Document Administration

Recipient

Department	Name

For the attention of

Department	Name

Summary

The following document comprises the Security Target Lite for a TOE evaluated according to Common Criteria Version 2.3. The TOE being subject of the evaluation is the smartcard product

MICARDO V3.0 R1.0

from Sagem Orga GmbH. The IT product under consideration shall be evaluated according to CC EAL 4 augmented with a minimum strength level for the TOE security functions of SOF-high.

Keywords

Target of Evaluation (TOE), Common Criteria, IC, Dedicated Software, Smartcard Embedded Software, Basic Software, Application Software, Security Objectives, Assumptions, Threats, TOE Security Function (TSF), TOE Security Enforcing Function (SEF), Level of Assurance, Strength of Functions (SOF), Security Functional Requirement (SFR), Security Assurance Requirement (SAR), Security Function Policy (SFP)

Responsibility for updating the document

Dr. Susanne Pingel

Sagem ORGA GmbH

MICARDO V3.0 R1.0

ST-Lite

Document Id:	3MIC3EVAL.CSL.0001
Archive:	3
Product/project/subject:	MIC3EVAL (Micardo V3 Evaluierung)
Category of document:	CSL (ST-Lite)
Consecutive number:	0001
Version:	V1.00
Date:	12 December 2006
Author:	Dr. Susanne Pingel
Confidentiality:	

Checked report:	not applicable
Authorized (Date/Signature):	not applicable
Accepted (Date/Signature):	not applicable

OdsDok V2.00

Document Organisation

i Notation

None of the notations used in this document need extra explanation.

ii Official Documents and Standards

See Bibliography.

iii Revision History

Version	Type of change	Author / team
V1.00	First edition	Dr. Susanne Pingel

Table of Contents

1 ST Introduction

1.1 ST Identification

This Security Target refers to the smartcard product “MICARDO V3.0 R1.0” (TOE) provided by Sagem Orga GmbH for a Common Criteria evaluation.

Title:	ST-Lite - MICARDO V3.0 R1.0
Document Category:	Security Target for a CC Evaluation (sanitized version of the complete Security Target)
Document ID:	Refer to Document Administration
Version:	Refer to Document Administration
Publisher:	Sagem Orga GmbH
Confidentiality:	public
TOE:	“MICARDO V3.0 R1.0” (Smartcard Product containing IC with Embedded Software, in particular intended to be used for high security applications within the Health, Identification and Banking market)
Certification ID:	BSI-DSZ-CC-0390
IT Evaluation Scheme:	German CC Evaluation Scheme
Evaluation Body:	SRC Security Research & Consulting GmbH
Certification Body:	Bundesamt für Sicherheit in der Informationstechnik (BSI)

This Security Target has been built in conformance with Common Criteria V2.3.

1.2 ST Overview

Target of Evaluation (TOE) and subject of this Security Target (ST) is the smartcard product “MICARDO V3.0 R1.0” developed by Sagem Orga GmbH.

The TOE provides the MICARDO V3.0 Operating System platform with a wide range of functionality which can be employed for different applications. The MICARDO V3.0 platform is designed as multifunctional platform for high security applications. The Operating System platform allows for an integration of a variety of applications, in particular in the following fields: Health Systems, ID Systems, Signature Applications with / without on-card signature key pair generation, Banking Systems, Loyalty Schemes.

In particular, the TOE and its technical functionality and inherently integrated security features are designed and developed under consideration of the following specifications, standards and requirements:

- Functional and security requirements defined in the specification /eHC1/ for the electronic Health Card (eHC) as employed within the German Health System
- Functional and security requirements defined in the specification /HPC-SMC1/ for the Health Professional Card (HPC) and the Security Module Card (SMC) as employed within the German Health System
- Functional and security requirements drawn from the EU Directive on electronic signatures /ECDir/, the German Signature Act /SigG01/, the German Signature Ordinance /SigV01/ and the catalogue of agreed cryptographic algorithms /ALGCAT/
- Requirements from the Protection Profiles /PP9911/, /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/, /PP SSCD Type2/
- Technical requirements defined in /ISO 7816/, Parts 1, 2, 3, 4, 8, 9, 15

Under technical view, the TOE comprises the following components:

- Integrated Circuit (IC) "Philips SmartMX P5CC036V1D Secure Smart Card Controller" provided by Philips Semiconductors GmbH
- Smartcard Embedded Software comprising the MICARDO V3.0 Operating System platform (based on a native implementation) and a so-called Minimal Filesystem provided by Sagem Orga GmbH

The Minimal Filesystem of the TOE consists of a minimal set of files and objects as necessary for using the MICARDO V3.0 Operating System platform.

Due to customer requirements, the Minimal Filesystem may be supplemented with further customer specific applications, files, objects and data whereat no executable code is allowed. However, these additional parts are explicitly not part of the CC-evaluation of the product. The supplements of the TOE´s Minimal Filesystem will be considered as configuration of the TOE. The configuration will be done by Sagem Orga GmbH prior to the delivery of the product, but due to the specified access rules the delivered configuration may be changed after delivery. However, the TOE contains at its delivery unalterable identification information on the delivered configuration.

For the delivery of the TOE two different ways are established:

- The TOE is delivered to the customer in form of a complete initialised smartcard.
- Alternatively, the TOE is delivered to the customer in form of an initialised module. In this case, the smartcard finishing process (embedding of the delivered modules, final card tests) is task of the customer.

As the form of the delivery of the TOE does not concern the security features of the TOE in any way the TOE will be named in the following with “MICARDO Card” for short, independently of its form of delivery.

In order to be compliant with the requirements from the German Health System and the EU Directive on electronic signatures /ECDir/, the German Signature Act /SigG01/ and the German Signature Ordinance /SigV01/ the TOE will be evaluated according to CC EAL 4 augmented with a minimum strength level for the TOE security functions of SOF-high.

The main objectives of this ST are

- to describe the TOE as a smartcard product for high security applications (in particular intended to be used within Health, Identification and Banking Systems)
- to define the limits of the TOE
- to describe the assumptions, threats and security objectives for the TOE
- to describe the security requirements for the TOE
- to define the TOE security functions

1.3 CC Conformance

The CC evaluation of the TOE is based upon

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 2.3, August 2005 (/CC 2.3 Part1/)
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements, Version 2.3, August 2005 (/CC 2.3 Part2/)
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements, Version 2.3, August 2005 (/CC 2.3 Part3/)

For the evaluation the following methodology will be used:

- Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology, Version 2.3, August 2005 (/CEM 2.3 Part2/)

This Security Target is written in accordance with the above mentioned Common Criteria Version 2.3 and claims the following CC conformances:

- Part 2 extended
- Part 3 conformant

The Security Target is based on the Protection Profile PP 9911 „Smartcard Integrated Circuit with Embedded Software“ (/PP9911/). The IC evaluation in compliance with the Protection Profile PP 9806 (/PP9806/) as required in PP 9911 is replaced by the comparable IC evaluation according to the Protection Profile BSI-PP-0002 (/BSI-PP-0002/). Refer for this to the report of the BSI concerning the comparibility of the Protection Profiles PP 9806 and BSI-PP-0002 (/CompPP9806-BSIPP0002/).

Furthermore, as the TOE is intended to be used for the German Health System (eHC, HPC, SMC) and for Secure Signature-Creation Devices (SSCD), the Security Target takes into account the following Protection Profiles: /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/, /PP SSCD Type2/.

The chosen level of assurance for the TOE is **EAL 4 augmented**. The augmentation includes the assurance components ADV_IMP.2, ATE_DPT.2, AVA_MSU.3 and AVA_VLA.4.

The minimum strength level for the TOE security functions is **SOF-high**.

In order to avoid redundancy and to minimize the evaluation efforts, the evaluation of the TOE will be conducted as a composite evaluation and will make use of the evaluation results of the CC evaluation of the underlying semiconductor "Philips SmartMX P5CC036V1D Secure Smart Card Controller" provided by Philips Semiconductors GmbH. The IC incl. its IC Dedicated Software is evaluated according to Common Criteria EAL 5 augmented with a minimum strength level for its security functions of SOF-high and is listed under the Certification ID BSI-DSZ-CC-0293. The evaluation of the IC is based on the Protection Profile BSI-PP-0002 (/BSI-PP-0002/).

2 TOE Description

2.1 TOE Definition

2.1.1 Overview

The Target of Evaluation (TOE) is the smartcard product " MICARDO V3.0 R1.0" (MICARDO Card for short in the following) implemented as platform for high security applications, in particular in the framework of Health, Identification and Banking Systems.

In technical view the MICARDO Card is realised as a proprietary operating system with an Application Layer directly set-up on this operating system layer.

The MICARDO Card is based on the microcontroller "Philips SmartMX P5CC036V1D Secure Smart Card Controller" provided by Philips Semiconductors GmbH. The IC incl. its Dedicated Software is evaluated according to Common Criteria EAL 5 augmented with a minimum strength level for its security functions of SOF-high (refer to Certification ID BSI-DSZ-CC-0293).

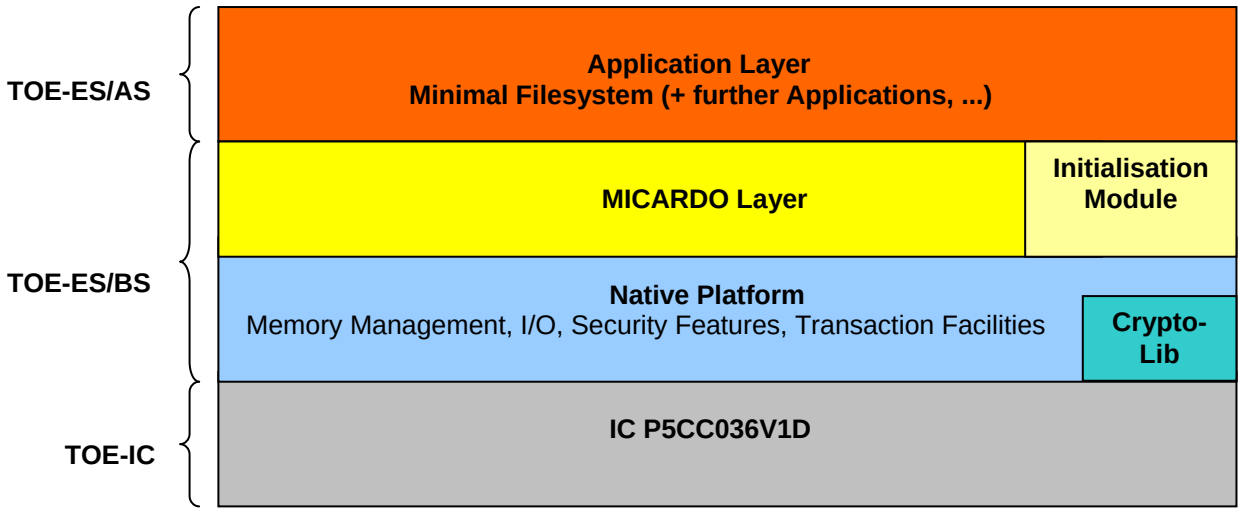
Roughly spoken, the TOE is composed from the following parts:

- Integrated Circuit (IC) with its proprietary IC Dedicated Software (TOE-IC)
- Smartcard Embedded Software (TOE-ES) consisting of
  - Basic Software (TOE-ES/BS)
  - Application Software (TOE-ES/AS)

While the Basic Software consists of the MICARDO V3.0 Operating System platform of the TOE (realised as native implementation), the Application Software covers the Application Layer which is directly set-up on the MICARDO V3.0 Operating System platform and implements the specific Minimal Filesystem and possibly further customer specific applications, objects, files and data.

Furthermore, the MICARDO Card itself offers the possibility to check its authenticity. For this purpose, the MICARDO Card contains the private part of a dedicated authentication key pair which depends on the configuration of the TOE and may be chosen customer specific (for more details see chap. 2.1.4.2).

The following figure shows the global architecture of the TOE and its components:



The different components of the TOE depicted in the figure above will be described more detailed in the following sections.

2.1.2 TOE Product Scope

The following table contains an overview of all deliverables associated to the TOE:

TOE component	Description / Additional Information	Type	Transfer Form
TOE-IC	Philips SmartMX P5CC036V1D Secure Smart Card Controller (incl. its IC Dedicated Software)	HW / SW	---
TOE-ES/BS	Smartcard Embedded Software / Part Basic Software (implemented in ROM/EEPROM of the microcontroller)	SW	---
TOE-ES/AS	Smartcard Embedded Software / Part Application Software (depending on the TOE´s configuration, implemented in the EEPROM of the microcontroller)	SW	---
Note: The TOE itself will be delivered as initialised smartcard or as initialised module.			
User Guide	User guidance for the Administrator / User of the MICARDO Card	DOC	Document in paper / electronic form
Identification Data Sheet of the MICARDO Card	Data Sheet with information on the actual identification data and configuration of the MICARDO Card delivered to the customer	DOC	Document in paper / electronic form
Aut-Key of the MICARDO Card	Public part of the authentication key pair relevant for the authenticity of the MICARDO Card Note: The card´s authentication key pair is generated by Sagem Orga GmbH and depends on the TOE´s configuration delivered to the customer. Furthermore, the key pair may be chosen customer specific.	KEY	Document in paper form / electronic file

Note: Deliverables in paper form require a personal passing on. For deliverables in electronic form an integrity and authenticity attribute will be attached.

