Common Criteria for IT Security Evaluation
Protection Profile

Smartcard Embedded Software
Protection Profile

November 19th, 1998

Registered at the French Certification Body under the number PP/9810

TABLE OF CONTENTS

1. PP introduction

1.1 PP Identification

Title :	Smartcard Embedded Software Protection Profile
Version number	V1.2 issued November 19th 1998
Registration :	
Origin :	Schlumberger

A glossary of terms used in this PP is given in annex A.

This PP is designed to be have the same security level as the " Smartcard Integrated Circuit Protection Profile " version V2.0 registered under reference PP/9806, september 1998..
The definition of the Smartcard life phases are taken from this IC PP.
An Integrated Circuit which meets PP/9806 will fulfill the objectives on the IC development phase.

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

1.2 PP overview

1.2.1 Context

The increase in the number and complexity of applications in the smartcard market is reflected in the increase of the level of data security required. The security needs for a smartcard can be summarized as being able to counter those who want to defraud, gain unauthorized access to data and control a system using a smartcard. Therefore it is mandatory to:

- maintain the integrity and the confidentiality of the content of the smartcard non-volatile memory (program and data memories).
- maintain the integrity and the confidentiality of the security enforcing and security relevant components (security mechanisms and associated functions) of the embedded software.

The assets to be protected are in general secret data as Personal Identification Numbers, Balance Value (Stored Value Cards), and Personal Data Files. Another set of assets is the Access Rights ; these include any cryptographic algorithms and keys needed for accessing and using the services provided by the system through the use of the smartcard.

The intended environment is very large and generally, once issued, the smartcard can be stored and used anywhere in the world at any time and no control can be applied to the smartcard and its associated end-user with the exception of those that are applicable when the smartcard comes to its end usage in the system in conformance with its specifications.

One of the key market drivers for smartcard is standardization of specifications such as the EMV specifications (Europay-Mastercard-Visa) for banking applications, the current revision of ETSI prN and GSM 11 which both include parts of the ISO 7816, and the specifications SET or C-SET for electronic commerce. Due to market demands, the major cryptographic schemes such as those using DES, RSA, DSA, are also now included in standard specifications.

1.2.2 PP and related TOE

The intention of this Protection Profile is to specify functional and assurance requirements applicable to smartcard embedded software. This PP covers software designed for major smartcard applications, typically :

- banking and finance market for credit/debit cards, electronic purse (stored value cards) and electronic commerce,
- network based transaction processing such as mobile phones (GSM SIM cards), pay-TV (subscriber and pay-per-view cards), communication highways (Internet access and transaction processing),
- transport and ticketing market (access control cards),
- governmental cards (ID-cards, healthcards, driver license etc....),
- multimedia commerce and Intellectual Property Rights protection.

This PP deals with the specific requirements of the Basic Software (BS) and the Application Software (AS) and both of their implementation on the Integrated Circuit (IC).

1.2.3 PP aim and contents

The main objectives of this Protection Profile are :

- to describe the Target of Evaluation (TOE) and position it in the smartcard product 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 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 its associated documentation during the development and production phases,
- to specify the security requirements which include the TOE IT functional requirements, the TOE IT Assurance requirements and the security requirements for the IT environment,
- the Evaluation Assurance Level for this PP is EAL4 augmented.

2. TOE Description

This part of the PP describes the TOE as an aid to the understanding of its security requirements and addresses the product type, the smartcard product life cycle, the TOE environment along the smartcard life cycle and the general IT features of the TOE.

2.1 Product type

The Target of Evaluation (TOE) is the embedded software (ES) and the associated embedded data of a smartcard working on a microcontroller unit in accordance with the functional specifications.
The microcontroller unit is outside the scope of the TOE, but is part of its environment.

Generally, a smartcard product may include other elements (such as specific hardware components, batteries, capacitors, antennae, holograms, magnetic stripes, security printing...) but these are outside the scope of this Protection Profile.

The typical TOE is composed of an operating system, several application software and some initialization data and process.

This PP addresses requirements upon the Basic Software (BS) and the Application Software (AP) embedded in the Integrated Circuit.

2.2 Smartcard Product Life-cycle

The smartcard product life-cycle is decomposed into 7 phases where the following authorities are involved:

Phase 1	Smartcard software development	**the smartcard embedded software developer** is in charge of the smartcard embedded software development and the specification of pre-personalization requirements,
Phase 2	IC Development	**the IC designer** designs the integrated circuit, develops IC firmware if applicable, provides information, software or tools to the smartcard software developer, and receives the software from the developer, through **trusted delivery and verification procedures**. From the IC design, IC firmware and smartcard embedded software, he constructs the smartcard IC database, necessary for the IC photomask fabrication.
Phase 3	IC manufacturing and testing	**the IC manufacturer** is responsible for producing the IC through three main steps : IC manufacturing, testing, and pre-personalization.
Phase 4	IC packaging and testing	**the IC packaging manufacturer** is responsible for the IC packaging and testing,
Phase 5	Smartcard product finishing process	**the smartcard product manufacturer** is responsible for the smartcard product finishing process and testing,
Phase 6	Smartcard personalization	**the personalizer** is responsible for the smartcard personalization and final tests. Other application software may be loaded onto the chip during the personalization process.
Phase 7	Smartcard end-usage	**the smartcard issuer** is responsible for the smartcard product delivery to **the smartcard end-user**, and for the end of life process.

.

The Target of Evaluation (TOE) is a functional software designed during phase 1, and embedded in an integrated circuit (IC) during phases 2 and 3 ; considering that the only purpose of the Embedded Software is to control and protect the operation of the Smartcard during phases 4 to 7 (operational phases).The global security requirements of the TOE mandate to consider, during the development phase, the threats to security occurring in the other phases. This is why this PP addresses the functions used in phases 4 to 7 but developed during phase 1.

The limits of the TOE correspond to phase 1 including the software and corresponding data delivery to the IC manufacturer , and to the embedded software working on the IC as delivered by the IC manufacturer at the end of phase 3.

These different phases may be performed at different sites. This implies that procedures on the delivery process of the TOE must exist and be applied for every delivery within a phase or between phases. This includes any kind of delivery performed from phase 1 to phase 7, including:

- intermediate delivery of the TOE or the TOE under construction within a phase,
- delivery of the TOE or the TOE under construction from one phase to the next.

These procedures must be compliant with the secure usage assumptions [A_DLV] developed in section 3.

2.3 TOE environment

Considering the TOE, three types of environment are defined :

- Development environment corresponding to phase 1,
- Integration environment corresponding to the construction of the IC database and IC photomask photomask (phase 2), integration of the embedded software into the IC and personalization of the smartcard with the user data during manifacturing (phase 3),
- User environment, from phase 4 to phase 7.

2.3.1 Development Environment

In order to ensure security, the environment in which the development takes place must be made secured with controllable accesses having traceability. Furthermore, it is important that all authorized personnel feels involved and fully understands the importance and the rigid implementation of the defined security procedures.

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

Design and development of the ES then follows. The engineer uses a secure computer system (preventing unauthorized access) to make his conception, design, implementation and test performances.

Storage of sensitive documents, databases on tapes, diskettes, and printed circuit layout information are in appropriately locked cupboards/safe. Of paramount importance also is the disposal of unwanted data (complete electronic erasures) and documents (e.g. shredding).

Testing, programming and deliveries of the TOE then take place. When these are done offsite, they must be transported and worked on in a secure environment with accountability and traceability of all (good and bad) products.
During the electronic transfer of sensitive data, procedures must be established to ensure that the data arrives, only at the destination and is not accessible at intermediate stages (e.g. stored on a buffer server where system administrators make backup copies). It must also be ensured that transfer is done without modification or alteration.

2.3.2 Integration environment

During the phase, the TOE is stored and transported to be integrated into the IC by the way of the IC database, or the personalization process. All the persons involved in such an operation should fully understand the importance of the defined security procedures.

Moreover, the environment in which these operations take place must be secured.

Storage of sensitive documents, databases on tapes, diskettes, information are in appropriately locked cupboards/safe. Of paramount importance also is the disposal of unwanted data (complete electronic erasures) and documents (e.g. shredding).

