresponds to an authentication but does not require any identification action. The signature verification is self-sufficient.

7.3.3 Strength of function level rationale

Due to the definition of the TOE, it is very important that the claimed SOF should be SOF-medium since the product critical mechanisms have to resist to attackers possessing a moderate attack potential and to be only defeated by attacker possessing a high attack potential.

7.3.4 Security assurance requirements rationale

The assurance requirements of this PP are summarised in the following table :

Ref.	Requirement	Name	Type
1	EAL4	Methodically designed, tested and reviewed	Assurance level
2	ADV_IMP.2	Implementation of the TSF	Higher hierarchical component
3	AVA_VLA.3	Moderately resistant	Higher hierarchical component

Table 11 : Assurance requirements

Evaluation Assurance level rationale

An assurance requirement of EAL4 is required for this type of TOE since it is intended to defend against sophisticated attacks. This evaluation assurance level was selected since it is designed to permit a developer to gain maximum assurance from positive security engineering based on good commercial practices. EAL4 represents the highest practical level of assurance expected for a commercial grade product.

The assurance level EAL4 is achievable, since it requires no specialist techniques on the part of the developer.

Assurance augmentations rationale

Additional assurance requirement are also required due to the definition of the TOE and to the conformance to the ITSEC evaluation level E3 with a strength of mechanism medium.

ADV_IMP.2 Implementation of the TSF

The implementation representation is used to express the notion of the least abstract representation of the TSF, specifically the one that is used to create the TSF itself without further design refinement. This assurance component is a higher hierarchical component to EAL4 (only ADV_IMP.1 is found in EAL4). It is important for a transactional smartcard reader that the evaluator evaluates

the implementation representation of the entire TSF and determines if the functional requirements in the Security Target are addressed by the representation of the TSF.

ADV_IMP.2 has dependencies with ADV_LLD.1 « descriptive Low-Level Design », ADV_RCR.1 « Informal correspondence demonstration », ALC_TAT.1 « Well defined development tools ». These components are included in EAL4, then these dependencies are satisfied.

AVA_VLA.3 Moderately resistant

Due to the definition of the TOE it must be shown to be moderately resistant to penetration attacks. This is due to the fact that a transactional smartcard reader can be placed in an hostile environment. This assurance requirement is achieved by the AVA_VLA.3 component. Independent vulnerability analysis is based on moderate 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 moderate level of technical sophistication.

AVA_VLA.3 has dependencies with ADV_FSP.1 « Informal functional specification », ADV_HLD.2 « Security enforcing high-level design », ADV_IMP.1 « Subset of the implementation of the TSF », ADV_LLD.1 « Descriptive low-level design », AGD_ADM.1 « Administrator guidance », AGD_USR.1 « User guidance ». These components are included in EAL4, then these dependencies are satisfied.

7.3.5 Security requirements are mutually supportive and internally consistent

The purpose of this part of the PP rationale is to show that the security requirements are mutually supportive and internally consistent.

No detailed analysis is given in respect to the security requirements because :

- EAL4 is an established set of mutually supportive and internally consistent assurance requirements.
- The dependencies analysis for the additional assurance components in the previous section has shown that the assurance requirements are mutually supportive and internally consistent (all the dependencies have been satisfied).
- The dependencies analysis for the functional requirements described above demonstrates mutual support and internal consistency between the functional requirements.
- Inconsistency between functional and assurance requirement can only arise if there are functional-assurance dependencies which are not met, a possibility which has been shown not to arise by the following section.

Therefore, the dependencies analysis described above demonstrates mutual support and internal consistency between the functional requirements.