2.1.3 Integrated Circuit (IC) with its Dedicated Software

Basis for the TOE´s Smartcard Embedded Software is the microcontroller “Philips SmartMX P5CC036V1D Secure Smart Card Controller”. The microcontroller and its Dedicated Software are developed and produced by Philips Semiconductors GmbH (within phase 2 and 3 of the smartcard product life-cycle, see chap. 2.2).

Detailed information on the IC Hardware, the IC Dedicated Software and the IC interfaces can be found in /ST-ICPhilips/.

2.1.4 Smartcard Embedded Software

The Smartcard Embedded Software of the TOE comprises the MICARDO V3.0 Operating System platform and applications running on this platform and is therefore divided into two parts with specific contents:

- Basic Software (MICARDO V3.0 Operating System platform)
- Application Software (Application Layer with Minimal Filesystem and possibly further customer specific applications, objects, files and data (not evaluation-relevant))

Each part of the Smartcard Embedded Software is designed and developed by Sagem Orga GmbH in phase 1 of the smartcard product life-cycle (see chap. 2.2). In particular, the additional customer specific application software supplementing the Minimal Filesystem (if required) is designed due to the customer´s requirements and implemented by Sagem Orga GmbH. Embedding of the Smartcard Embedded Software into the TOE is performed in the later phases 3 and 5.

The main parts of the Basic Software are brought into the card by the IC manufacturer in form of the ROM mask and stored in the User-ROM of the IC (phase 3). The Application Software, and perhaps additional parts of the Basic Software, are located in the EEPROM area and are lateron loaded by specific initialisation routines of the TOE (phase 5). Hereby, the loading requires an encrypted and with a cryptographic checksum secured initialisation file. The necessary keys for securing the initialisation process are stored inside the IC during production time.

2.1.4.1 Basic Software

The Basic Software of the Smartcard Embedded Software comprises the MICARDO V3.0 Operating System platform of the TOE. Its main and security related parts are stored in the User-ROM of the underlying IC and are brought into the smartcard in form of the so-called ROM mask during the production process of the IC within phase 3 of the smartcard product life-cycle (see chap. 2.2).

The MICARDO V3.0 Operating System platform of the TOE is designed as proprietary software consisting of two layers. In detail, the integral parts of the TOE´s operating system consist of the MICARDO Layer and the Initialisation Module. Both are based on a Native Platform which serves as an abstraction layer towards the IC. On the other side, the MICARDO Layer and the Initialisation Module provide an interface between the operating system and the overlying Application Layer with the Minimal Filesystem and further applications, files, objects and data.

The MICARDO Layer implements the executable code for the card commands and all general technical and security functionality of the MICARDO V3.0 Operating System platform as data objects and structures, file and object handling, security environments, security resp. cryptographic algorithms, key and PIN management, security states, access rules, secure messaging etc.

As mentioned, the Native Platform of the TOE´s operating system serves as an abstraction layer between the MICARDO Layer resp. the Initialisation Module and the IC. For this task, it provides essential operating system components and low level routines concerning memory management, I/O handling, transaction facilities, system management, security features and cryptographic mechanisms.

For the cryptographic features, the Native Platform integrates a specific module, the Crypto Library, which supports and implements the TOE´s core cryptographic functionality. In view of the Smartcard Embedded Software, the Crypto Library is accessible only via the Native Platform.

For the initialisation process of the TOE conducted within phase 5 of the smartcard product life-cycle (see chap. 2.2) the operating system of the TOE puts dedicated initialisation routines at disposal which are solely accessible during the initialisation phase and which are realised within the Initialisation Module. After the initialisation has been successfully completed these commands are no longer available. Furthermore, the functionality of deleting the complete initialisation file after the initialisation (deletion of the whole EEPROM area) is disabled for the TOE.

The Initialisation Module puts the following features at disposal:

- specific initialisation routines
- specific test routines for the EEPROM area

Loading of an initialisation file is only possible by use of the TOE´s specific initialisation routines. Hereby, the initialisation file to be loaded has to be secured before with an encryption and a cryptographic checksum, both done with dedicated keys of the TOE.

The test routines for the EEPROM area can be used for a check of the correct functioning of the memory.

Furthermore, the Initialisation Module manages the specific states of the TOE´s operating system according to specified and unalterable rules.

2.1.4.2 Application Software

The Application Software part of the TOE´s Smartcard Embedded Software comprises the Application Layer and is directly set-up on the TOE´s Basic Software. It consists of the TOE´s Minimal Filesystem and possibly further customer specific applications, objects, files and data, whereat the objects mentioned at last are explicitly not relevant for the TOE´s CC-evaluation.

The Minimal Filesystem of the TOE consists of a minimal set of files and objects as necessary for initial use of the MICARDO V3.0 Operating System platform. Due to customer requirements, the Minimal Filesystem may be supplemented with further customer specific applications, files, objects and data whereat no executable code is allowed. However, as mentioned before, these additional parts are explicitly not part of the CC-evaluation of the product.

The Application Software will be brought into the smartcard in cryptographically secured form during the initialisation process within phase 5 of the smartcard product life-cycle (see chap. 2.2). The initialisation process uses the specific initialisation routines of the TOE´s operating system, and the Application Software will be stored in the EEPROM area of the IC.

After the initialisation of the MICARDO Card has been completed, the following conditions hold: There is no possibility to delete the Minimal Filesystem. In particular, there will be neither a deletion of the Minimal Filesystem itself nor a deletion of the whole EEPROM area by a regular command of the operating system possible.

The MICARDO Card offers the capability to check its authenticity. For this purpose, the Minimal Filesystem contains the private part of a dedicated authentication key pair (RSA key od at least 1024 Bit modulus length) over which by an internal authentication procedure the authenticity of the MICARDO Card can be proven. The authentication key pair depends on the Initialisation File (containing the Application Software to be initialised) and its configuration and may be chosen customer specific. The corresponding public part of the authentication key pair is delivered through a trusted way to the external world.

Furthermore, the Minimal Filesystem contains a data area for storing identification data of the TOE and its configuration. The data area will be filled in the framework of the initialisation of the TOE with a specific operating system command and can be read out with a further specific operating system command. Once the identification data have been written, there is afterwards no change possible.

2.2 TOE Life-Cycle

The smartcard product life-cycle of the TOE is decomposed into seven phases. In each of these phases different authorities with specific responsibilities and tasks are involved:

Phase		Description
Phase 1	**Smartcard Embedded Software Development**	The **Smartcard Embedded Software Developer (Sagem Orga GmbH)** is in charge of • the development of the Smartcard Embedded Software (Basic Software, Application Software) and • the specification of the IC initialisation and pre-personalisation requirements (though the actual data for the IC initialisation and pre-personalisation come from Phase 4, 5 resp. 6). The purpose of the Smartcard Embedded Software designed during phase 1 is to control and protect the TOE during phases 4 to 7 (product usage).The global security requirements of the TOE are such that it is mandatory during the development phase to anticipate the security threats of the other phases.
Phase 2	**IC Development**	The **IC Designer (Philips Semiconductors GmbH)** • designs the IC, • develops the IC Dedicated Software, • provides information, software or tools to the Smartcard Embedded Software Developer, and • receives the Smartcard Embedded Software (only Basic Software) from the developer through trusted delivery and verification procedures. From the IC design, IC Dedicated Software and Smartcard Embedded Software, the **IC Designer (Philips Semiconductors GmbH)** • constructs the smartcard IC database, necessary for the IC photomask fabrication.
Phase 3	**IC Manufacturing and Testing**	The **IC Manufacturer (Philips Semiconductors GmbH)** is responsible for • producing the IC through three main steps: - IC manufacturing, - IC testing, and - IC pre-personalisation. The **IC Mask Manufacturer (Philips Semiconductors GmbH)** • generates the masks for the IC manufacturing based upon an output from the smartcard IC database.
Phase 4	**IC Packaging and Testing**	The **IC Packaging Manufacturer (Sagem Orga GmbH)** is responsible for

		• the IC packaging (production of modules) and • testing.
Phase 5	**Smartcard Product Finishing Process**	The **Smartcard Product Manufacturer (Sagem Orga GmbH)** is responsible for • the initialisation of the TOE (in form of the initialisation of the modules of phase 4) and • its testing. The smartcard product finishing process comprises the embedding of the initialised modules for the TOE and the card production what is done alternatively by **Sagem Orga GmbH or by the customer.** Final card tests only aim at checking the quality of the card production, in particular concerning the bonding and implantation of the modules.
Phase 6	**Smartcard Personalisation**	The **Personaliser** is responsible for • the administration of the smartcard (loading of new objects, files, applications due to customer requirements) • the smartcard personalisation and • final tests. Loading of additional executable code is not allowed. The personalisation of the smartcard includes the printing of the (card holder specific) visual readable data onto the physical smartcard, and the writing of (card holder specific) TOE User Data and TSF Data into the smartcard.
Phase 7	**Smartcard End-Usage**	The **Smartcard Issuer** is responsible for • the smartcard product delivery to the smartcard end-user (card holder), and the end of life process.

Appropriate procedures for a secure delivery process of the TOE or parts of the TOE under construction from one development resp. production site to another site within the smartcard product life-cycle are established. This concerns any kind of delivery performed from phase 1 to 5, including:

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

In particular, the delivery of the ROM mask and the EEPROM pre-personalisation data from Sagem Orga GmbH to Philips Semiconductors GmbH is done by using the dedicated secured delivery procedure specified by Philips Semiconductors GmbH following the so-called Philips Order Entry Form P5CC036V1D.

The IC manufacturer Philips Semiconductors GmbH delivers the IC with its IC Dedicated Software and the ROM mask supplied by Sagem Orga GmbH at the end of phase 3 in form of wafers according to /UG-ICPhilips/, chap. 2.1, Delivery Method 2, bullet point 1. The IC

Dedicated Test Software stored in the Test-ROM is disabled before the delivery of the IC and cannot be used in the following phases.

The FabKey procedure described in /UG-ICPhilips/, chap. 2.1, Delivery Method 2, bullet point 2 is replaced by the following procedure which provides at least equivalent security: The TOE´s operating system puts in the non-initialised status the command “Verify ROM” at disposal, with which a SHA-1 hash value over the complete ROM and data freely chosen by the external world can be generated. Prior to the initialisation of the IC, the authenticity of the IC with its ROM mask will be proven by using the functionality “Verify ROM” and comparing the new generated hash value over the ROM data and the data freely chosen with a corresponding external reference value which is accessible only for Sagem Orga GmbH.

With regard to the smartcard product life-cycle of the TOE described above, the different development and production phases of the TOE with its IC incl. its IC Dedicated Software and with its Smartcard Embedded Software (Basic Software, Application Software) are part of the evaluation of the TOE. Two different ways for the delivery of the TOE are established:

- The TOE is delivered at the end of phase 5 in form of complete cards, i.e. after the initialisation process of the TOE has been successfully finished, final card tests have been successfully conducted and the card production has been fulfilled.
- Alternatively, the TOE is delivered in form of initialised and tested modules. In this case, the smartcard finishing process (embedding of the delivered modules, final card tests) is task of the customer.

2.3 TOE Environment

Considering the TOE and its life-cycle described above, four types of environments can be distinguished:

- development environment corresponding to phase 1 and 2,
- production environment corresponding to phase 3 to phase 5,
- personalisation environment corresponding to phase 6,
- end-user environment corresponding to phase 7.

2.3.1 Development Environment

Phase 1 - Smartcard Embedded Software Development

To assure security of the development process of the Smartcard Embedded Software, a secure development environment with appropriate personnel, organisational and technical security measures at Sagem Orga GmbH is established.

Only authorized and experienced personnel which understands the importance and the rigid implementation of the defined security procedures is involved in the development activities.

The development process comprises the specification, the design, the coding and the testing of the Smartcard Embedded Software. For design, implementation and test purposes secure computer systems preventing unauthorized access are used. For security reasons the coding and testing activities will be done independently of each other.

All sensitive documentation, data and material concerning the development process of the Smartcard Embedded Software are handled in an appropriately and sufficiently secure way. This concerns both the transfer as well as the storing of all related sensitive documents, data and material. Furthermore, all development activities run under a configuration control system which guarantees for an appropriate traceability and accountability.

The Smartcard Embedded Software of the developer, more precise the Basic Software part dedicated for the ROM of the IC, is delivered to the IC manufacturer through trusted delivery and verification procedures. The Application Software and additional parts of the Basic Software are delivered in form of a cryptographically secured initialisation file as well through trusted delivery and verification procedures to the initialisation centre.

Phase 2 – IC Development

During the design and layout process only people involved in the specific development project for the IC have access to sensitive data. Different people are responsible for the design data of the IC and for customer related data. The security measures installed at Philips Semiconductors GmbH ensure a secure computer system and provide appropriate equipment for the different development tasks.

2.3.2 Production Environment

Phase 3 - IC Manufacturing and Testing

The verified layout data are provided by the developers of Philips Semiconductors GmbH directly to the wafer fab. The wafer fab generates and forwards the layout data related to the relevant photomask to the IC mask manufacturer (Philips Semiconductors GmbH).