2.3.3 User environment

As high volumes of smartcards are commonly produced, adequate control procedures are necessary to account for all products at all stages of production.
They must be transported and worked on in a secure environment with accountability and traceability of all (good and bad) products.

Smartcards are used in a wide range of applications to ensure authorized conditional access. Examples of such are Pay-TV, Banking Cards, Portable communication SIM cards, Health cards, Transportation cards.

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

2.4 TOE logical phases

During its construction and usage, the TOE may be under several life cycle logical phases. These phases are sorted under a logical controlled sequence. The change from one phase to the next is under the TOE control.

During the phases 1, 2, 3, the TOE is being developed and produced. The **administrators** are the following:
- the smartcard embedded software developer,
- the IC designer,
- the IC manufacturer.

During phases 4 to 7, the users of the TOE are the following

Phase 4	- the packaging manufacturer, (**administrator**) - the smartcard embedded software developer, - the system integrators such as the terminal software developer.
Phase 5	- the smartcard product manufacturer, (**administrator**) - the smartcard embedded software developer, - the system integrators such as the terminal software developer.
Phase 6	- the personalizer (**administrator**), - the smartcard issuer (**administrator**), - the smartcard embedded software developer, - the system integrators such as the terminal software developer.
Phase 7	- the smartcard issuer **(administrator**), - the smartcard end-user, - the smartcard embedded software developer, - the system integrators such as the terminal software developer. The IC manufacturer, the smartcard product manufacturer and the software developer may also receive smartcards for analysis if problems occur during the smartcard usage.

2.5 General IT features of the TOE

The TOE IT functionality consist of data storage and processing such as:
- arithmetical functions (e.g. incrementing counters in electronic purses, calculating currency conversion in electronic purses...),
- data communication,
- cryptographic operations (e.g. data encryption, digital signature computation/verification).

3. TOE 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 envisaged threats, the organizational security policies and the assumptions made upon the TOE intended environment.

The assets to be protected are :

- the specification, design, development tools and technology, (both for software and IC hardware),
- the Dedicated Software,
- the Basic Software (including operating system programs and documentation),
- the Application Software,
- the application data of the TOE (such as initialization and personalization requirements).

These assets have to be protected in terms of confidentiality and integrity.

3.1 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.

Threats have to be split into :

- threats which can be countered by the TOE (class I),
- threats which can be countered by the TOE environment (class II).

3.1.1 Threats on all phases (1 to 7)

The threat agents are very general and are case dependent.

- During phase 1 to 3, developers are the most apt to mount the threats.
- During phases 1 to 3, external persons can spy on communications or steal the TOE so to attack it. They have less capabilities than the developers, but they cannot be screened out.
- For phases 4 to 6, the main potential threat agents are personnel allowed to manipulate the TOE or personalization data, but external parties can also be active.
- During phase 7, the administrator, issuer, or at least it’s agents, can in some cases be considered a threat agent.
- During phase 7, in some cases, such as electronic purses, the card holder can be interested in breaking the TOE.
- During phases 3 to 7, threats coming from outsiders must be preceded by the stealing of the TOE.

T.CLON Functional cloning of the TOE (full or partial) appears to be relevant to any phase of the TOE life-cycle, from phase 1 to phase 7.

Generally, this threat is derived from specific threats combining unauthorized disclosure, modification or theft of assets at different phases.

T.DIS Unauthorized disclosure of the smartcard embedded software, data or any related information.

T.MOD Unauthorized modification of the smartcard embedded software and data.

3.1.2 Threats on phase 1

During phase 1, two types of threats have to be considered:

a) threats on the smartcard embedded software and its development environment,

b) threats on software development tools coming from the IC manufacturer.

The main threat agents are developers, but they can also be other parties working in the same company or outside.

T.T_TOOLS Theft or unauthorized use of the smartcard embedded software development tools (such as PC, databases,...).

T.FLAW Introduction of flaws in the TOE due to malicious intents or insufficient development.

T.T_SAMPLE Theft or unauthorized use of integrated circuit samples containing the embedded software (e. g. bound out, dil,...).

T.MOD_INFO Unauthorized modification. of any information (technical or detailed specifications, implementation code, design technology, tools characteristics) used for developing software or loading data.

T.DIS_TEST Unauthorized disclosure of the smartcard embedded software test information including interpretations.

T.DIS_INFO Unauthorized disclosure of any information (technical or detailed specifications, implementation code, design technology, tools characteristics) used for developing software or loading data. This includes sensitive information on IC specification, design and technology, software and tools.

3.1.3 Threats on delivery of software and related information from phases 1 and 2 to phases 2, 3 and 6

These threats address

- software to be embedded send by the software developer to the IC designer (for designing the photomask) : phase 1 to phase 2,
- Transformed software send from the IC developer to the IC manufacturer : phase 2 to phase 3,
- prepersonalization data send by software developer to IC manufacturer for prepersonalization, phase 3 : phase 1 to phase 3,
- personalization data send by software developer to the personalizer, phase 1 to phase 6.

Data send directly from smartcard issuer to the IC manufacturer and to personalizer are considered as belonging respectively to phase 3 and phase 6.
The main threats agents are eavesdroppers on networks or on other delivery processes.

T.T_DEL Theft or unauthorized use of the smartcard embedded software and any additional application data delivered to the IC designer, IC manufacturer or to the personalizer.

T.MOD_DEL Unauthorized modification of the smartcard embedded software and any additional application data delivered to the IC designer, IC manufacturer or to the personalizer.

T.DIS_DEL Unauthorized disclosure of the smartcard embedded software and any additional application data delivered to the IC designer, IC manufacturer or to the personalizer.

3.1.4 Threats on phase 2

The main threat agents are persons working inside the IC designing plant or persons breaking in.

T.DIS_TEST Unauthorized disclosure of the smartcard embedded software test information including interpretations.

T.DESIGN_IC Poor IC design leading to IC security mechanisms not meeting state of the art level.

3.1.5 Threats on phases 3 to 6

The main threats agents are persons working inside the plants or working for the agents responsible for transportation between plants.

T.T_PRODUCT Theft or unauthorized use of the smartcard product or any related information. For example unauthorized use of the embedded software application functions.

T.DIS_TEST Unauthorized disclosure of the smartcard embedded software test information including interpretations.

3.1.6 Threat on phase 7

The threat can come from outside parties who first steal the smartcard product. The realisation of the threat is a first step toward breaking open the product.

T.T_PRODUCT Theft or unauthorized use of the smartcard product or any related information. For example unauthorized use of the embedded software application functions.

The table given below indicates the relationship between the smartcard life-cycle phases, the threats and the type of the threats.

Threats	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase 6	Phase 7
Functional cloning							
T.CLON	Class II	Class II	Class I/II	Class I/II	Class I/II	Class I/II	Class I/II
Unauthorized disclosure of assets							
T.DIS	Class II	Class II	Class I/II	Class I	Class I	Class I/II	Class I
T.DIS_INFO	Class II						
T.DIS_DEL	Class II	Class II	Class II			Class II	
T.DIS_TEST	Class II	Class II	Class II	Class II	Class II	Class II	
Theft of assets							
T.T_TOOLS	Class II						
T.T_SAMPLE	Class II						
T.T_DEL	Class II	Class II	Class II			Class II	
T.T_PRODUCT			Class I/II	Class I/II	Class I/II	Class I/II	Class I/II
Unauthorized modification or faulty development of assets							
T.FLAW	Class II						
T_DESIGN_IC		ClassII					
T.MOD	Class II	Class II	Class I/II	Class I	Class I	Class I/II	Class I
T.MOD_INFO	Class II						
T.MOD_DEL	Class II	Class II	Class II			Class II	

Table 3.1 : Threats during phases

3.2 Organizational Security policies

It can be considered as a good policy to define an organizational security policy. As this policy is application dependent, no security policy has been defined within the scope of this PP. It is up to the Security Target writer to define the security policy which is to be applied by the TOE.

3.3 Assumptions

This section concerns assumptions about security aspects of the environment in which the TOE is intended to be used.
Assumptions described hereafter have to be considered for a secure system using smartcard products:

- assumptions on the TOE delivery process from phase to phase,
- assumptions on IC development,
- assumptions on phases 2 to 7.