The photomask is generated off-site and verified against the design data of the development before usage. The accountability and traceability is ensured among the wafer fab and the photomask provider.

The production of the wafers includes two different steps regarding the production flow. In the first step the wafers are produced with the fixed mask independent of the customer. After that step the wafers are completed with the customer specific mask and the remaining mask. The computer tracking ensures the control of the complete process including the storage of the semifinished wafers.

The test process of every die is performed by a test centre of Philips Semiconductors GmbH.

Delivery processes between the involved Philips Semiconductors GmbH sites provide accountability and traceability of the produced wafers. The delivery of the ICs from Philips Semiconductors GmbH to Sagem Orga GmbH is made in form of wafers whereby non-functional ICs are marked on the wafer.

Phase 4 – IC Packaging and Testing

For security reasons the processes of IC packaging and testing at Sagem Orga GmbH are done in a secure environment with adequate personnel, organisational and technical security measures.

Only authorized and experienced personnel which understands the importance and the rigid implementation of the defined security procedures is involved in these activities.

All sensitive material and documentation concerning the production process of the TOE is handled in an appropriately and sufficiently secure way. This concerns both the transfer as well as the storing of all related sensitive material and documentation. All operations are done in such a way that appropriate traceability and accountability exist.

Phase 5 - Smartcard Product Finishing Process

To assure security of the initialisation process of the TOE, a secure environment with adequate personnel, organisational and technical security measures at Sagem Orga GmbH is established.

Only authorized and experienced personnel which understands the importance and the rigid implementation of the defined security procedures is involved in the initialisation and test activities.

The initialisation process of the TOE comprises the loading of the TOE´s Application Software and the remaining EEPROM-parts of the TOE´s Basic Software which have been

specified, coded, tested and cryptographically secured in phase 1 of the product life-cycle. The TOE allows only the initialisation of the intended initialisation file with its Application Software and its parts of the Basic Software. For security reasons, secure systems within a separate network and preventing unauthorized access are used for the initialisation process.

If the TOE is delivered in form of initialised and tested modules, the smartcard finishing process, i.e. the embedding of the delivered modules and final card tests, is task of the customer.

Otherwise, the smartcard finishing process is part of the production process at Sagem Orga GmbH, and the TOE is delivered in form of complete (initialised) cards.

All sensitive documentation, data and material concerning the production processes of the TOE at Sagem Orga GmbH within phase 5 are handled in an appropriately and sufficiently secure way. This concerns both the transfer as well as the storing of all related sensitive documents, data and material. Furthermore, all operations run under a control system which supplies appropriate traceability and accountability.

At the end of this phase, the TOE is complete as smartcard and can be supplied for delivery to the personalisation centre for personalisation.

2.3.3 Personalisation Environment

Note: The phases from the TOE delivery at the end of phase 5 to phase 7 in the smartcard product life-cycle are not part of the TOE development and production process in the sense of this Security Target. Information about the phases 6 and 7 are just included to describe how the TOE is used after its development and production.

Phase 6 - Smartcard Personalisation

Central task for the personaliser is the personalisation of the initialised product, i.e the loading of card resp. card holder specific data (e.g. into the applications already existing on the initialised card). Furthermore, the personaliser may perform card management activities tasks and, for instance, add or delete applications, objects and data on the card by using the regular card commands if allowed by the present card structure and the access rules already set.

The personalisation process and its security depends directly on the access rules which have been initialised. For instance, the already existing applications on the card may require for their personalisation a mutual authentication between the card and the personalisation unit with session key agreement and a following data transfer secured by Secure Messaging using the agreed session keys.

However, the establishment of a secure environment for the personalisation process with adequate personnel, organisational and technical security measures is in the responsibility of the personalisation centre itself. In particular, the personaliser is responsible for the set-up of a secure personalisation process and for taking into account the requirements and recommendations given in the TOE´s user guidance. The secure key management and handling of the cryptographic keys for securing the data transfer within the personalisation process (if applicable) and the secure handling of the personalisation data itself is task of the personalisation centre.

2.3.4 End-User Environment

Phase 7 – Smartcard End-usage

In the end-usage phase, the TOE is under control of the card holder, and the applications, objects and data residing on the card are used in their intended way. However, according to the card structure and the access rules set for the different objects, further card management activities (as e.g. deleting or adding applications, inserting further personalisation data) may be possible.

2.4 TOE Intended Usage

Introducing information on the intended usage of the TOE is given within chap. 1.2. The present chapter will provide additional and more detailed information on the Operating System platform and on the Minimal Filesystem residing on the card at delivery time point.

The MICARDO V3.0 Operating System platform is designed as multifunctional platform for high security applications. Therefore, the TOE provides an Operating System platform with a wide range of technical functionality and an adequate set of inherently integrated security features.

The MICARDO V3.0 Operating System platform supports the following services:

- Oncard-generation of RSA key pairs of high quality (with appropriate key lengths)
- Different signature schemes (based on RSA with appropriate key lengths and padding schemes)
- Different encryption schemes (based on DES and RSA with appropriate key lengths and padding schemes)
- Key derivation schemes
- PIN based authentication scheme
- Different key based authentication schemes (based on DES and RSA, with / without session key agreement)
- Hash value calculation
- Random number generation of high quality
- Calculation and verification of cryptographic checksums
- Verification of CV certificates
- Protection of the communication between the TOE and the external world against disclosure and manipulation (Secure Messaging)
- Protection of files and data by access control functionality
- Life-cycle state information related to the Operating System itself as well as to all objects processed by the card
- Confidentiality of cryptographic keys, PINs and further security critical data
- Integrity of cryptographic keys, PINs and further security critical data
- Confidentiality of operating system code and its internal data
- Integrity of operating system code and its internal data (self test functionality)
- Resistance of crypto functionality against Side Channel Analysis (SPA, DPA, TA, DFA)
- Card management functionality
- Channel management (with separation of channel related objects)

To support the security of the above mentioned features of the TOE, the MICARDO V3.0 Operating System platform provides appropriate countermeasures for resistance especially against the following attacks:

- Cloning of the product
- Unauthorised disclosure of confidential data (during generation, storage and processing)
- Unauthorised manipulation of data (during generation, storage and processing)
- Identity usurpation
- Forgery of data to be processed
- Derivation of information on the private key from the related public part for oncard-generated RSA key pairs
- Side Channel Attacks

The resistance of the TOE against such attack scenarios is reached by usage of appropriate security features already integrated in the underlying IC as well as by implementing additional appropriate software countermeasures.

The Minimal Filesystem of the TOE comprises a system of objects which contains a minimum of files and data necessary for the usage of the MICARDO V3.0 Operating System platform.

The further usage resp. configuration of the TOE initially configured with the Minimal Filesystem lies in the responsibility of the customer. The correct and secure usage of the functionality provided by the TOE as described above – as far as accessible for the user of the MICARDO Card – is task for the customer resp. for the designer of the applications which are intended to be run on the MICARDO V3.0 Operating System platform. In particular, it is up to the customer to appropriately make use of the functionality of the MICARDO V3.0 Operating System platform and its security features and to appropriately set up file and data systems with adequate security structures as required by the intended customer applications.

3 TOE Security Environment

3.1 Assets

Assets are security–relevant elements to be directly protected by the TOE whereby assets have to be protected in terms of confidentiality and integrity. Confidentiality of assets is always intended with respect to untrusted users of the TOE and its security-critical components, whereas the integrity of assets is relevant for the correct operation of the TOE and its security-critical components.

The confidentiality of the code of the TOE is included in this ST for several reasons. First, the confidentiality is needed for the protection of intellectual/industrial property on security or effectiveness mechanisms. Second, though protection shall not rely exclusively on code confidentiality, disclosure of the code may weaken the security of the involved application. For instance, knowledge about the implementation of the operating system or the applications running on the operaing system may benefit an attacker. This also applies to internal data of the TOE, which may similarly provide leads for further attacks.

For a description of the TOE´s assets refer to /PP9911/, chap. 3.1, /BSI-PP-0002/, chap. 3.1, /ST-ICPhilips/, chap. 3.1. The assets of the TOE sorted in primary and seconday assets are listed in the tables below:

Primary Assets	
Part of the TOE	**Definition**
IC	---
Smartcard Embedded Software / Basic Software	---
Smartcard Embedded Software / Application Software	- identification data (information on the configuration of the card)

Secondary Assets	
Part of the TOE	**Definition**
IC	- logical design data - physical design data - IC Dedicated Software - initialisation data - pre-personalisation data - specific development aids - test and characterisation related data - material for software development support - photomasks - special functions for the communication with an external interface device - cryptographic co-processor for Triple-DES - FameXE co-processor for basic arithmetic functions to perform asymmetric cryptographic algorithms - random number generator - TSF data
Smartcard Embedded Software / Basic Software	- specifications - code - related documentation - system specific data - initialisation data - specific development aids - test and characterisation related data - material for software development support - TSF data
Smartcard Embedded Software / Application Software	- specifications - code - related documentation - system specific data - initialisation data - specific development aids - test and characterisation related data - material for software development support - user data related documentation - TSF data

3.2 Assumptions

3.2.1 General Assumptions for the TOE

The general assumptions made on the environment of the TOE are defined according to /PP9911/, chap. 3.2 and are suitably supplemented for the TOE. The complete set of assumptions is listed in the table below.

Note:

In the following table, items of /PP9911/ which are common with /PP9806/ are indicated by a "*".

Assumptions for the Environment of the TOE	
Name	**Definition**
Assumptions on Phase 1 to 5	
A.DEV_ORG* (PP9911+supplement)	**Protection of the TOE under Development and Production** Procedures dealing with physical, personnel, organizational, technical measures for the confidentiality and integrity of the Smartcard Embedded Software (e.g. source code and any associated documents) and IC designer proprietary information (tools, software, documentation ...) shall exist and be applied in software development. All authorities involved in the development and production of the TOE shall carry out their development and production activities in a suitable and secure environment. Each party has to ensure that the development and production of the TOE (incl. IC with its Dedicated Software, Smartcard Embedded Software) is secure so that no information is unintentionally made available for the later operational phase of the TOE. In particular, the confidentiality and integrity of design information and test data shall be guaranteed, access to development and test tools, samples and other sensitive material shall be restricted to authorised persons only etc.
Assumptions on the TOE Delivery Process (Phases 4 to 7)	
A.DLV_PROTECT* (PP9911)	**Protection of the TOE under Delivery and Storage** Procedures shall ensure protection of TOE material / information under delivery and storage.
A.DLV_AUDIT* (PP9911)	**Audit of Delivery and Storage** Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process and storage.

A.DLV_RESP* (PP9911)	**Responsibility within Delivery** Procedures shall ensure that people dealing with the procedure for delivery have got the required skill.
Assumptions on Phases 4 to 6	
A.USE_TEST* (PP9911)	**Testing of the TOE** It is assumed that appropriate functionality testing of the TOE is used in phases 4, 5 and 6.
A.USE_PROD* (PP9911)	**Protection of the TOE under Testing and Manufacturing** It is assumed that security procedures are used during all manufacturing and test operations through phases 4, 5, 6 to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).
Assumptions on Phase 6	
A.PERS	**Personalisation Process** The originator of the personalisation data and the personalisation center responsible for the personalisation of the applications running on the TOE handles the personalisation data in an adequate secure manner. This concerns especially the security data to be personalised as secret cryptographic keys and PINs. The storage and processing of the personalisation data at the originator and at the personalisation center as well as the transfer of these data between the different sites is conducted with respect to data integrity and confidentiality. Furthermore, the personalisation center treats the data for securing the personalisation process, i.e. the personalisation keys suitably secure (if applicable). It is in the responsibility of the originator of the personalisation data to garantuee for a sufficient quality of the personalisation data, especially of the cryptographic material to be personalised. The preparation and securing of the personalisation data appropriately for the applications residing on the MICARDO Card is as well in the responsibility of the external world and is done with care.
Assumptions on Phase 7	
A.USE_DIAG* (PP9911)	**Secure Communication** It is assumed that secure communication protocols and procedures are used between smartcard and terminal.

3.2.2 Specific Assumptions for the TOE

There do not exist any specific assumptions for the environment of the TOE.

3.3 Threats

The TOE is required to counter different type of attacks against its specific assets. A threat agent could try to threaten these assets either by functional attacks or by environmental manipulation, by specific hardware manipulation, by a combination of hardware and software manipulations or by any other type of attacks.

Generally, threats can be split into the following types:

- threats against which a specific protection by the TOE is required
- threats against which a specific protection by the environment is required
- threats against which a specific protection by a combination of the TOE and the environment is required

Before listing the general threats for the TOE, several preliminary remarks about these threats:

Threats on phase 1

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

- threats on the TOE-ES and its development environment, such as unauthorized disclosure, modification or theft of the TOE-ES and/or initialisation data
- threats on the assets transmitted from the IC designer to the TOE-ES developer during the TOE-ES development
- threats on the TOE-ES and initialisation data transmitted during the delivery process from the TOE-ES software developer to the IC designer

Furthermore, one can consider the threats under the aspect of disclosure, theft, use or modification:

- Unauthorized disclosure of assets:

  This type of threats covers unauthorized disclosure of assets by attackers who may possess a wide range of technical skills, resources and motivation. Such attackers may also have technical awareness of the product.