3.3.1 Assumptions on the TOE delivery process from phase to phase

A.DLV_CONTROL procedures must guarantee the control of the TOE delivery and storage process and conformance to its objectives as described in the following secure usage assumptions. Secure storage and handling procedures are applicable for all TOE's parts (programs, data, documents,...).

A.DLV_CONF procedures must also prevent if applicable any non-conformance to the confidentiality convention and must have a corrective action system in case any non-conformance or misprocessed procedures are identified.

A.DLV_PROTECT procedures shall ensure protection of material/information under delivery including the following objectives:

- non-disclosure of any security relevant information,
- identification of the elements under delivery,
- meeting confidentiality rules (confidentiality level, transmittal form, reception acknowledgment), physical protection to prevent external damage.

A.DLV_TRANS procedures shall ensure that material/information is delivered to the correct party.

A.DLV_TRACE procedures shall ensure traceability of delivery including the following parameters:

- origin and shipment details,
- reception, reception acknowledgment,
- location material/information.

A.DLV_AUDIT procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process and highlight all non-conformance to this process.

A.DLV_RESP procedures shall ensure that people dealing with the procedures for delivery have got the required skill, training and knowledge to meet the procedure requirements and to act to be fully in accordance with the above expectations.

3.3.2 Assumptions on IC development (phase 2)

There are two types of assumptions : the assumptions on the development of the TOE and the assumptions on the personnel aspects.

secure development :

A.IC_PRODUCT the Smartcard integrated circuit is designed and built using state of art technology with the aim of achieving security objectives.

secure personnel assumptions

A.IC_ORG procedures dealing with physical, personnel, organizational, technical measures for the confidentiality and integrity of smartcard embedded software and data (e.g. source code and any associated documents) shall exist and be applied in the smartcard IC database construction.

3.3.3 Assumptions on phases 3 to 6

A.USE_TEST it is assumed that appropriate functionality testing of the smartcard functions is used in phases 3 to 6.

A.USE_PROD it is assumed that security procedures are used during all manufacturing and test operations through smartcard production phases to maintain the confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).

3.3.4 Assumption on phase 7

A.USE_SYS it is assumed that the security of sensitive data stored/handled by the system (terminals, communications ...) is maintained.

4. Security objectives

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

- integrity and confidentiality of assets,
- protection of the TOE and associated documentation during development and production phases.

4.1 Security objectives for the TOE

The TOE shall use state of art technology to achieve the following TOE security objectives

O.INTEGRITY	The TOE must provide the means of detecting loss of integrity affecting security information stored in memories.
O.TAMPER	The TOE must prevent tampering with it's security functions
O.FUNCTION	The TOE must provide protection against unauthorized use of it's software application functions.
O.CLON	The TOE functionality needs to be protected from cloning.
O.OPERATE	The TOE must ensure the continued correct operation of its security functions.
O.DIS_MECHANISM	The TOE shall ensure that the software security mechanisms are protected against unauthorized disclosure.
O.DIS_MEMORY	The TOE shall ensure that the embedded software does not allow unauthorized access to information stored in memories.
O.MOD_MEMORY	The TOE shall ensure that the embedded software does not allow unauthorized modification or corruption of the information stored in memories.
O.FLAW	The TOE must not contain flaws in design, data values or implementation.

4.2 Security objectives for the environment

4.2.1 Objectives on phase 1 (development phase)

O.SOFT_ACS The embedded software shall be accessible only by authorized personnel (physical, personnel, organizational, technical procedures).

O.MECH_ACS Details of software security mechanisms shall be accessible only by authorized personnel.

O.TI_ACS Security relevant technology information shall be accessible only by authorized personnel. This information includes software test information including the interpretations of the test results.

O.INIT_ACS Application data shall be accessible only by authorized personnel (physical, personnel, organizational, technical procedures).

O.TOOLS_ACS Embedded software development tools shall be accessible only by authorized personnel.

O.SAMPLE_ACS Samples used to run test shall be accessible only by authorized personnel.

NOTE : These security objectives are designed to correspond to A.SOFT_ARCHI from IC PP.

4.2.2 Objective on phase 2

O.MECH_IC The IC shall be designed using state of art technology focusing on :

- preventing physical tempering with its security critical parts,
- protection from cloning,
- ensuring correct operation of its security functions ,
- not containing flaws in design, implementation or operation,
- protecting stored memory from unauthorized disclosure,
- protection of sensitive stored information against any corruption or unauthorized modification.

4.2.3 Objectives on phases 2, 3 and 6 and on delivery to these phases.

O.DIS_DEV The IC designer and the personalizer must have procedures to control the sales, distribution, storage and usage of the software and classified documentation, suitable to maintain the integrity and the confidentiality of the assets of the TOE.
It must be ensured that tools are only delivered to the parties' authorized personnel.
It must be ensured that confidential information on defined assets are only delivered to the parties' authorized personnel.

O.SOFT_DLV The embedded software must be delivered from the smartcard software developer to the IC designer through a trusted delivery and verification procedure that shall be able to maintain the integrity of the software and its confidentiality. The same goes for the delivery of the personalization data from the product manufacturer to the personalizer.

4.2.4 Objectives on phase 2 to 7

O.PRODUCT_DEV The IC designer, manufacturer, personalizer and issuer must have procedures to control the sales, distribution, storage and usage of the product, suitable to maintain the integrity and the confidentiality of the assets of the TOE.. This applies also to test information whenever it is pertinent.
It must be ensured that the product is only delivered to the parties' authorized personnel and authorized end users.

5. TOE security functional requirements

The TOE security functional requirements define the functional requirements for the TOE using only functional requirements components drawn from the CC part 2.

The permitted operations such as assignment, selection, refinement and iteration will have to be completed in a Security Target, compliant with this PP.

5.1 FAU Security audit

5.1.1 FAU_ARP Security audit automatic response

FAU_ARP.1 Security alarms

Hierarchical to: No other components.

FAU_ARP.1.1 **The TSF shall take [assignment: *list of the least disruptive actions*] upon detection of a potential security violation.**

Dependencies: **FAU_SAA.1 Potential violation analysis**

5.1.2 FAU_SAA Security audit analysis

FAU_SAA.1 Potential violation analysis

Hierarchical to: No other components.

FAU_SAA.1.1 **The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.**

FAU_SAA.1.2 **The TSF shall enforce the following rules for monitoring audited events:**

a) Accumulation or combination of [assignment: *subset of defined auditable events*] known to indicate a potential security violation;

b) [assignment: *any other rules*].

Dependencies: **FAU_GEN.1 Audit data generation**

This dependency is not met by the PP (see the Rationale for details).

5.2 FCS Cryptographic support

5.2.1 FCS_CKM Cryptographic Key Management

FCS_CKM.3 Cryptographic key access

Hierarchical to: No other components.

FCS_CKM.3.1 **The TSF shall perform [assignment: *type of cryptographic key access*] in accordance with a specified cryptographic key access method [assignment: *cryptographic key access method*] that meets the following: [assignment: *list of standards*].**

Dependencies: **[FDP_ITC.1 Import of user data without security attributes**

or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

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

Dependencies: **[FDP_ITC.1 Import of user data without security attributes**

or

FCS_CKM.1 Cryptographic key generation]

FMT_MSA.2 Secure security attributes

5.2.2 FCS_COP Cryptographic Operation

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

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*].**

Dependencies: **[FDP_ITC.1 Import of user data without security attributes**

or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

5.3 FDP : User data protection

5.3.1 FDP_ACC Access Control Policy

FDP_ACC.2 Complete access control

Hierarchical to: FDP_ACC.1

FDP_ACC.2.1 The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects and objects*] **and all operations among subjects and objects covered by the SFP.**

FDP_ACC.2.2 **The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.**

Dependencies: FDP_ACF.1 Security attribute based access control

5.3.2 FDP_ACF Access Control Functions

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 **The TSF shall enforce the [assignment: *access control SFP*] to objects based on [assignment: *security attributes, named groups of 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 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].**

Dependencies: **FDP_ACC.1 Subset access control**

FMT_MSA.3 Static attribute initialisation

5.3.3 FDP_DAU Data Authentication

FDP_DAU.1 Basic data authentication

Hierarchical to: No other components.

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.**

Dependencies: No dependencies.