- Theft or unauthorized use of assets:

  Potential attackers may gain access to the TOE and perform operations for which they are not authorized. For example, such an attacker may personalize, modify or influence the product in order to gain access to the smartcard application system.

- Unauthorized modification of assets:

The TOE may be subjected to different types of logical or physical attacks which may compromise security. Due to the intended usage of the TOE (the TOE environment may be hostile), the TOE security may be bypassed or compromised reducing the integrity of the TOE security mechanisms and disabling their ability to manage the TOE security. This type of threats includes the implementation of malicious Trojan horses.

Threats on delivery from phase 1 to phases 4 to 6

Threats on data transmitted during the delivery process from the TOE-ES developer to the IC packaging manufacturer, the Finishing process manufacturer or the Personalizer.

Threats on phases 4 to 7

During these phases, the assumed threats could be divided in three types:

- Unauthorized disclosure of assets:

  This type of threats covers unauthorized disclosure of assets by attackers who may possess a wide range of technical skills, resources and motivation. Such attackers may also have technical awareness of the product.

- Theft or unauthorized use of assets:

  Potential attackers may gain access to the TOE and perform operation for which they are not allowed. For example, such attackers may personalize the product in an unauthorized manner, or try to gain fraudulently access to the smartcard system.

- Unauthorized modification of assets:

  The TOE may be subjected to different types of logical or physical attacks which may compromise security. Due to the intended usage of the TOE (the TOE environment may be hostile), the TOE security parts may be bypassed or compromised reducing the integrity of the TOE security mechanisms and disabling their ability to manage the TOE security. This type of threat includes the implementation of malicious Trojan horses, Trapdoors, downloading of viruses or unauthorized programs.

3.3.1 Threats of the IC (TOE-IC)

For the definition of the threats related to the TOE-IC refer to /BSI-PP-0002/, chap. 3.3 and /ST-ICPhilips/, chap. 3.3. Here, only the threats concerning phase 7 of the product life-cycle are considered.

3.3.2 General Threats of the Smartcard Embedded Software (TOE-ES)

The table below lists the general threats to the assets of the TOE-ES against which specific protection within the TOE or its environment is required. Several groups of threats are distinguished according to the means used in the attack and to the phases of the TOE that are affected. The threats to the TOE-ES are defined as indicated in /PP9911/, chap. 3.3.

Note:

In the following table, items of /PP9911/ which are common with /PP9806/ are indicated by a "*".

Threats / TOE-ES (General Threats)	
Name	**Definition**
Threats on all Phases	
T.CLON* (PP9911)	**Cloning of the TOE** Unauthorized full or partional functional cloning of the TOE. Note: This threat is derived from specific threats combining unauthorized disclosure, modification or theft of assets at different phases.
Threats on Phase 1	
T.DIS_INFO* (PP9911)	**Disclosure of IC Assets** Unauthorized disclosure of the assets delivered by the IC designer to the Smartcard Embedded Software developer, such as sensitive information on IC specification, design and technology, software and tools if applicable.
T.DIS_DEL* (PP9911)	**Disclosure of the Smartcard Embedded Software / Application Data during Delivery** Unauthorized disclosure of the Smartcard Embedded Software and any additional application data (such as IC Pre-personalization requirements) during the delivery from the Smartcard Embedded Software developer to the IC designer.
T.DIS_ES1 (PP9911)	**Disclosure of the Smartcard Embedded Software / Application Data within the Development Environment** Unauthorized disclosure of the Smartcard Embedded Software (technical or detailed specifications, implementation code) and/or Application Data (such as secrets, or control parameters for protection system, specification and implementation for security mechanisms) within the development environment.
T.DIS_TEST_ES (PP9911)	**Disclosure of Smartcard Embedded Software Test Programs / Information** Unauthorized disclosure of the the Smartcard Embedded Software test programs or any related information.
T.T_DEL* (PP9911)	**Theft of the Smartcard Embedded Software / Application Data during Delivery** Theft of the Smartcard Embedded Software and any additional application data (such as pre-personalization requirements) during the delivery process from the Smartcard Embedded Software developer to the IC designer.
T.T_TOOLS (PP9911)	**Theft or Unauthorized Use of the Smartcard Embedded Software Development Tools**

	Theft or unauthorized use of the Smartcard Embedded Software development tools (such as PC, development software, data bases).
T.T_SAMPLE2 (PP9911)	**Theft or Unauthorized Use of TOE Samples** Theft or unauthorized use of TOE samples (e.g. bond-out chips with the Smartcard Embedded Software).
T_MOD_DEL* (PP9911)	**Modification of the Smartcard Embedded Software / Application Data during Delivery** Unauthorized modification of the Smartcard Embedded Software and any additional application data (such as IC prepersonalization requirements) during the delivery process from the Smartcard Embedded Software developer to the IC designer.
T.MOD (PP9911)	**Modification of the Smartcard Embedded Software / Application Data within the Development Environment** Unauthorized modification of the Smartcard Embedded Software and/or Application Data or any related information (technical specifications) within the development environment.
Threats on Delivery from Phase 1 to Phases 4 / 5 / 6	
T.DIS_DEL1 (PP9911)	**Disclosure of Application Data during Delivery** Unauthorized disclosure of Application Data during delivery from the Smartcard Embedded Software developer to the IC Packaging manufacturer, the Finishing process manufacturer or the Personalizer.
T.DIS_DEL2 (PP9911)	**Disclosure of Delivered Application Data** Unauthorized disclosure of Application Data delivered from the Smartcard Embedded Software developer to the IC Packaging manufacturer, the Finishing process manufacturer or the Personalizer.
T.MOD_DEL1 (PP9911)	**Modification of Application Data during Delivery** Unauthorized modification of Application Data during delivery from the Smartcard Embedded Software developer to the IC Packaging manufacturer, the Finishing process manufacturer or the Personalizer.
T.MOD_DEL2 (PP9911)	**Modification of Delivered Application Data** Unauthorized modification of Application Data delivered from the Smartcard Embedded Software developer to the IC Packaging manufacturer, the Finishing process manufacturer or the Personalizer.
Threats on Phases 4 to 7	
T.DIS_ES2	**Disclosure of the Smartcard Embedded Software / Application Data**

(PP9911)	Unauthorized disclosure of the Smartcard Embedded Software and Application Data (such as data protection systems, memory partitioning, cryptographic programs and keys).
T.T_ES (PP9911)	**Theft or Unauthorized Use of TOE** Theft or unauthorized use of the TOE (e.g. bound out chips with the Smartcard Embedded Software).
T.T_CMD (PP9911)	**Use of TOE Command-Set** Unauthorized use of instructions or commands or sequence of commands sent to the TOE.
T.MOD_LOAD (PP9911)	**Program Loading** Unauthorized loading of programs.
T.MOD_EXE (PP9911)	**Program Execution** Unauthorized execution of programs.
T.MOD_SHARE (PP9911)	**Modification of Program Behavior** Unauthorized modification of program behavior by interaction of different programs.
T.MOD_SOFT* (PP9911)	**Modification of Smartcard Embedded Software / Application Data** Unauthorized modification of the Smartcard Embedded Software and Application Data.

3.3.3 Specific Threats of the Smartcard Embedded Software (TOE-ES)

The following specific threats of the TOE-ES have to be considered:

Threats / TOE-ES (Specific Threats)	
Name	**Definition**
T.KEYGEN	**RSA Key Pair Generation** An attacker derives the private key from public known data, such as the public key corresponding to the private key.

3.4 Organisational Security Policies of the TOE

The TOE reaches its specific security functionality only by a correct and effective implementation of the underlying IC and its security functionality by the Smartcard Embedded Software (TOE-ES). In particular this means, that the TOE-ES must fulfill the assumptions for the TOE-ES as defined in the Security Target for the TOE-IC.

The relevant assumptions for the TOE-ES as given in /ST-ICPhilips/, chap. 3.2 (refer also to /BSI-PP-0002/, chap. 3.2) are suitably redefined in terms of Organisational Security Policies for the TOE as follows:

Organisational Security Policies for the TOE	
Name	**Definition**
P.Process-Card (A.Process-Card in ST-ICPhilips)	**Protection during Packaging, Finishing and Personalisation** Security procedures shall be used after TOE-IC delivery up to delivery to the end-user to maintain confidentiality and integrity of the TOE-IC and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).
P.Design-Software (A.Plat-Appl, A.Resp-Appl, A.Check-Init, A.Key-Function in ST-ICPhilips)	**Design of the Smartcard Embedded Software** To ensure that the TOE-IC is used in a secure manner the Smartcard Embedded Software (TOE-ES) shall be designed so that the requirements from the following documents are met: - hardware data sheet for the TOE-IC, - TOE-IC application notes, - findings of the TOE-IC evaluation reports relevant for the Smartcard Embedded Software (TOE-ES). Security relevant user data (especially cryptographic keys) are treated by the Smartcard Embedded Software (TOE-ES) as required by the security needs of the specific application context. For example the Smartcard Embedded Software (TOE-ES) will not disclose security relevant user data to unauthorised users or processes when communicating with a terminal. The Smartcard Embedded Software shall provide a function to check initialisation data. The data is defined by the customer and injected by the TOE Manufacturer into the non-volatile memory to provide the possibility for TOE identification and for traceability. Key-dependent functions shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks.

4 Security Objectives

4.1 Security Objectives for the TOE

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

- integrity and confidentiality of the TOE´s assets
- protection of the TOE and its associated documentation and environment during the development and production phases.

4.1.1 Security Objectives for the IC (TOE-IC)

For the definition of the security objectives related to the TOE-IC refer to /BSI-PP-0002/, chap. 4.1 and /ST-ICPhilips/, chap. 4.1. Here, only the security objectives concerning phase 7 of the product life-cycle are considered.

4.1.2 General Security Objectives for the Smartcard Embedded Software (TOE-ES)

Nearly all security objectives mentioned in the table below concern the general security objectives for the TOE-ES as defined in /PP9911/, chap. 4.1. These security objectives are supplemented by security objectives drawn from /BSI-PP-0002/, chap. 4.2 and /ST-ICPhilips/, chap. 4.2, which will be in the current scope switched from assumptions resp. security objectives for the environment of the IC to security objectives for the TOE-ES. The complete set of general security objectives for the TOE-ES is listed in the table below.

Note:

For clarity, within the description of the security objectives in the following table as indicated in /BSI-PP-0002/ and /ST-ICPhilips/ the word „TOE“ is replaced by „TOE-IC“ and the term „Smartcard Embedded Software“ is supplemented by „TOE-ES“.

Note:

In the following table, items of /PP9911/ which are common with /PP9806/ are indicated by a “*”.

Security Objectives / TOE-ES (General Security Objectives)	
Name	**Definition**
O.CLON* (PP9911)	**Cloning** The TOE functionality must be protected from cloning.
O.OPERATE*	**Correct Operation**

(PP9911)	The TOE must ensure continued correct operation of its security functions.
O.FLAW* (PP9911)	**Flaws** The TOE must not contain flaws in design, implementation or operation.
O.DIS_MEMORY* (PP9911)	**Disclosure of Memory Contents** The TOE shall ensure that sensitive information stored in memories is protected against unauthorized disclosure.
O.MOD_MEMORY* (PP9911)	**Modification of Memory Contents** The TOE shall ensure that sensitive information stored in memories is protected against any corruption or unauthorized modification.
O.TAMPER_ES (PP9911)	**Tampering of the Smartcard Embedded Software** The TOE must prevent tampering with its security critical parts. Security mechanisms have especially to prevent the unauthorized change of functional parameters, security attributes and secrets such as the life cycle sequence flags and cryptographic keys. The Smartcard Embedded Software must be designed to avoid interpretations of electrical signals from the hardware part of the TOE.
O.DIS_MECHANISM2 (PP9911)	**Disclosure of Security Mechanisms of the Smartcard Embedded Software** The TOE shall ensure that the Smartcard Embedded Software security mechanisms are protected against unauthorized disclosure.
O.Plat-Appl (OE.Plat-Appl in ST-ICPhilips)	**Usage of Hardware Platform** To ensure that the TOE-IC is used in a secure manner the Smartcard Embedded Software (TOE-ES) shall be designed so that the requirements from the following documents are met: - hardware data sheet for the TOE-IC, - TOE-IC application notes, - findings of the TOE-IC evaluation reports relevant for the Smartcard Embedded Software (TOE-ES).
O.Resp-Appl (OE.Resp-Appl in ST-ICPhilips)	**Treatment of User Data** Security relevant user data (especially cryptographic keys) shall be treated by the Smartcard Embedded Software (TOE-ES) as required by the security needs of the specific application context. For example the Smartcard Embedded Software (TOE-ES) shall not disclose security relevant user data to unauthorised users or processes when communicating with a terminal.
O.Check-Init (OE.Check-Init in ST-ICPhilips)	**Check of initialisation data by the Smartcard Embedded Software** To ensure the receipt of the correct TOE-IC, the Smartcard Embedded Software (TOE-ES) shall provide a function to check initialisation data. The data is defined by the customer and injected by the TOE Manufacturer into the non-volatile memory to provide the possibility for TOE identification and for traceability.
O.Key-Function (A.Key-Function in ST-	**Usage of Key-dependent Functions**

ICPhilips)	Key-dependent functions shall be implemented in the Smartcard Embedded Software (TOE-ES) in a way that they are not susceptible to leakage attacks.