5.3.4 FDP_ETC Export to Outside TSF Control

FDP_ETC.1 Export of user data without security attributes

Hierarchical to: No other components.

FDP_ETC.1.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when exporting user data, controlled under the SFP(s), outside of the TSC.**

FDP_ETC.1.2 **The TSF shall export the user data without the user data's associated security attributes.**

Dependencies: **[FDP_ACC.1 Subset access control, or**

FDP_IFC.1 Subset information flow control]

5.3.5 FDP_ITC Import from Outside TSF Control

FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components.

FDP_ITC.1.1 **The TSF shall enforce the [assignment: *access control SFP and/or information flow control SFP*] when importing user data, controlled under the SFP, from outside of the TSC.**

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

FDP_ITC.1.3 **The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [assignment: *additional importation control rules*].**

Dependencies: [**FDP_ACC.1 Subset access control**, **or**

FDP_IFC.1 Subset information flow control]

FMT_MSA.3 Static attribute initialisation

5.3.6 FDP_RIP Residual Information Protection

FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

FDP_RIP.1.1 **The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to*, *deallocation of the resource from*] the following objects: [assignment: *list of objects*].**

Dependencies: No dependencies.

5.3.7 FDP_SDI Stored Data Integrity

FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to: FDP_SDI.1

FDP_SDI.2.1 The TSF shall monitor user data stored within the TSC for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*].

FDP_SDI.2.2 **Upon detection of a data integrity error, the TSF shall [assignment: *action to be taken*].**

Dependencies: No dependencies.

5.4 FIA : Identification and authentication

5.4.1 FIA_AFL Authentication Failures

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

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

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

Dependencies: **FIA_UAU.1 Timing of authentication**

5.4.2 FIA_ATD User Attribute Definition

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1 **The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: *list of security attributes*].**

Dependencies: No dependencies.

5.4.3 FIA_UAU User Authentication

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

FIA_UAU.1.1 **The TSF shall allow [assignment: *list of TSF mediated actions*] on behalf of the user to be performed before the user is authenticated.**

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

Dependencies: **FIA_UID.1 Timing of identification**

FIA_UAU.3 Unforgeable authentication

Hierarchical to: No other components.

FIA_UAU.3.1 **The TSF shall [selection: *detect, prevent*] use of authentication data that has been forged by any user of the TSF.**

FIA_UAU.3.2 **The TSF shall [selection: *detect, prevent*] use of authentication data that has been copied from any other user of the TSF.**

Dependencies: No dependencies.

FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to: No other components.

FIA_UAU.4.1 **The TSF shall prevent reuse of authentication data related to [assignment: *identified authentication mechanism(*s)].**

Dependencies: No dependencies.

5.4.4 FIA_UID User Identification

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

FIA_UID.1.1 **The TSF shall allow [assignment: *list of TSF-mediated actions*] on behalf of the user to be performed before the user is identified.**

FIA_UID.1.2 **The TSF shall require each user to be succesfully identified before allowing any other TSF-mediated actions on behalf of that user.**

Dependencies: No dependencies.

5.4.5 FIA_USB User-Subject Binding

FIA_USB.1 User-subject binding

Hierarchical to: No other components.

FIA_USB.1.1 **The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user.**

Dependencies: **FIA_ATD.1 User attribute definition**

5.5 FMT : Security management

5.5.1 Class FMT : Actions to be taken

Function	Actions	Function	Actions
FAU_ARP.1	NA	FIA_UAU.4	NM
FAU_SAA.1	NA	FIA_UID.1	NA
FCS_CKM.3	a)	FIA_USB.1	a)
FCS_CKM.4	a)	FMT_MOF.1	a)
FCS_COP.1	NM	FMT_MSA.1	a)
FDP_ACC.2	NM	FMT_MSA.2	NM
FDP_ACF.1	a)	FMT_MSA.3	a)
FDP_DAU.1	a)	FMT_MTD.1	a)
FDP_ETC.1	NM	FMT_SMR.1	NA
FDP_ITC.1	a)	FPR_UNO.1	NA
FDP_RIP.1	NA	FPT_FLS.1	NM
FDP_SDI.2	NA	FPT_PHP.3	NM
FIA_AFL.1	a)	FPT_SEP.1	NM
FIA_ATD.1	a)	FPT_TDC.1	NM
FIA_UAU.1	a)	FPT_TST.1	NA
FIA_UAU.3	NM		

a) the letter refers to the respective management actions defined in CC V2.0 Part 2 Security functional requirement

NM No management activity

NA : Not Applicable

5.5.2 FMT_MOF Management of Functions in TSF

FMT_MOF.1 **Management of security functions behaviour**

Hierarchical to: No other components.

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*].**

Dependencies: **FMT_SMR.1 Security roles**

5.5.3 FMT_MSA Management of Security Attributes

FMT_MSA.1Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 **The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] 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*].**

Dependencies: **[FDP_ACC.1 Subset access control or**

FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_MSA.2Secure security attributes

Hierarchical to: No other components.

FMT_MSA.2.1 **The TSF shall ensure that only secure values are accepted for security attributes.**

Dependencies: **ADV_SPM.1 Informal TOE security policy model**

[FDP_ACC.1 Subset access control or

FDP_IFC.1 Subset information flow control]

FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1 **The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to provide [selection: *restrictive, permissive, other property*] default values for security attributes that are used to enforce the *SFP*.**

FMT_MSA.3.2 **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.**

Dependencies: **FMT_MSA.1 Management of security attributes**

FMT_SMR.1 Security roles

5.5.4 FMT_MTD Management of TSF Data

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

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*].**

Dependencies: **FMT_SMR.1 Security roles**

5.5.5 FMT_SMR Security Management Roles

FMT_SMR.1Security roles

Hierarchical to: No other components.

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.**

Dependencies: **FIA_UID.1 Timing of identification**

5.6 FPR : Privacy

5.6.1 FPR_UNO Unobservability

FPR_UNO.1 Unobservability

Hierarchical to: No other components.

FPR_UNO.1.1 **The TSF shall ensure that [assignment: *list of users and/or subjects*] are unable to observe the operation [assignment: *list of operations*] on [assignment: *list of objects*] by [assignment: *list of protected users and/or subjects*].**

Dependencies: No dependencies.

5.7 FPT : Protection of TOE security functions

5.7.1 FPT_FLS Fail Secure

FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

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*].**

Dependencies: **ADV_SPM.1 Informal TOE security policy model**

5.7.2 FPT_PHP TSF Physical Protection

FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

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.**

Dependencies: No dependencies.

5.7.3 FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

FPT_SEP.1.1 **The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.**

FPT_SEP.1.2 **The TSF shall enforce separation between the security domains of subjects in the TSC.**

Dependencies: No dependencies.

5.7.4 FPT_TDC Inter-TSF TSF Data Consistency

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

FPT_TDC.1.1 **The TSF shall provide the capability to consistently interpret [assignment: *list of TSF data types*] when shared between the TSF and another trusted IT product.**

FPT_TDC.1.2 **The TSF shall use [assignment: *list of interpretation rules to be applied by the TSF*] when interpreting the TSF data from another trusted IT product.**

Dependencies: No dependencies.

5.7.5 FPT_TST Self Test

FPT_TST.1 TSF testing

Hierarchical to: No other components.

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.**

Dependencies: **FPT_AMT.1 Abstract machine testing**

This dependency is not met by the PP (see rationale).

6. TOE Security Assurance Requirements

The security assurance requirement level is EAL 4 augmented with some of the assurance components as listed in the following section.
The claimed strength of TOE Security Functions is SOF-High.

6.1 ADV_IMP.2 : Implementation of the TSF

Dependencies:

ADV_LLD.1	Descriptive low-level design
ADV_RCR.1	Informal correspondence demonstration
ALC_TAT.1	Well-defined development tools

Developer action elements:

ADV_IMP.2.1D The developer shall provide the implementation representation for **the entire TSF**.

Content and presentation of evidence elements:

ADV_IMP.2.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.2.2C The implementation representation shall be internally consistent.

ADV_IMP.2.3C **The implementation representation shall describe the relationships between all portions of the implementation.**

Evaluator action elements:

ADV_IMP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_IMP.2.2E The evaluator shall determine that the **implementation representation** is an accurate and complete instantiation of the TOE security functional requirements.

6.2 ALC_DVS.2 : Sufficiency of security measures

ALC_DVS.2 Sufficiency of security measures

Dependencies:

No dependencies.

Developer action elements:

ALC_DVS.2.1D The developer shall produce development security documentation.

Content and presentation of evidence elements:

ALC_DVS.2.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.2.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.2.3C **The evidence shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.**

Evaluator action elements:

ALC_DVS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.2.2E The evaluator shall confirm that the security measures are being applied.

6.3 AVA_VLA.4 Highly resistant

Dependencies:

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

Developer action elements:

AVA_VLA.4.1D The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

AVA_VLA.4.2D The developer shall document the disposition of identified vulnerabilities.

Content and presentation of evidence elements:

AVA_VLA.4.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.4.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.4.3C The evidence shall show that the search for vulnerabilities is systematic.

AVA_VLA.4.4C **The analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.**

Evaluator action elements:

AVA_VLA.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.4.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.4.3E The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.4.4E	The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.
AVA_VLA.4.5E	The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a **high** attack potential.

7. Rationales

This section provides a rationale for the Protection Profile which demonstrates :

- how the security objectives provide effective countermeasures to the identified threats to security,
- how the security requirements are suitable to meet the TOE IT security objectives and together form a mutually supportive and internally consistent whole.

7.1 Security objectives rationale

This section demonstrates that the stated security objectives counter all the identified threats.
The following tables show which security objectives counter which threats phase by phase. It demonstrates that at least one security objective is correlated to at least one threat, and that each threat is countered by at least one objective.

7.1.1 Classes of threats relative to life cycle phases

As shown in table 3.1, threats can be expected in different phases of the TOE life-cycle, and can be countered either by the TOE (class I) or by the environment (class II) or by both . The TOE is designed during phase 1, but is constructed only at the end of phase 3.

T.CLON	Cloning can be done at any phase of card life. During phases 1 and 2, as the product is not materialized, it cannot contribute to countering the threat. During these phases, threat T.CLON can only be met by security objectives for the environment. TOE samples are finished products which are used during phase 1 for evaluations, and can help to counter T.CLON, but still the security objectives for the environment must be sufficient to meet the threat. For the remaining phases, 3 to 7, the TOE participates to countering the threats, but environment security procedures must still be applied.
T.DIS	Disclosure of software and data can be done at any phase of card life. During phases 1 and 2, as the product is not materialized, it cannot contribute to countering the threat and then only environmental procedures counter the threat. For the remaining phases, 3 to 7, the TOE counters the threats on embedded software and data. During the phases 3 and 6, more data are loaded in the TOE, so environmental procedures must also be taken to counter the threat.
T.DIS_INFO	The threat concerns data used for developing software. This data is present only during Smartcard software development, phase 1.
T.DIS_DEL	This threat is relative to delivery of information, software and/or data from phase 1 (software developer) to phase 2 (IC designer) and phase 3 (IC manufacturer). Part of the data, software, is transferred in a modified form from phase 2 to phase 3. Delivery to personalizer (phase 6), can come from the software developer (phase 1) or from the smartcard issuers, in which case it is considered inside the phase 6. As the data is not yet implemented in the TOE, the threat can only be countered by environment procedures.
T.DIS_TEST	Tests are conducted at the end of phases 1,2,3,4,5,6. These tests being part of the environmental procedures, this threats is countered by environmental procedures.

T.T_TOOLS	TOE development tools are used only during phase 1, therefore this threat only exists during phase 1. As the TOE is not yet manufactured, this threat is countered by environmental procedures.
T.T_SAMPLE	TOE samples are used only during phase 1, therefore this threat only exists during phase 1. The theft or unofficial use of samples is countered by environmental procedures.
T.T_DEL	This threat is relative to delivery of information, software and/or data from phase 1 (software developer) to phase 2 (IC designer) and phase 3 (IC manufacturer). Part of the data, software, is transferred in a modified form from phase 2 to phase 3. Delivery to personalizer (phase 6), can come from the software developer (phase 1) or from the smartcard issuers, in which case it is considered inside the phase 6. As the data is not yet implemented in the TOE, the threat can only be countered by environment procedures.
T.T_PRODUCT	The product exists only from phase 3 on. The threat can only be carried out during phases 3 to 7. The threat is partly met by environmental procedures. The product, when manufactured (phases 3 to 7) also counters the threat by limiting usage to the authenticated rightful owners.
T.FLAW	Flaws in the design of the TOE can only be introduced during the development phase (phase 1).
T.DESIGN_IC	The Integrated Circuit is designed during phase 2, so the threat concerns only this phase.
T.MOD	Modification of software and data can be done at any phase of Smartcard life cycle. During phases 1 and 2, as the product is not materialized, it cannot contribute to counter the threat. During the beginning of phase 3, (test phase) the TOE cannot counter the threat, but at the end (once the fuse has been blown), the TOE participates to countering it. For the remaining phases, 3 to 7, the TOE counters the threats on embedded software and data. During personalization phase (phase 6) more data is loaded, so that environmental procedures must also be taken to counter the treat.
T.MOD_INFO	The threatened information is only used for software development, so it can only be modified during phase 1.
T.MOD_DEL	This threat is relative to delivery of information, software and/or data from phase 1 (software development) to phase 2 (IC designer) and phase 3 (IC manufacturer). Part of the data, software is transferred in a modified form, from phase 2 to phase 3. Delivery to personalizer (phase 6), can come from the software developer (phase 1) or from the smartcard issuers , in which case it is considered inside the phase 6.. As the data is not yet implemented in the TOE, the threat can only be countered by environment procedures.

7.1.2 Threats addressed by security Objectives for the TOE

The product is constructed only after the end of phase 3 , therefore it can only meet functional requirements during phases 3 to 7. The threats to be addressed by the TOE are :

- T.CLON
- T.DIS
- T.T_PRODUCT
- T.MOD

The threat T.FLAW which appears only in phase 1 is to be covered by the TOE development methodology.

O.INTEGRITY addresses the integrity of the TOE once it is completed, thus it counters the threat T.MOD during phases 3 to 7.

O.TAMPER addresses illegal modification of the TOE once it is completed, thus it counters the threat T.MOD during phases 3 to 7.

O.FUNCTION addresses illegal use of the TOE , thus it counters the threat T.PRODUCT during phases 3 to 7. It also counters the use of a duplicate of the TOE, thus it counters T.CLON.

O.OPERATE Correct operations of the TOE security functions assures that it's confidential information cannot be disclosed, threat T.DIS, and that the operations cannot be corrupted, T.MOD, during phases 3 to 7.

O.FLAW addresses the threat T.FLAW during the conception of the TOE. This objective allows the TOE to counter the threats T.DIS and T.MOD once it is manufactured, phases 3 to 7.

O.DIS_MECHANISM addresses the Threat T.DIS. As knowledge of the security mechanisms is necessary for cloning, it also contributes to counter T.CLON. It helps to counter T.MOD by keeping confidential the security mechanisms which have to be broken to realize the threat. The TOE can fulfill this objective during phases 3 to 7.

O.DIS_MEMORY addresses the disclosure of TOE memory, threat T.DIS. As cloning requires knowledge of memory content. As knowledge of memory content is necessary for cloning, T.CLON is also addressed. The TOE can fulfill this objective during phases 3 to 7.

O.MOD_MEMORY addresses the modification of TOE memory, threat T.MOD. The TOE can fulfill this objective during phases 3 to 7.

O.CLON addresses the cloning of the TOE, threat T.CLON. By extension, this objective addresses the unauthorized use of embedded software functions which is part of T.T_PRODUCT. The TOE can fulfill this objective during phases 3 to 7.

	Security Objectives for the TOE								
Threats	INTEGRITY	TAMPER	FUNCTION	OPERATE	FLAW	DIS_ MECHANISM	DIS_ MEMORY	MOD_ MEMORY	CLON
CLON			X			X	X		X
DIS				X	X	X	X		
T_PRODUCT			X						X
MOD	X	X		X	X	X		X	
FLAW					X				

Table 7.1 : mapping of TOE objectives to threat

It is demonstrated that all class I threats and T.FLAW are addressed by at least one Security Objectives for the TOE.