4.1.3 Specific Security Objectives for the Smartcard Embedded Software (TOE-ES)

For the TOE-ES the following specific security objectives are defined:

Security Objectives / TOE-ES (Specific Security Objectives)	
Name	**Definition**
O.KEYGEN	**RSA Key Pair Generation** The TOE shall ensure for high cryptographic quality of the RSA key pair generated oncard. In particular, the generated key pair shall be usable for qualified electronic signatures. The private key can practically occur only once and cannot be reconstructed from the corresponding public key. In that context 'practically occur once' means that the probability of equal private keys is negligible low. In addition, the TOE shall ensure the correspondence between the private and public key.

4.2 Security Objectives for the Environment

4.2.1 General Security Objectives for the Environment of the TOE

Nearly all general security objectives for the environment of the TOE are defined in /PP9911/, chap. 4.2. These security objectives are supplemented by security objectives drawn from /BSI-PP-0002/, chap. 4.2 and /ST-ICPhilips/, chap. 4.2 and a further specific security objective for the personalisation of applications running on the TOE.

All of these security objectives have to be fulfilled by organisational measures, thus they are security objectives for the Non-IT-Environment of the TOE. Security objectives for the IT-Environment of the TOE are not present.

The complete set of security objectives for the environment is listed in the table below.

Note:

For clarity, within the description of the security objectives in the following table as given in /BSI-PP-0002/ and /ST-ICPhilips/ the word „TOE“ is replaced by „TOE-IC“ and the term „Smartcard Embedded Software“ is supplemented by „TOE-ES“.

Note:

In the following table, items of /PP9911/ which are common with /PP9806/ are indicated by a “*”.

Security Objectives for the Environment of the TOE	
Name	**Definition**
Objectives on Phase 1	
O.DEV_TOOLS* (PP9911)	**Development Tools for the Smartcard Embedded Software** The Smartcard Embedded Software shall be designed in a secure manner, by using exclusively software development tools (compilers assemblers, linkers, simulators, etc.) and software-hardware integration testing tools (emulators) that will result in the integrity of program and data.
O.DEV_DIS_ES (PP9911)	**Development of the Smartcard Embedded Software** The Smartcard Embedded Software developer shall use established procedures to control storage and usage of the classified development tools and documentation, suitable to maintain the integrity and the confidentiality of the assets of the TOE. It must be ensured that tools are only delivered and accessible to the parties authorized personnel. It must be ensured that confidential information on defined assets are only delivered to the parties authorized personnel on a need to know basis.
O.SOFT_DLV* (PP9911)	**Protection of the Delivery of the Smartcard Embedded Software** The Smartcard Embedded Software must be delivered from the Smartcard Embedded Software developer (Phase 1) 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, if applicable.
O.INIT_ACS (PP9911)	**Access to Initialisation Data** Initialisation Data shall be accessible only by authorized personnel (physical, personnel, organizational, technical procedures).
O.SAMPLE_ACS (PP9911)	**Access to Samples** Samples used to run tests shall be accessible only by authorized personnel.
Objectives on the TOE Delivery Process (Phases 4 to 7)	

O.DLV_PROTECT* (PP9911)	**Protection of the Delivery of TOE Material / Information**
	
	Procedures shall ensure protection of TOE material / information under delivery including the following objectives:
	- non-disclosure of any security relevant information
	- identification of the element under delivery
	- meet confidentiality rules (confidentiality level, transmittal form, reception acknowledgement)
	- physical protection to prevent external damage
	- secure storage and handling procedures (including rejected TOE's)
	- traceability of TOE during delivery including the following parameters:
	  - origin and shipment details
	  - reception, reception acknowledgement
	  - location material/information

O.DLV_AUDIT* (PP9911)	**Audit of Delivery** Procedures shall ensure that corrective actions are taken in case of improper operation in the delivery process (including if applicable any non-conformance to the confidentiality convention) and highlight all non-conformance to this process.
O.DLV_RESP* (PP9911)	**Responsibility** Procedures shall ensure that people (shipping department, carrier, reception department) dealing with the procedure for delivery have got the required skill, training and knowledge to meet the procedure requirements and be able to act fully in accordance with the above expectations.
Objectives on Delivery from Phase 1 to Phases 4, 5 and 6	
O.DLV_DATA (PP9911)	**Delivery of Application Data** The Application Data must be delivered from the Smartcard Embedded Software developer (phase 1) either to the IC Packaging manufacturer, the Finishing Process manufacturer or the Personalizer through a trusted delivery and verification procedure that shall be able to maintain the integrity and confidentiality of the Application Data.
Objectives on Phases 4 to 6	
O.TEST_OPERATE* (PP9911)	**Testing of the TOE** Appropriate functionality testing of the TOE shall be used in phases 4 to 6. During all manufacturing and test operations, security procedures shall be used through phases 4, 5 and 6 to maintain confidentiality and integrity of the TOE and its manufacturing and test data.
O.Process-Card	**Protection during Packaging, Finishing and Personalisation**

(OE.Process-Card in ST-ICPhilips)	Security procedures shall be used after TOE-IC Delivery up to delivery to the end-user to maintain confidentiality and integrity of the TOE-IC and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).
Objectives on Phase 6	
O.PERS	**Maintaining of Personalisation Data** The originator of the personalisation data and the personalisation center responsible for the personalisation of the applications running on the TOE shall handle the personalisation data in an adequate secure manner. This concerns especially the security data to be personalised as secret cryptographic keys and PINs. The storage and processing of the personalisation data at the originator and at the personalisation center as well as the transfer of these data between the different sites shall be conducted with respect to data integrity and confidentiality. Furthermore, the personalisation center shall treat the data for securing the personalisation process, i.e. the personalisation keys suitably secure (if applicable). It is in the responsibility of the originator of the personalisation data to garantuee for a sufficient quality of the personalisation data, especially of the cryptographic material to be personalised. The preparation and securing of the personalisation data appropriately for the applications residing on the MICARDO Card is as well in the responsibility of the external world and shall be done with care.
Objectives on Phase 7	
O.USE_DIAG* (PP9911)	**Secure Communication** Secure communication protocols and procedures shall be used between the smartcard and the terminal.

4.2.2 Specific Security Objectives for the Environment of the TOE

There do not exist any specific security objectives for the TOE-ES.

5 IT Security Requirements

5.1 TOE Security Requirements

This section covers the subsections “TOE Security Functional Requirements” and ”TOE Security Assurance Requirements”.

5.1.1 TOE Security Functional Requirements

The TOE Security Functional Requirements (SFRs) define the functional requirements for the TOE using functional requirement components drawn from /CC 2.3 Part2/, functional requirement components of /CC 2.3 Part2/ with extension as well as self-defined functional requirement components (only for the IC with its IC Dedicated Software). This chapter considers the SFRs concerning the IC (TOE-IC) as well as the SFRs concerning the Smartcard Embedded Software (TOE-ES).

Note:

The SFRs for the TOE are listed in the following chapters within tables. Thereby, the tables contain in the left column the original definition of the respective SFR and its elements, dependencies, hierarchical information, management and audit functions. The right column supplies the iterations, selections, assignments and refinements chosen for the TOE.

5.1.1.1 TOE Security Functional Requirements for the IC (TOE-IC)

For the definition of the SFRs related to the TOE-IC refer to /BSI-PP-0002/, chap. 5.1.1, 8.4, 8.5, 8.6 and to the Security Target of the IC /ST-ICPhilips/, chap. 5.1.1.

5.1.1.2 TOE Security Functional Requirements for the Smartcard Embedded Software (TOE-ES)

The following section gives a survey of the SFRs related to the TOE´s Smartcard Embedded Software as specified in the Protection Profile /PP9911/. As far as possible and applicable, SFRs defined in the Protection Profiles /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/ and /PP SSCD Type2/ are taken into account as well.

The TOE maintains a Security Function Policy as defined as follows:

SFP Access Rule Policy

Subjects:

- TOE manufacturer
- Personalisation service provider / Personaliser
- Card management service provider / Administrator
- Card holder

Security attributes for subjects:

- USER_GROUP
  (authorised user, non-authorised user)

Objects:

- Files (MF, DF, EF) incl. attributes
- Key objects incl. attributes
- PIN objects incl. attributes

Security attributes for objects:

- Access Rules

Operations (Access Modes):

- MICARDO V3.0 operating system commands

The SFP Access Rule Policy controls the access of subjects to objects on the basis of security attributes.

The TOE maintains the following **type of security attributes**:

- Access Rule (AR) consisting of one or more Partial Access Rules (PAR) whereat each PAR consists of one Access Mode (AM) and one or more Access Conditions (AC)

The AM indicates the command type for accessing the object. The AC defines the conditions under which a command executed by a subject is allowed to access the object.

The access modes to the above mentioned objects are defined above. Further, the TOE maintains the following **types of elementary ACs**:

- NEV (Never)
  The command can never be executed.
- ALW (Always)
  The command can be executed without restrictions.

- AUT (Key based user authentication)
  The right corresponding to a successful external key based authentication must be opened up (done by the command EXTERNAL AUTHENTICATE) before the command can be executed.
- PWD (Password based user authentication)
  The right corresponding to a successful password based authentication must be opened up (done by the command VERIFY) before the command can be executed.
- SM CMD MAC, SM RSP MAC (Secure Messaging providing data integrity and authenticity for command resp. response)
  The command must be secured with a cryptographic checksum using Secure Messaging as defined in /eHC1/ and /HPC-SMC1/.
- SM CMD ENC, SM RSP ENC (Secure Messaging providing data confidentiality for command resp. response)
  The command must be secured with an encryption using Secure Messaging as defined in in /eHC1/ and /HPC-SMC1/.
- OR
  Boolean OR relation.

For rule decisions, the SFP Access Rule Policy uses the actual security status set in the card as reference value.

The SFP Access Rule Policy explicitly authorises access of subjects to objects based on the following rules:

- The TSF allows access to an object for a defined access mode, if the object's access condition is valid for this access mode.
- The TSF evaluates within an AC resp. PAR the logical expression of elementary AC elements according to the following rules:
  - AC element NEV is set to "false".
  - AC element AUT is set to "true", if AUT complies with the actual security status (preceding external authentication has been conducted successfully).
  - AC element SM CMD MAC / SM RSP MAC is set to "true", if SM CMD MAC, SM RSP MAC complies with the user indication for SM CMD MAC, SM RSP MAC and SM CMD MAC, SM RSP MAC complies with the actual security status (preceding external authentication has been conducted successfully).
  - AC element SM CMD ENC, SM RSP ENC is set to "true", if SM CMD ENC, SM RSP ENC complies with the user indication for SM CMD ENC, SM RSP ENC and SM CMD ENC, SM RSP ENC complies with the actual security status (preceding external authentication has been conducted successfully).

FAU **Security Audit**	
FAU_SAA **Security Audit Analysis**	

FAU_SAA.1 **Potential Violation Analysis**	PP 9911
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*]. Hierarchical to: No other components Dependencies: - FAU_GEN.1 Audit data generation Management: a) maintenance of the rules by (adding, modifying, deletion) of rules from the set of rules Audit: a) Minimal: Enabling and disabling of any of the analysis mechanisms b) Minimal: Automated responses performed by the tool	**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 [- **self test error (in particular, integrity error of executable code, application data and operating system specific security critical data)** - **error in the framework of secured data exchange (concerning data integrity and / or data confidentiality)** - **software and / or hardware failure** - **cardholder authentication failure (x consecutive unsuccessful PIN checks)** - **failure in key based authentication operations**] known to indicate a potential security violation; b) [**none**].