7.1.3 Threats addressed by Security Objectives for the environment

7.1.3.1 Phase 1 Security Objectives

The threats present during phase 1 and which are not linked to delivery are :

- Threats occurring all the phases
  - T.CLON
  - T.DIS
  - T.MOD
- Threats specific to phase 1
  - T.DIS_INFO
  - T.DIS_TEST
  - T.T_TOOLS
  - T.T_SAMPLE
  - T.MOD_INFO

Threat T.FLAW is already addressed by the TOE development objective O.FLAW.

Threats T.T_DEL, T.MOD_DEL and T.DIS_DEL are considered later.

O.SOFT_ACS Restricting software access to authorized developers meets the threats T.DIS and T.MOD which require access to the software or related data. This knowledge is also necessary to mount threat T.CLON.

O.MECH_ACS Restricting access to the security mechanisms to authorized developers meets the threats T.DIS and T.MOD which require access to the software or related data. This knowledge is also necessary to mount threat T.CLON.

O.TI_ACS addresses disclosure and modification of related information. It thus addresses threats related to the illegal disclosure these information, T.DIS_INFO, and T.DIS_TEST or to their illegal modification T.MOD_INFO. This objective also helps addressing T.CLON, threat easier to mount if related information is known.

O.INIT_ACS addresses the part of T.DIS and T.MOD concerning initialization information. As this information is necessary to construct a TOE, O.INIT_ACS also addresses T.CLON.

O.TOOLS_ACS addresses specifically the threat T_TOOLS. If complete knowledge of the embedded software is not known, the development tools are necessary to build replica of the TOE. Thus O.TOOLS_ACS addresses T.CLON.

O.SAMPLE_ACS addresses specifically T.T_SAMPLE. Possession of samples is also a great help to finding the embedded software and data so to clone the TOE. Thus O.SAMPLE_ACS addresses also T.DIS and T.CLON.

	Security Objectives for the environment					
Threats	SOFT_ACS	MECH_ACS	TI_ACS	INIT_ACS	TOOLS_ACS	SAMPLE_ACS
CLON	X	X	X	X	X	X
DIS	X	X		X		X
MOD	X	X		X		
DIS_INFO			X			
DIS_TEST			X			
T_TOOLS					X	
T_SAMPLE						X
MOD_INFO			X			

Table 7.2 : mapping of objectives for the environment to threats relative to phase 1

It is demonstrated that all class II threats during phase 1 are addressed by at least one Security Objectives for the environment.

7.1.3.2 Phases 2 and delivery to phases 2, 3 and 6.

These phases concern more specifically the IC designer, the IC developer and the Personalizer who have to load data into the TOE and must exchange data with the preceding phases. Delivery of TOE itself is not addressed here.
The threats to be addressed are :

- Threats occurring during all the phases
  - T.CLON
  - T.DIS
  - T.MOD
- Threats on phase 2
  - T.DIS_TEST
  - T.DESIGN_IC
- Threats on delivery of data to phases 2, 3 and 6.
  - T.T_DEL
  - T.MOD_DEL
  - T.DIS_DEL.

O.DIS_DEV During phase 2 software test information is used by IC designer and IC manufacturer. O.DIS_DEV addresses threat T.DIS_TEST. Software data and personalization data is manipulated also during these phases so that O.DIS_DEV addresses also T.DIS and T.MOD. As the realization of these threats can lead to cloning, O.DIS_DEV addresses also T.CLON.

O.SOFT_DLV addresses specifically threats linked to delivery processes of data, T.T_DEL, T.MOD_DEL and T.DIS_DEL. As the realization of these threats allows T.CLON, T.DIS and T.MOD to be materialized, these threats are also addressed.

O_MECH_IC addresses specifically T_DESIGN_IC..

	Security Objectives for the environment		
Threats	DIS_DEV	SOFT_DLV	MECH_IC
CLON	X	X	
DIS	X	X	
DIS_DEL		X	
DIS_TEST	X		
DESIGN_IC			X
T_DEL		X	
MOD	X	X	
MOD_DEL		X	

Table 7.2 bis : mapping of objectives for the environment to threats relative to phases 2 and to delivery to phases 2, 3 and 6.

It is demonstrated that all class II threats during phases 2 and threats concerning delivery to phases 2, 3 and 6 are addressed by at least one security objectives for the environment.

7.1.3.3 Phases 3 to 7

The threats considered are those concerning the delivery of the product and it's management as well as threats using the physical characteristic of the IC in which the software and the data is embedded.

The threats are :

- T.CLON
- T.DIS
- T.MOD
- T.T_PRODUCT
- T.DIS_TEST

O.PRODUCT_DEV contributes to the protection of TOE data and related information including test information during phases 3 and 6, and thus addresses T.DIS, T.MOD and T.DIS_TEST. O.PRODUCT_DEV addresses directly T.T_PRODUCT and thus helps to counter T.CLON.

	Security Objectives for the environment
Threats	PRODUCT_DEV
CLON	X
DIS	X
T_PRODUCT	X
DIS_TEST	X
MOD	X

Table 7.2 ter : mapping of security objectivesfor the environment to threats relative to phases3 to 7

It is demonstrated that all class II threats during phases 3 to 7 are addressed by at least one Security Objectives for the environment.

7.1.4 Security assumptions met by the Security Objectives for the environment (after the development phase)

This section demonstrates that the security assumptions are suitably satisfied by the identified security objectives for the environment.
Each of the security objectives for the environment is addressed by assumptions.
The following tables (table 7.3 and 7.3 bis) demonstrate which assumptions contribute to the satisfaction of each IT security objective. For clarity, the table does not identify indirect dependencies.
This section describes why the security assumptions are suitable to provide each of the IT security objectives.

O.DIS_DEV is linked to A.DLV_CONTROL, A.DLV_CONF, A.DLV_PROTECT, A.DLV_TRANS, A.DLV_TRACE, A.DLV_AUDIT, A.DLV_RESP, A.IC_ORG, A.USE_PROD and A.USE_SYS

O.SOFT_DLV is linked to A.DLV_CONTROL, A.DLV_CONF, A.DLV_PROTECT, A.DLV_TRANS, A.DLV_TRACE, A.DLV_AUDIT, A.DLV_RESP, A.USE_TEST, A.USE_PROD and A.USE_SYS.

	Security Objectives for the environment							
Assumptions	SOFT_ACS	MECH_ACS	TI_ACS	INIT_ACS	TOOLS_ACS	SAMPLE_ACS	DIS_DEV	SOFT_DLV
DLV_CONTROL							X	X
DLV_CONF:							X	X
DLV_PROTECT							X	X
DLV_TRANS							X	X
DLV_TRACE							X	X
DLV_AUDIT							X	X
DLV_RESP							X	X
IC_ORG							X	
USE_TEST								X
USE_PROD							X	X
USE_SYS							X	X

Table7.3 : mapping of security assumptions and objectives for the environment

O.PRODUCT_DLV is linked to A.DLV_CONTROL, A.DLV_PROTECT, A.DLV_TRANS, A.DLV_TRACE, A.DLV_AUDIT, A.DLV_RESP, A.IC_ORG and A.USE_PROD.
O_MECH_IC is linked to A.IC_PRODUCT

	Security Objectives for the environment	
Assumptions	PRODUCT_DLV	MECH_IC
DLV_CONTROL	X	
DLV_PROTECT	X	
DLV_TRANS	X	
DLV_TRACE	X	
DLV_AUDIT	X	
DLV_RESP	X	
IC_ORG	X	
USE_PROD	X	
IC_PRODUCT		X

able7.3 bis : mapping of security assumptions and objectives for the environment

7.2 Security Requirements Rationale

Each of the security objectives for the environment during the development phase is addressed by at least one assurance requirement.

7.2.1 Security Assurance Requirements meet Security Objectives for the environment (during the development phase)

This section demonstrates that the combination of the Assurance components is suitable to satisfy the identified security objectives for the environment during the development phase.
Each of the security objectives for the environment is addressed by assurance components.

The following table (table 7.4) demonstrates which Assurance component contribute to the satisfaction of each security objective for the environment. For clarity, the table does not identify indirect dependencies.

	Security Objectives					
Assurance Components	SOFT_ACS	MECH_ACS	TI_ACS	INIT_ACS	TOOLS_ACS	SAMPLE_ACS
ALC_DVS.2	X	X	X	X	X	X

Table7.4 : mapping of assurance components and security objectives for the environment during the development phase.

The augmented assurance component ALC_DVS.2 measures are designed to meet access objectives and specifically O.SOFT_ACS, O.MECH_ACS, O.TI_ACS, O.INIT_ACS, O.TOOLS_ACS and O.SAMPLE_ACS.

7.2.2 Security Functional Requirements rationale

This section demonstrates that the combination of the security requirement objectives is suitable to satisfy the identified IT security objectives.
Each of the IT security objectives is addressed by functional requirements.

The following table (table 7.5) demonstrates which functional requirements contribute to the satisfaction of each security objective for the TOE. For clarity, the table does not identify indirect dependencies.
This section describes why the security requirements are suitable to provide each of the IT security objectives.

	Security Objectives for the TOE								
Security Functional Requirements	INTEGRITY	TAMPER	FUNCTION	OPERATE	O.FLAW	DIS_ MECHANISM	DIS_ MEMORY	MOD_ MEMORY	CLON
EAL4 requirements					X				
FAU_ARP.1	X					X	X	X	
FAU_SAA.1	X					X	X	X	
FCS_CKM.3		X					X		partial
FCS_CKM.4		X					X		partial
FCS_COP.1		X					X		partial
FDP_ACC.2		X	X	partial		X	X	X	partial
FDP_ACF.1		X	X			X	X	X	partial
FDP_DAU.1	X	X		partial				X	partial
FDP_ETC.1							X		partial
FDP_ITC.1	X							X	
FDP_RIP.1		X					X		partial
FDP_SDI.2	X			partial				X	partial
FIA_AFL.1		X		X					partial
FIA_ATD.1		X							partial
FIA_UAU.1		X					X	X	partial
FIA_UAU.3		X					X	X	partial
FAU_UAU.4							X	X	partial
FIA_UID.1		X					X	X	partial
FIA_USB.1		X					X	X	partial
FMT_MOF.1		X		X		X	X	X	
FMT_MSA.1		X				X	X	X	partial
FMT_MSA.2		X							
FMT_MSA.3		X				X	X	partial	partial
FMT_MTD.1		X				X	X	X	
FMT_SMR.1		X		X					
FPR_UNO.1						X			
FPT_FLS.1		X		X					
FPT_PHP.3		X							
FPT_SEP.1		X				X	X		
FPT_TDC.1	X	X							
FPT_TST.1	X								

Table7.5 : mapping of security functional requirements and IT objectives

The EAL4 assurance requirements contribute to the satisfaction of the O.FLAW security objective. They are suitable because they provide the assurance that the TOE is designed, implemented and operates so that the IT functional requirements are correctly provided.

Security audit functional requirements FAU_ARP.1 and FAU_SAA.1 detect security violating actions such as integrity loss (corresponding to the security objective O.INTEGRITY), and actions which could disclose security mechanisms (corresponding to the security objective O.DIS_MECHANISM), stored memory (corresponding to the security objective O.DIS_MEMORY), or modification of stored information (corresponding to the objective O.MOD_MEMORY).

Cryptographic support functional requirements : FCS_CKM.3, FCS_CKM.4 and FCS_COP.1 support the access control to the assets. These functions cooperate to meet the security objectives of O.TAMPER, O.DIS_MEMORY, and thus participate to meet the O.CLON security objective.

Access control functional requirements FDP_ACC.2 and FDP_ACF.1 control the access conditions. This fulfills the security objectives, O.TAMPER, O.FUNCTION, O.DIS_MECHANISM (Code) , O.DIS_MEMORY and O.MOD_MEMORY (Data). They participate to the fulfillment of O.CLON. FDP.ACC.2 contributes to the correct operation of the TOE (corresponding to the security objectives O.OPERATE. and O.CLON).

Data authentication functional requirement FDP.DAU.1 assures the objectives O.INTEGRITY, O.TAMPER, and O.MOD_MEMORY by verifying the evidence of validity of the data. It contributes to the correct operation of TOE, corresponding to the objective O.OPERATE and makes cloning more difficult, O.CLON.

Export to outside TSF control function FDP_ETC.1 contributes to realization of O.DIS_MEMORY by controlling the export of user data. It contributes to the correct operation of TOE, O.CLON.

Import from outside TSF control function FDP_ITC.1 contributes to realization of O.MOD_MEMORY by controlling the import of user data. This also contributes to O.INTEGRITY.

FDP_RIP.1 functional requirement meets O.TAMPER, and O.DIS_MEMORY objectives by assuring that previous information cannot be used out of context.. It also contributes to the correct operation of TOE, O.CLON which relies on disclosure of confidential information.

FDP_SDI.2 functional requirement meets O.INTEGRITY, and O.MOD_MEMORY objectives by detecting and acting on integrity errors. It also contributes to the correct operation of TOE, corresponding to the security objective O.OPERATE and O.CLON.

Identification and authentication functional requirements FIA_AFL.1 and FIA_ATD.1 meet the security objective O.TAMPER by handling authentication failures and maintaining user’s attributes. FIA_AFL.1 also meets the security objective O.OPERATE. They both also contribute to the correct operation of TOE, O.CLON.

Identification and authentication functional requirements FIA_UAU.1 and FIA.UAU.3 meet O.TAMPER, O.DIS_MEMORY and O.MOD_MEMORY objectives by managing the authentication of candidates. . All of them also contribute to the correct operation of TOE, O.CLON.

Identification and authentication functional requirement FIA_UAU.4 prevents an unauthorized access to stored memory, and thus contributes to fulfilling the security objectives O.DIS_MEMORY and O.MOD_MEMORY. It also contributes to the correct operation of TOE, O.CLON.

Identification and authentication functional requirements FIA_UID.1 and FIA_USB.1 meet O.TAMPER, O.DIS_MEMORY and O.MOD_MEMORY objectives by use of identification and by binding it to the subject. They also contribute to the correct operation of TOE, O.CLON.

FMT_MOF.1 functional requirement meets O.TAMPER, O.OPERATE, O.DIS_MECHANISM, O.DIS_MEMORY and O.MOD_MEMORY objectives by managing the security functions which fulfill these objectives.

Management of TSF data functional requirements FMT_MSA.1, FMT_MSA.2 and FMT_MSA.3 meet O.TAMPER objectives. FMT_MSA.1 and FMT_MSA.3 also meet O.DIS_MECHANISM, O.DIS_MEMORY and O.MOD_MEMORY objectives. They also contribute to the correct operation of TOE, O.CLON.

FMT_MTD.1 functional requirement meets O.TAMPER, O.DIS_MECHANISM, O.DIS_MEMORY and O.MOD_MEMORY objectives by management of TSF data.

FMT_SMR.1 functional requirement meets O.TAMPER, and O.OPERATE objectives due to role management.

Unobservability requirement FPR_UNO.1 assures that unauthorized parties cannot over look settings of cards security mechanisms, corresponding to the security objective O.DIS_MECHANISM.

FPT_FLS.1 functional requirement meets O.TAMPER and O.OPERATE, objectives by assuring secure state when failures occur (intentional or not).

FPT_PHP.3 by resisting tampering meets O.TAMPER by resisting to physical attacks. FPT_SEP.1 functional requirement meets O.TAMPER, O.DIS_MECHANISM and O.DIS_MEMORY objectives by keeping domain separation and preventing tempering outside allowed domains.

FPT_TDC.1 functional requirement meets O.INTEGRITY and O.TAMPER objectives by assuring inter TSF data consistency.

FPT_THT.1 functional requirement meets O.INTEGRITY objective by detecting non integrity during self tests..

7.2.3 Dependencies of security requirements.

This section is intended to be a demonstration that the dependencies between the security requirements components (functional and assurance) included in this PP are satisfied.

The assurance requirements specified in this PP are precisely as defined in EAL4 with several higher hierarchical components (ADV_IMP.2, ALC_DVS.2 and AVA_VLA.4). This is asserted to be a known set of assurance components for which all dependencies are satisfied.