FCS **Cryptographic Support**	
FCS_CKM **Cryptographic Key Management**	
FCS_CKM.1 **Cryptographic Key Generation**	PP SSCD Type3 / PP HPC / PPeHC
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*]. Hierarchical to: No other components Dependencies: - [FCS_CKM.2 Cryptographic key distribution	**FCS_CKM.1/RSA-KeyGen:** **FCS_CKM.1.1/RSA-KeyGen** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**RSA key pair generation with randomly generated resp. externally chosen public exponent (up to 64 bit) (command GENERATE ASYMMETRIC KEY PAIR)**] and specified cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **/ALGCAT/, chap. 1.3, 3.1, 4**

or FCS_COP.1 Cryptographic operation] - FCS_CKM.4 Cryptographic key destruction - FMT_MSA.2 Secure security attributes Management: a) the management of changes to cryptographic key attributes. Examples of key attributes include user, key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption) Audit: a) Minimal: Success and failure of the activity b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys)	].
	FCS_CKM.1/DES-KeyGen-Asym: **FCS_CKM.1.1/DES-KeyGen-Asym:** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**3DES session key generation (including SSC) in the framework of an internal-external authentication procedure based on asymmetric cryptography (e.g. asymmetric card-to-card authentication with key agreement)**] and specified cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **/HPC-SMC1/, Annex E.3** - **/eHC1/, Annex E.3** - **/eHC2/, chap. 3.6**].
	FCS_CKM.1/DES-KeyGen-Sym: **FCS_CKM.1.1/DES-KeyGen-Sym** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**3DES session key generation (including SSC) in the framework of an internal-external authentication procedure based on symmetric cryptography (e.g. symmetric card-to-card authentication with key agreement)**] and specified cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **/ANSI X9.63/** - **/HPC-SMC1/, Annex E.4** - **/eHC1/, Annex E.4** - **/eHC2/, chap. 3.7**].
	FCS_CKM.1/KeyDerivation:

	FCS_CKM.1.1/KeyDerivation The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**3DES individual key derivation**] and specified cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **/ISO 9796-2/ (with hash function HF2 according to /ISO 10118-2/)**].
FCS_CKM.2 **Cryptographic Key Distribution**	PP SSCD Type3
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*]. Hierarchical to: No other components Dependencies: - [FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] - FCS_CKM.4 Cryptographic key destruction - FMT_MSA.2 Secure security attributes Management: a) the management of changes to cryptographic key attributes. Examples of key attributes include user, key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption) Audit: a) Minimal: Success and failure of the activity b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys)	**FCS_CKM.2.1** The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [**integrity and authenticity secured export of a public RSA key**] that meets the following: [**MAC secured export of the public key (command GET ATTRIBUTE) or export of a technical RSA signature over the public key (command GENERATE ASYMMETRIC KEY PAIR)**].
FCS_CKM.3 **Cryptographic Key Access**	PP 9911
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 follow-	**FCS_CKM.3.1** The TSF shall perform [**the access to a key for usage within the following cryptographic operation(s)**] in accordance with a specified cryptographic key access method [**access to the key by its key**

ing: [assignment: *list of standards*]. Hierarchical to: No other components Dependencies: [FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] - FCS_CKM.4 Cryptographic key destruction - FMT_MSA.2 Secure security attributes Management: a) the management of changes to cryptographic key attributes; examples of key attributes include user, key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption) Audit: a) Minimal: Success and failure of the activity b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys)	**reference**] that meets the following: [**evaluation of the key reference explicitly or implicitly set before, searching of the referenced key by the TOE specific key search algorithm and evaluation of the related access rules for the key**].
FCS_CKM.4 **Cryptographic Key Destruction**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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*]. Hierarchical to: No other components Dependencies: - [FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] - FMT_MSA.2 Secure security attributes Management: a) the management of changes to cryptographic key attributes. Examples of key attributes include user, key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption)	**FCS_CKM.4/RSA-PrKey-Erasure:** **FCS_CKM.4.1/RSA-PrKey-Erasure** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**erasure of a private RSA key**] that meets the following: [**physical erasure of the key**]. **Application Note** The private RSA key of an RSA key pair generated by the TOE itself shall be destroyed before the RSA key pair is re-generated by the TOE (command GENERATE ASYMMETRIC KEY PAIR).

Audit: a) Minimal: Success and failure of the activity b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys)	
	FCS_CKM.4/DES-Key-Erasure: **FCS_CKM.4.1/DES-Key-Erasure** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**erasure of a 3DES session key**] that meets the following: [**physical erasure of the key**]. **Application Note** The 3DES session key shall be destroyed at the deallocation of the ressource. The TOE shall destroy the 3DES session key used for secure messaging after reset or termination of the secure messaging session or reaching a fail secure state according to FPT_FLS.1.
	FCS_CKM.4/RSA-PubKey-Erasure: **FCS_CKM.4.1/RSA-PubKey-Erasure** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**destruction of a key**] that meets the following: [**overwriting of the old key parameters by the new key parameters**]. **Application Note** In particular, a new key can be imported by setting the new key parameters. Furthermore, a public RSA key can be imported via a CV certificate (command PSO VERIFY CERTIFICATE).
FCS_COP **Cryptographic Operation**	
FCS_COP.1 **Cryptographic Operation**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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*]. Hierarchical to: No other components Dependencies: - [FDP_ITC.1 Import of user data without security	**FCS_COP.1/Exp-RSA-GenDigSig-PKCS1:** **FCS_COP.1.1/Exp-RSA-GenDigSig-PKCS1** The TSF shall perform [**the explicit generation of a digital signature (command PSO COMPUTE DIGITAL SIGNATURE)**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA signature scheme with appendix according to PKCS #1 (based on SHA-1 as hash algorithm): /PKCS1/, chap. 8.2.1 with-**

attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] - FCS_CKM.4 Cryptographic key destruction - FMT_MSA.2 Secure security attributes <u>Management:</u> --- <u>Audit:</u> a) Minimal: Success and failure, and the type of cryptographic operation b) Basic: Any applicable cryptographic mode(s) of operation, subject attributes and object attributes	**out SHA-1 hash value calculation inside step 1 of chap. 9.2; /HPC-SMC1/, chap. 11, /eHC1/, chap. 10**].
	FCS_COP.1/Exp-RSA-GenDigSig-ISO9796-2: **FCS_COP.1.1/Exp-RSA-GenDigSig-ISO9796-2** The TSF shall perform [**the explicit generation of a digital signature (command PSO COMPUTE DIGITAL SIGNATURE)**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA signature scheme with appendix according to ISO/IEC 9796-2 with random number (based on SHA-1 as hash algorithm): /ISO 9796-2/ without SHA-1 hash value calculation; /HPC-SMC1/, chap. 11, /eHC1/, chap. 10**].
	FCS_COP.1/RSA-Corresp: **FCS_COP.1.1/RSA-Corresp** The TSF shall perform [**correspondence verification for oncard generated RSA key pairs**] in accordance with a specified cryptographic algorithm [**generation of an RSA digital signature**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA signature scheme with appendix according to PKCS #1 (based on SHA-1 as hash algorithm): /PKCS1/, chap. 8.2.1 without SHA-1 hash value calculation inside step 1 of chap. 9.2; /HPC-SMC1/, chap. 11, /eHC1/, chap. 10** **or alternatively** - **RSA signature scheme with appendix according to ISO/IEC 9796-2 with random number (based on SHA-1 as hash algorithm): /ISO 9796-2/ without SHA-1 hash value calculation; /HPC-SMC1/, chap. 11,**

	/eHC1/, chap. 10]. **Application Note** The SCD/SVD correspondence verification shall be realised by the generation of a digital signature using the SCD (to be done by the signatory resp. the TOE) followed by the verification of this signature by the external world using the corresponding SVD.
	FCS_COP.1/Imp-RSA-GenDigSig-PKCS1: **FCS_COP.1.1/Imp-RSA-GenDigSig-PKCS1** The TSF shall perform [**the internal authentication with an implicit generation of a digital signature (command INTERNAL AUTHENTICATE)**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA signature scheme according to PKCS #1 without hash and OID, but with an additional limitation of the length of the input message: /PKCS1/, chap. 8.2.1 without steps 1 and 2 in chap. 9.2; /HPC-SMC1/, chap. 11, Annex E.6, /eHC1/, chap. 10, Annex E.6**].
	FCS_COP.1/Imp-RSA-GenDigSig-ISO9796-2: **FCS_COP.1.1/Imp-RSA-GenDigSig-ISO9796-2** The TSF shall perform [**the internal authentication with an implicit generation of a digital signature (command INTERNAL AUTHENTICATE)**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA signature scheme with message recovery according to ISO/IEC 9796-2 (based on SHA-1 as hash algorithm): /ISO 9796-2/; /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3**].
	FCS_COP.1/Exp-RSA-Dec: **FCS_COP.1.1/Exp-RSA-Dec** The TSF shall perform [**the explicit decryption operation (command PSO DECIPHER)**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [

	- **RSA decryption with formatted encoded message according to PKCS #1: /PKCS1/, chap. 7.2.2; /HPC-SMC1/, chap. 11, 4.1, /eHC1/, chap. 10, 3.1.1**].
	FCS_COP.1/RSA-Dec-Primitive: **FCS_COP.1.1/RSA-Dec-Primitive** The TSF shall perform [**the decryption operation**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA decryption primitive according to PKCS #1: /PKCS1/, chap. 5.1.2**].
	FCS_COP.1/Imp-RSA-VerDigSig-ISO9796-2: **FCS_COP.1.1/Imp-RSA-VerDigSig-ISO9796-2** The TSF shall perform [**the external authentication with an implicit verification of a digital signature (command EXTERNAL AUTHENTICATE)**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA signature scheme with message recovery according to ISO/IEC 9796-2 (based on SHA-1 as hash algorithm): /ISO 9796-2/; /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3**].
	FCS_COP.1/Imp-RSA-VerDigSig-CV: **FCS_COP.1.1/Imp-RSA-VerDigSig-CV** The TSF shall perform [**the implicit verification of a digital signature (command PSO VERIFY CERTIFICATE)**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024 bit modulus length**] that meet the following: [- **RSA verification scheme for CV certificates according to ISO/IEC 9796-2 (based on SHA-1 as hash algorithm): /ISO 9796-2/ ; /HPC-SMC1/, Annex B, /eHC1/, Annex B**].
	FCS_COP.1/Exp-RSA-Enc: **FCS_COP.1.1/Exp-RSA-Enc** The TSF shall perform [**the explicit encryption operation (command PSO ENCIPHER)**] in accordance with a specified cryptographic algorithm [**RSA**] and

	cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA encryption with formatted plain message according to PKCS #1: /PKCS1/, chap. 7.2.1 ; /HPC-SMC1/, chap. 11, 4.1, /eHC1/, chap. 10, 3.1.1**].
	FCS_COP.1/RSA-Enc-Primitive: **FCS_COP.1.1/RSA-Enc-Primitive** The TSF shall perform [**the encryption operation**] in accordance with a specified cryptographic algorithm [**RSA**] and cryptographic key sizes [**of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length**] that meet the following: [- **RSA encryption primitive according to PKCS #1: /PKCS1/, chap. 5.1.1**].
	FCS_COP.1/Imp-DES-Enc-MACGen: **FCS_COP.1.1/Imp-DES-Enc-MACGen** The TSF shall perform [**the internal authentication with an implicit encryption and MAC generation (command INTERNAL AUTHENTICATE)**] in accordance with a specified cryptographic algorithm [**3DES**] and cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **3DES CBC encryption with ICV = 0: /FIPS 46-3/, /ANSI X9.52/** - **3DES Retail-MAC generation in CBC resp. CFB mode: /FIPS 46-3/, /ANSI X9.19/** - **/HPC-SMC1/, chap. 11, Annex E.4** - **/eHC1/, chap. 10, Annex E.4**].
	FCS_COP.1/Imp-DES-MACVer-Dec: **FCS_COP.1.1/Imp-DES-MACVer-Dec** The TSF shall perform [**the external authentication with an implicit MAC verification and decryption (command EXTERNAL AUTHENTICATE)**] in accordance with a specified cryptographic algorithm [**3DES**] and cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **3DES CBC decryption with ICV = 0: /FIPS 46-3/, /ANSI X9.52/** - **3DES Retail-MAC generation in CBC resp. CFB mode: /FIPS 46-3/, /ANSI X9.19/** - **/HPC-SMC1/, chap. 11, Annex E.4**

	- **/eHC1/, chap. 10, Annex E.4**].
	FCS_COP.1/Imp-DES-MACVer-Dec-Enc-MACGen: **FCS_COP.1.1/Imp-DES-MACVer-Dec-Enc-MACGen** The TSF shall perform [**the mutual authentication with an implicit MAC verification, decryption, encryption and MAC generation (command MUTUAL AUTHENTICATE)**] in accordance with a specified cryptographic algorithm [**3DES**] and cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **3DES CBC encryption / decryption with ICV = 0: /FIPS 46-3/, /ANSI X9.52/** - **3DES Retail-MAC generation in CBC resp. CFB mode: /FIPS 46-3/, /ANSI X9.19/** - **/HPC-SMC1/, chap. 11, Annex E.4** - **/eHC1/, chap. 10, Annex E.4**].
	FCS_COP.1/DES-Enc-Dec: **FCS_COP.1.1/DES-Enc-Dec** The TSF shall perform [**the encryption and decryption operation (command PSO ENCIPHER, PSO DECIPHER, Secure Messaging)**] in accordance with a specified cryptographic algorithm [**3DES**] and cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **3DES CBC encryption with ICV = 0 and ISO-Padding: /FIPS 46-3/, /ANSI X9.52/, /ISO 7816-4/, /HPC-SMC1/, chap. 11, 4.1, /eHC1/, chap. 10, 3.1.1**].
	FCS_COP.1/DES-MACGen-MACVer: **FCS_COP.1.1/DES-MACGen-MACVer** The TSF shall perform [**the MAC generation and verification (command PSO COMPUTE CRYPTOGRAPHIC CHECKSUM, command PSO VERIFY CRYPTOGRAPHIC CHECKSUM, Secure Messaging)**] in accordance with a specified cryptographic algorithm [**3DES**] and cryptographic key sizes [**of 128 bit (with keying option 2 and 112 bit entropy, no parity bits set)**] that meet the following: [- **3DES Retail-MAC generation in CBC resp. CFB mode: /FIPS 46-3/, /ANSI X9.19/**].
	FCS_COP.1/SHA1: **FCS_COP.1.1/SHA1**

	The TSF shall perform [**cryptographic checksum generation (command PSO HASH)**] in accordance with the specified cryptographic algorithm [**SHA-1**] and cryptographic key sizes [**none**] that meet the following: [- **standard FIPS 180-1 resp. FIPS 180-2**].
FCS_RND **Generation of Random Numbers**	
FCS_RND.1 **Quality Metric for Random Numbers**	PP HPC / PP eHC
FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet [assignment: a defined quality metric]. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: ---	**FCS_RND.1.1** The TSF shall provide a mechanism to generate random numbers that meet [**deterministic RNG of quality class K4**].