The following table (table 7.7) lists all functional requirements components including security requirements on the IT environment. For each component, the dependencies specified in Common Criteria are listed, and a reference to the component number is given.

	Security functions	Dependencies	N°
1	FAU_ARP.1 Security Alarms	**FAU_SAA.1**	**2**
2	FAU_SAA.1 Potential violation analysis	FAU_GEN.1	*
3	FCS_CKM.3 : Cryptographic Key Access	FMT_MSA.2	22
		FDP_ITC.1 or FCS_CKM.1	10
		FCS_CKM.4	4
4	FCS_CKM.4 : Cryptographic Key Destruction	FMT_MSA.2	22
		FDP_ITC.1 or FCS_CKM.1	10
5	FCS_COP.1 : Cryptographic Operation	FMT_MSA.2	22
		FDP_ITC.1 or FCS_CKM.1	10
		FCS_CKM.4	4
6	FDP_ACC.2 : Access Control Policy	FDP_ACF.1	7
7	FDP_ACF.1 : security attributes based Access Control Functions	FDP_ACC.1	6
		FMT_MSA.3	23
8	FDP_DAU.1 : basic Data Authentication	no dependencies	
9	FDP_ETC.1 : Export of user data without security attributes	FDP_ACC.1 or FDP_IFC.1	6
10	FDP_ITC.1 : Import of user data without security attributes	FDP_ACC.1 or FDP_IFC.1	6
		FMT_MSA.3	23
11	FDP_RIP.1 : subset residual information protection	no dependencies	
12	FDP_SDI.2 : stored data integrity monitoring and action	no dependencies	
13	FIA_AFL.1 : basic authentication failure handling	FIA_UAU.1	15
14	FIA_ATD.1 : user attribute definition	no dependencies	
15	FIA_UAU.1 : timing of authentication	FIA_UID.1	18
16	FIA_UAU.3 : unforgettable authentication	no dependencies	
17	FIA_UAU.4 Single Use Authentication Mechanism	no dependencies	
18	FIA_UID.1 : timing of identification	no dependencies	
19	FIA_USB.1 : user-subject binding	FIA_ATD.1	14
20	FMT_MOF.1 : management of security functions behavior	FMT_SMR.1	25
21	FMT_MSA.1 : management of security attributes	FDP_ACC.1 or FDP_IFC.1	6
		FMT_SMR.1	25
22	FMT_MSA.2 : safe security attributes	ADV_SPM.1	by EAL4
		FDP_ACC.1 or FDP_IFC.1	6
		FMT_MSA.1	21
		FMT_SMR.1	25
23	FMT_MSA.3 : safe attributes initialization	FMT_MSA.1	21
		FMT_SMR.1	25
24	FMT_MTD.1 : management of TSF data	FMT_SMR.1	25
25	FMT_SMR.1 : security roles	FIA_UID.1	18
26	FPR_UNO.1 Unobservability	no dependencies	
27	FPT_FLS.1 : failure with preservation of secure state	ADV_SPM.1	by EAL4
28	FPT_PHP.3 Resistance to physical attacks	no dependencies	
29	FPT_SEP.1 : TSF domain separation	no dependencies	
30	FPT_TDC.1 : inter-TSF basic TSF data consistency	no dependencies	
31	FPT_TST.1 Testing	**FPT_AMT.1**	*****

Table 7.6 : dependencies analysis

* Dependencies not met for reasons given below.

The following dependencies marked by " * " in table 7.6 are not applicable to the TOE security functional requirements :

FDP_ACC.2 is hierarchical to FDP_ACC.1, therefore dependencies on FDP_ACC.1 can be met by FDP_ACC.2.

FAU_GEN.1 is not applicable to the TOE : Indeed if FAU_GEN.1 is chosen in the PP, it forces many security relevant events to be recorded, and this is not applicable to the smartcard as many of these events bring the card to an insecure state where recording itself could open a security breach.. We consider that the function FAU_SAA.1 may be used and specific audited events will have to be defined in an ST independently with FAU_GEN.1

FPT_TST.1 is self consistent for the TOE and the FPT_AMT.1 (Abstract Machine Testing) dependency does not need to be satisfied. As a matter of fact, the TOE software is not to be tested within the scope of FPT_TST.1. Moreover, in its relations with the outside world, typically the card reader, the TOE is always the slave, and thus cannot test the “outside world”. These are the reasons why FPT_TST.1 is self consistent and FPT_AMT.1 is not applicable.

7.2.4 Strength of functional level rationale

Due to the definition of the TOE, it is very important that the claimed SOF should be high since the product critical security mechanisms have to be only defeated by attackers possessing a high level of expertise, opportunity and resources, successful attack being judged to be beyond normal practicality.

7.2.5 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. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defense against such attacks, the evaluators should have access to the low level design and source code. The lowest for which such access is required is EAL4.
The assurance level EAL4 is achievable, since it requires no specialist techniques on the part of the developer.

7.2.6 Assurance augmentation rationale

Additional assurance requirements 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 high.

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 refinement. Embedded software source code is an example of TSF implementation representation.

The assurance component is a higher hierarchical component to EAL4 (only ADV_IMP.1 is found in EAL4). It is important for a smartcard embedded software 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.1 has dependencies with ADV_LLD.1 (“Description Low-Level design ”), ADV_RCR.1 (“ Informal correspondence demonstration ”), ALC_TAT.1 (“ well defined development tools ”). All these dependencies are satisfied by EAL4.

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 found in EAL4). Due to the nature of the TOE, there is a need for a justification of the sufficiency of these procedures to protect the confidentiality and integrity of the TOE.

ALC_DVS.2 has no dependencies.

AVA_VLA.4 Highly resistant.

Due to the definition of the TOE, it must be shown to be highly resistant to penetration attacks. This is due to the fact that a smartcard can be placed in a hostile environment, such as an electronic laboratory.

This assurance requirement is achieved by the AVL_VLA.4 component. 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 sophistication.

AVA_VLA.4 has dependencies with ADV_FSP.1 (“ Informal functional specification ”), ADV_HDL.2 (“ Security enforcing high-level design ”), ADV_LLD.1 (“ Descriptive low-level design ”), ADV-IMP.1 (“ Subset of the implementation of the TSF ”), AGD_ADM.1 (“ Administrator guidance ”), AGD_USR.1 (“ User guidance ”). All these dependencies are satisfied by EAL4.

7.2.7 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.

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 demonstrate mutual support and internal consistency between the functional requirements.

Inconsistency between functional and assurance requirements can only arise if there are functional-assurance dependencies that are not met, a possibility which has been shown not to arise.

Annex A

Glossary

Basic Software (BS) : is the part of ES in charge of the generic functions of the Smartcard IC such as Operating System, general routines and Interpreters.

DAC : Discretionary Access Control.

Dedicated Software (DS) : is defined as the part of ES provided to test the component and/or to manage specific functions of the component.

Embedded Software (ES) : is defined as the software embedded in the Smartcard Integrated Circuit. The ES may be in any part of the non-volatile memories of the Smartcard IC.

Embedded software developer : Institution (or its agent) responsible for the smartcard embedded software development and the specification of pre-personalization requirements.

Initialization : is the process to write specific information in the NVM during IC manufacturing and testing (phase 3) as well as to execute security protection procedures by the IC manufacturer. The information could contain protection codes or cryptographic keys.

Integrated Circuit (IC) : Electronic component(s) designed to perform processing and/or memory functions.

IC designer : Institution (or its agent) responsible for the IC development.

IC manufacturer : Institution (or its agent) responsible for the IC manufacturing, testing, and pre-personalization.

IC packaging manufacturer : Institution (or its agent) responsible for the IC packaging and testing**.**

Personalizer : Institution (or its agent) responsible for the smartcard personalization and final testing.

Personalization data : Specific information in the non volatile memory during personalization phase.

RBAC : Role-Based Access Control.

Security Information : Secret data, initialization data or control parameters for protection system.

Smartcard : A credit sized plastic card which has a non volatile memory and a processing unit embedded within it.

Smartcard Issuer : Institution (or its agent) responsible for the smartcard product delivery to the smartcard end-user.

Smartcard product manufacturer : Institution (or its agent) responsible for the smartcard product finishing process and testing.

Smartcard Application Software (AS) : is the part of ES dedicated to the applications.