FDP **User Data Protection**	
FDP_ACC **Access Control Policy**	
FDP_ACC.2 **Complete Access Control**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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. Hierarchical to:	**FDP_ACC.2.1** The TSF shall enforce the [**SFP Access Rule Policy**] on [**all subjects and objects defined by the SFP Access Rule Policy**] 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.

FDP_ACC.1 Dependencies: - FDP_ACF.1 Security attribute based access control Management: --- Audit: ---	
FDP_ACF **Access Control Functions**	
FDP_ACF.1 **Security Attribute Based Access Control**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
FDP_ACF.1.1 The TSF shall enforce the [assignment: *access control SFP*] to objects based on the following: [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]. **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*]. Hierarchical to: No other components Dependencies: - FDP_ACC.1 Subset access control - FMT_MSA.3 Static attribute initialisation Management: a) Managing the attributes used to make explicit access or denial based decisions Audit:	**FDP_ACF.1.1** The TSF shall enforce the [**SFP Access Rule Policy**] to objects based on the following: [**all subjects and objects together with their respective security attributes as defined in the SFP Access Rule Policy**]. **FDP_ACF.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**rules for all access methods and the access rules defined in the SFP Access Rule Policy**]. **FDP_ACF.1.3** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**none**]. **FDP_ACF.1.4** The TSF shall explicitly deny access of subjects to objects based on the [**rules for all access methods and the access rules defined in the SFP Access Rule Policy**].

a) Minimal: Successful requests to perform an operation on an object covered by the SFP b) Basic: All requests to perform an operation on an object covered by the SFP c) Detailed: The specific security attributes used in making an access check	
FDP_DAU **Data Authentication**	
FDP_DAU.1 **Basic Data Authentication**	PP 9911
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. Hierarchical to: No other components Dependencies: No dependencies Management: a) The assignment or modification of the objects for which data authentication may apply could be configurable in the system Audit: a) Minimal: Successful generation of validity evidence b) Basic: Unsuccessful generation of validity evidence c) Detailed: The identity of the subject that requested the evidence	**FDP_DAU.1.1** The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [**a public RSA key generated by the TOE**]. **FDP_DAU.1.2** The TSF shall provide [**external world**] with the ability to verify evidence of the validity of the indicated information. **Note** The TOE generates on demand of the external world a digital signature over the public key of the RSA key pair generated by the TOE. This signature is output to the external world and can be proved by usage of the public key itself corresponding to the TOE´s private signature key which has been used for the signature creation. Alternatively, the public key of the RSA key pair generated by the TOE can be exported together with a cryptographic checksum over the key data.
FDP_ETC **Export to Outside TSF Control**	
FDP_ETC.1 **Export of User Data without Security Attributes**	PP 9911 / PP SSCD Type3 / PP SSCD Type2
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.	**FDP_ETC.1.1** The TSF shall enforce the [**SFP Access Rule Policy**] 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.

Hierarchical to: No other components Dependencies: - [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] Management: --- Audit: a) Minimal: Successful export of information b) Basic: All attempts to export information	
FDP_ITC **Import from Outside TSF Control**	
FDP_ITC.1 **Import of User Data without Security Attributes**	PP 9911 / PP SSCD Type3 / PP SSCD Type2
FDP_ITC.1.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when importing user data, controlled under the SFP, from outside of the 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*]. Hierarchical to: No other components Dependencies: - [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] - FMT_MSA.3 Static attribute initialisation Management: a) The modification of the additional control rules used for import Audit: a) Minimal: Successful import of user data, including any security attributes b) Basic: All attempts to import user data, including any security attributes c) Detailed: The specification of security attributes for imported user data supplied by an authorised user	**FDP_ITC.1.1** The TSF shall enforce the [**SFP Access Rule Policy**] 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: [**rules for all access methods and the access rules defined in the SFP Access Rule Policy**].

FDP_RIP **Residual Information Protection**	
FDP_RIP.1 **Subset Residual Information Protection**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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*]. Hierarchical to: No other components Dependencies: No dependencies Management: a) The choice of when to perform residual information protection (i.e. upon allocation or deallocation) could be made configurable within the TOE Audit: ---	**FDP_RIP.1.1** The TSF shall ensure that any previous information content of a resource is made unavailable upon the [**deallocation of the resource from**] the following objects: [**security relevant material (as secret and private cryptographic keys, PINs, PUCs, data in all files which are not freely accessible, ...)**].
FDP_SDI **Stored Data Integrity**	
FDP_SDI.2 **Stored Data Integrity Monitoring and Action**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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*]. Hierarchical to: FDP_SDI.1 Dependencies: No dependencies Management: a) The actions to be taken upon the detection of an integrity error could be configurable Audit: a) Minimal: Successful attempts to check the integrity of user data, including an indication of the results of	**FDP_SDI.2/Int-PersData:** **FDP_SDI.2.1/Int-PersData** The TSF shall monitor user data **and specific TSF data** stored within the TSC for [**integrity errors**] on all objects, based on the following attributes: [**checksum secured persistently stored data**]. **Application Note** The following data persistently stored by the TOE have the attribute „checksum secured persistently stored data“: - User / application data (e.g. in files of the card) - Keys (incl. attributes) - PINs / PUCs (incl. attributes) - File and object management information (as e.g. access rules, object life cycle states) - Card life cycle status information **Refinement** The check for integrity errors shall be done before usage resp. processing of the data. The checksum

the check b) Basic: All attempts to check the integrity of user data, including an indication of the results of the check, if performed c) Detailed: The type of integrity error that occurred d) Detailed: The action taken upon detection of an integrity error	securing shall concern the data objects as well as the data values themselves. **FDP_SDI.2.2/Int-PersData** Upon detection of a data integrity error, the TSF shall [- **warn the entity connected about the integrity error** - **prohibit the usage resp. processing of the altered data**].
	FDP_SDI.2/Int-TempData: **FDP_SDI.2.1/Int-TempData** The TSF shall monitor user data **and specific TSF data** stored within the TSC for [**integrity errors**] on all objects, based on the following attributes: [**checksum secured temporarily stored data**]. **Application Note** The following data temporarily stored by the TOE have the attribute „checksum secured temporarily stored data“: - User / application data (as hash values, ...) - Keys (incl. attributes) - Card Context including different Channel Contexts (actual Security Environment, status information as the actual security status for Key and PIN based authentication, information on the availability of session keys, ...) - Input data for electronic signatures **Refinement** The check for integrity errors shall be done before usage resp. processing of the data. The checksum securing shall concern the data objects as well as the data values themselves. **FDP_SDI.2.2/Int-TempData** Upon detection of a data integrity error, the TSF shall [- **warn the entity connected about the integrity error** - **prohibit the usage resp. processing of the altered data**].
FDP_UCT **Inter-TSF User Data Confidentiality Transfer Protection**	
FDP_UCT.1 **Basic Data Exchange Confidentiality**	PP HPC / PP eHC
FDP_UCT.1.1 The TSF shall enforce the [assignment: *access con-*	**FDP_UCT.1.1** The TSF shall enforce the [**SFP Access Rule Policy**]

trol SFP(s) and/or information flow control SFP(s)] to be able to [selection*: transmit, receive*] objects in a manner protected from unauthorised disclosure. Hierarchical to: No other components Dependencies: - [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] - [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] Management: --- Audit: a) Minimal: The identity of any user or subject using the data exchange mechanisms b) Basic: The identity of any unauthorised user or subject attempting to use the data exchange mechanisms c) Basic: A reference to the names or other indexing information useful in identifying the user data that was transmitted or received. This could include security attributes associated with the information	to be able to [**transmit and receive**] objects in a manner protected from unauthorised disclosure.
FDP_UIT **Inter-TSF User Data Integrity Transfer Protection**	
FDP_UIT.1 **Data Exchange Integrity**	PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
FDP_UIT.1.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to be able to [selection*: transmit, receive*] user data in a manner protected from [selection: *modification, deletion, insertion, replay*] errors. **FDP_UIT.1.2** The TSF shall be able to determine on receipt of user data, whether [selection: *modification*, *deletion, insertion, replay*] has occurred. Hierarchical to: No other components Dependencies: - [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] - [FTP_ITC.1 Inter-TSF trusted channel or FTP_TRP.1 Trusted path]	**FDP_UIT.1.1** The TSF shall enforce the [**SFP Access Rule Policy**] to be able to [**transmit and receive**] user data in a manner protected from [**modification, deletion, insertion and replay**] errors. **FDP_UIT.1.2** The TSF shall be able to determine on receipt of user data, whether [**modification, deletion, insertion and replay**] has occurred.

Management: --- Audit: a) Minimal: The identity of any user or subject using the data exchange mechanisms b) Basic: The identity of any user or subject attempting to use the user data exchange mechanisms, but who is unauthorised to do so c) Basic: A reference to the names or other indexing information useful in identifying the user data that was transmitted or received; this could include security attributes associated with the user data d) Basic: Any identified attempts to block transmission of user data e) Detailed: The types and/or effects of any detected modifications of transmitted user data	

FIA **Identification and Authentication**	
FIA_AFL **Authentication Failures**	
FIA_AFL.1 **Authentication Failure Handling**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
FIA_AFL.1.1 The TSF shall detect when [selection: [assignment: *positive integer number*], "*an administrator configurable positive integer within* [assignment: *range of acceptable values*]"] 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*]. Hierarchical to: No other components Dependencies: - FIA_UAU.1 Timing of authentication Management: a) management of the threshold for unsuccessful authentication attempts b) management of actions to be taken in the event of an authentication failure Audit: a) Minimal: the reaching of the threshold for the un-	**FIA_AFL.1/PIN:** **FIA_AFL.1.1/PIN** The TSF shall detect when [**predefined value in the range 0 – 255, "an administrator configurable positive integer within [0 and 255]"**] unsuccessful authentication attempts occur related to [**consecutive failed PIN based authentication (human user authentication) with the referenced PIN**]. **FIA_AFL.1.2/PIN** When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [- **warn the entity connected** - **not set the actual security state for the PIN** - **block the PIN resp. the verification mechanism for this PIN such that any subsequent authentication attempt with this PIN will fail (until successful unblock with resetting code)** - **be able to indicate to subsequent users the reason for the blocking of the PIN**].

successful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal)	
	FIA_AFL.1/PUC: **FIA_AFL.1.1/PUC** The TSF shall detect when [**predefined value in the range 0 – 255, "an administrator configurable positive integer within [0 and 255]"**] **successful or** unsuccessful authentication attempts occur related to [**PUC based authentication for resetting a referenced blocked PIN (usage of unblocking code)**]. **FIA_AFL.1.2/PUC** When the defined number of **successful or** unsuccessful authentication attempts has been met or surpassed, the TSF shall [- **warn the entity connected** - **not unblock the referenced blocked PIN** - **block the PUC resp. the verification mechanism for this PUC such that any subsequent authentication attempt with this PUC will fail and an unblocking of all blocked PINs related to this PUC is no longer possible** - **be able to indicate to subsequent users the reason for the blocking of the PUC**].
	FIA_AFL.1/KeyUsage: **FIA_AFL.1.1/KeyUsage** The TSF shall detect when [**predefined value in the range 0 – 65334, "an administrator configurable positive integer within [0 and 65334]"**] **successful or** unsuccessful authentication attempts occur related to [**key based authentication with the referenced key**]. **FIA_AFL.1.2/KeyUsage** When the defined number of **successful or** unsuccessful authentication attempts has been met or surpassed, the TSF shall [- **warn the entity connected** - **not set the actual security state for the key** - **block the key resp. the authentication mechanism for this key such that any subsequent authentication attempt with this key will fail** - **be able to indicate to subsequent users the reason for the blocking of the key**].
FIA_ATD	

User Attribute Definition	
FIA_ATD.1 **User Attribute Definition**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: *list of security attributes*]. Hierarchical to: No other components Dependencies: No dependencies Management: a) if so indicated in the assignment, the authorised administrator might be able to define additional security attributes for users Audit: ---	**FIA_ATD.1.1** The TSF shall maintain the following list of security attributes belonging to individual users: [**identity and role**].
FIA_UAU **User Authentication**	
FIA_UAU.1 **Timing of Authentication**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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. Hierarchical to: No other components Dependencies: - FIA_UID.1 Timing of identification Management: a) management of the authentication data by an administrator b) management of the authentication data by the associated user c) managing the list of actions that can be taken before the user is authenticated Audit: a) Minimal: Unsuccessful use of the authentication mechanism b) Basic: All use of the authentication mechanism	**FIA_UAU.1.1** The TSF shall allow [**reading the ATR, execution of commands allowed without preceding successful authentication due to the access rules set**] 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.

c) Detailed: All TSF mediated actions performed before authentication of the user	
FIA_UAU.3 **Unforgeable Authentication**	PP 9911
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. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: a) Minimal: Detection of fraudulent authentication data b) Basic: All immediate measures taken and results of checks on the fraudulent data	**FIA_UAU.3.1** The TSF shall [**detect**] use of authentication data that has been forged by any user of the TSF. **FIA_UAU.3.2** The TSF shall [**detect**] use of authentication data that has been copied from any other user of the TSF.
FIA_UAU.4 **Single-use Authentication Mechanisms**	PP 9911 / PP HPC / PP eHC
FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [assignment: *identified authentication mechanism(*s)]. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: a) Minimal: Attempts to reuse authentication data	**FIA_UAU.4.1** The TSF shall prevent reuse of authentication data related to [**key based authentication mechanisms**].
FIA_UAU.6 **Re-Authenticating**	PP HPC
FIA_UAU.6.1	**FIA_UAU.6.1**

The TSF shall re-authenticate the user under the conditions [assignment: *list of conditions under which re-authentication is required*]. Hierarchical to: No other components Dependencies: No dependencies Management: a) if an authorised administrator could request re-authentication, the management includes a re-authentication request. Audit: a) Minimal: Failure of reauthentication b) Basic: All reauthentication attempts	The TSF shall re-authenticate the user under the conditions [**successful established secure messaging**].
FIA_UID **User Identification**	
FIA_UID.1 **Timing of Identification**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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 successfully identified before allowing any other TSF-mediated actions on behalf of that user. Hierarchical to: No other components Dependencies: No dependencies Management: a) the management of the user identities b) if an authorised administrator can change the actions allowed before identification, the managing of the action lists Audit: a) Minimal: Unsuccessful use of the user identification mechanism, including the user identity provided b) Basic: All use of the user identification mechanism, including the user identity provided	**FIA_UID.1.1** The TSF shall allow [**reading the ATR, execution of commands with access rule set to Always**] 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 successfully identified before allowing any other TSF-mediated actions on behalf of that user.
FIA_USB **User-Subject Binding**	
FIA_USB.1	PP 9911

User-Subject Binding	
FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: *list of user security attributes*]. **FIA_USB.1.2** The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *rules for the initial association of attributes*]. **FIA_USB.1.3** The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: *rules for the changing of attributes*]. Hierarchical to: No other components Dependencies: - FIA_ATD.1 User attribute definition Management: a) an authorised administrator can define default subject security attributes b) an authorised administrator can change subject security attributes Audit: a) Minimal: Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject) b) Basic: Success and failure of binding of user security attributes to a subject (e.g. success or failure to create a subject)	**FIA_USB.1.1** The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**USER_GROUP (authorised user, not-authorised user)**]. **FIA_USB.1.2** The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**assignment in the framework of the TOE´s access rule mechanism**]. **FIA_USB.1.3** The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [**change possible only for authorised users**].

FMT **Security Management**	
FMT_LIM **Limited capabilities and availability**	
FMT_LIM.1 **Limited capabilities**	PP HPC / PP eHC
FMT_LIM.1.1 The TSF shall be designed in a manner that limits their capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced [assignment: *Limited capability and availability policy*].	**FMT_LIM.1.1** The TSF shall be designed in a manner that limits their capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced [**Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated,**

Hierarchical to: No other components Dependencies: - FMT_LIM.2 Limited availability Management: --- Audit: ---	**software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks**].
FMT_LIM.2 **Limited availability**	PP HPC / PP eHC
FMT_LIM.2.1 The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced [assignment: *Limited capability and availability policy*]. Hierarchical to: No other components Dependencies: - FMT_LIM.1 Limited capability Management: --- Audit: ---	**FMT_LIM.2.1** The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced [**Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks**].
FMT_MOF **Management of Functions in TSF**	
FMT_MOF.1 **Management of Security Functions Behaviour**	PP 9911 / PP SSCD Type3 / PP SSCD Type2
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*]. Hierarchical to: No other components Dependencies: - FMT_SMF.1 Specification of management functions - FMT_SMR.1 Security roles Management: a) managing the group of roles that can interact with the functions in the TSF	**FMT_MOF.1.1** The TSF shall restrict the ability to [**modify the behaviour of**] the functions [**modify access rules, modify object attributes**] to [**authorised users**].

Audit: a) Basic: All modifications in the behaviour of the functions in the TSF	
FMT_MSA **Management of Security Attributes**	
FMT_MSA.1 **Management of Security Attributes**	PP 9911 / PP SSCD Type3 / PP SSCD Type2
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*]. Hierarchical to: No other components Dependencies: - [FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] - FMT_SMF.1 Specification of management functions - FMT_SMR.1 Security roles Management: a) managing the group of roles that can interact with the security attributes Audit: a) Basic: All modifications of the values of security attributes	**FMT_MSA.1.1** The TSF shall enforce the [**SFP Access Rule Policy**] to restrict the ability to [**modify, delete, insert, write**] the security attributes [**access rules**] to [**authorised users**].
FMT_MSA.2 **Secure Security Attributes**	PP 9911 / PP SSCD Type3 / PP SSCD Type2
FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes. Hierarchical to: No other components 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 Management:	**FMT_MSA.2.1** The TSF shall ensure that only secure values are accepted for security attributes.

--- Audit: a) Minimal: All offered and rejected values for a security attribute b) Detailed: All offered and accepted secure values for a security attribute	
FMT_MSA.3 **Static Attribute Initialisation**	PP 9911 / PP SSCD Type3 / PP SSCD Type2
FMT_MSA.3.1 The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to provide [selection: *choose one of: restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP. **FMT_MSA.3.2** 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. Hierarchical to: No other components Dependencies: - FMT_MSA.1 Management of security attributes - FMT_SMR.1 Security roles Management: a) managing the group of roles that can specify initial values b) managing the permissive or restrictive setting of default values for a given access control SFP Audit: a) Basic: Modifications of the default setting of permissive or restrictive rules b) Basic: All modifications of the initial values of security attributes	**FMT_MSA.3.1** The TSF shall enforce the [**SFP Access Rule Policy**] to provide [**permissive**] default values for security attributes that are used to enforce the SFP. **FMT_MSA.3.2** The TSF shall allow the [**authorised user**] to specify alternative initial values to override the default values when an object or information is created.
FMT_MTD **Management of TSF Data**	
FMT_MTD.1 **Management of TSF Data**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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*]. Hierarchical to:	**FMT_MTD.1/General** **FMT_MTD.1.1/General** The TSF shall restrict the ability to [**change_default, modify, delete, insert, write**] the [**objects**] to [**authorised users**].

No other components Dependencies: - FMT_SMF.1 Specification of management functions - FMT_SMR.1 Security roles Management: a) managing the group of roles that can interact with the TSF data Audit: a) Basic: All modifications to the values of TSF data	
	FMT_MTD.1/Init **FMT_MTD.1.1/Init** The TSF shall restrict the ability to [**write**] the [**initialisation data**] to [**TOE manufacturer**].
FMT_SMF **Specification of Management Functions**	
FMT_SMF.1 **Specification of Management Functions**	PP HPC / PP eHC
FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [assignment: *list of security management functions to be provided by the TSF*]. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: a) Minimal: Use of the management functions.	**FMT_SMF.1.1** The TSF shall be capable of performing the following security management functions: [**initialisation, personalisation, card management, management of card objects**].
FMT_SMR **Security Management Roles**	
FMT_SMR.1 **Security Roles**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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.	**FMT_SMR.1.1** The TSF shall maintain the roles [**TOE manufacturer, personalisation service provider / personaliser, card management service provider / administrator, card holder**].

Hierarchical to: No other components Dependencies: - FIA_UID.1 Timing of identification Management: a) managing the group of users that are part of a role Audit: a) Minimal: modifications to the group of users that are part of a role b) Detailed: every use of the rights of a role	**FMT_SMR.1.2** The TSF shall be able to associate users with roles.

FPR **Privacy**	
FPR_UNO **Unobservability**	
FPR_UNO.1 **Unobservability**	PP 9911
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*]. Hierarchical to: No other components Dependencies: No dependencies Management: a) the management of the behaviour of the unobservability function Audit: a) Minimal: The invocation of the unobservability mechanism	**FPR_UNO.1/Sec-CryptoOp:** **FPR_UNO.1.1/Sec-CryptoOp** The TSF shall ensure that [**users**] are unable to observe the operation [**cryptographic operations based on symmetric or asymmetric cryptography as e.g. calculation of a digital signature, key based authentication processes, data processing in the framework of Secure Messaging, ...**] on [**user data, keys**] by [**users, TOE internal processes**].
	FPR_UNO.1/Sec-PINOp: **FPR_UNO.1.1/Sec-PINOp** The TSF shall ensure that [**users**] are unable to observe the operation [**PIN based authentication**] on [**PIN**] by [**users, TOE internal processes**].

FPT **Protection of the TSF**	
FPT_AMT **Underlying Abstract Machine Test**	
FPT_AMT.1 **Abstract Machine Testing**	PP SSCD Type3 / PP SSCD Type2
FPT_AMT.1.1 The TSF shall run a suite of tests [selection: *during initial start-up, periodically during normal operation, at the request of an authorised user, other conditions*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF. Hierarchical to: No other components Dependencies: No dependencies Management: a) management of the conditions under which abstract machine test occurs, such as during initial start-up, regular interval, or under specified conditions b) management of the time interval if appropriate Audit: a) Basic: Execution of the tests of the underlying machine and the results ofthe tests	**FPT_AMT.1.1** The TSF shall run a suite of tests [**during initial start-up, periodically during normal operation**] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF. **Application Note** The test of the underlying abstract machine is performed in the framework of the self test functionality of the TOE (refer to SFR FPT_TST.1).
FPT_EMSEC **TOE Emanation**	
FPT_EMSEC.1 **TOE Emanation**	PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
FPT_EMSEC.1.1 The TOE shall not emit [assignment: types of emissions] in excess of [assignment: specified limits] enabling access to [assignment: list of types of TSF data] and [assignment: list of types of user data]. **FPT_EMSEC.1.2** The TSF shall ensure [assignment: type of users] are unable to use the following interface [assignment: type of connection] to gain access to [assignment: list of types of TSF data] and [assignment: list of types of user data]. Hierarchical to: No other components	**FPT_EMSEC.1.1** The TOE shall not emit [**information on IC power consumption, information on command execution time, information on electromagnetic emanations**] in excess of [**non useful information**] enabling access to [**security critical data as PINs and PUCs**] and [**security critical data as cryptographic keys**]. **FPT_EMSEC.1.2** The TSF shall ensure [**any user**] are unable to use the following interface [**IC contacts as Vcc, I/O and GND, IC surface**] to gain access to [**security critical data as PINs and PUCs**] and [**security critical data as cryptographic keys**].

Dependencies: No dependencies Management: --- Audit: ---	**Application Note** The TOE shall prevent attacks against secret data where the attack is based on external observable physical phenomena of the TOE. Such attacks may be observable at the interfaces of the TOE or may origin from internal operation of the TOE or may origin by an attacker that varies the physical environment under which the TOE operates. The set of measurable physical phenomena is influenced by the technology employed to implement the TOE. Examples of measurable phenomena are variations in the power consumption, the timing of transitions of internal states, electromagnetic radiation due to internal operation, radio emission. Due to the heterogeneous nature of the technologies that may cause such emanations, evaluation against state-of-the-art attacks applicable to the technologies employed by the TOE is assumed. Examples of such attacks are, but are not limited to, evaluation of TOE's electromagnetic radiation**,** simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc.
FPT_FLS **Fail Secure**	
FPT_FLS.1 **Failure with Preservation of Secure State**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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*]. Hierarchical to: No other components Dependencies: - ADV_SPM.1 Informal TOE security policy model Management: --- Audit: a) Basic: Failure of the TSF	**FPT_FLS.1.1** The TSF shall preserve a secure state when the following types of failures occur: [- **HW and/or SW induced reset** - **Power supply cut-off or variations** - **Unexpected abortion of the execution of the TSF due to external or internal events (in particular, break of a transaction before completion)** - **System breakdown** - **Internal HW and/or SW failure** - **Manipulation of executable code** - **Corruption of status information (as e.g. card status information, object life cycle state, actual security state related to key and PIN based authentication, ...)** - **Environmental stress** - **Input of inconsistent or improper data** - **Tampering** - **Manipulation resp. insufficient quality of the HW-RNG resp. SW-RNG** - **Fault injection attacks** - **Exposure to operating conditions where therefore a malfunction could occur** - **Failure detected by TSF according to FPT_TST.1**].

	Refinements The TOE shall preserve a secure state during power supply cut-off or variations. If power is cut or if power variations occur from the TOE, or if a transaction is stopped before completion, or on any other reset conditions, the TOE shall be reset cleanly.
FPT_PHP **Physical Protection**	
FPT_PHP.1 **Passive Detection of Physical Attack**	PP SSCD Type3 / PP SSCD Type2
FPT_PHP.1.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF. **FPT_PHP.1.2** The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: a) Minimal: if detection by IT means, detection of intrusion.	**FPT_PHP.1.1** The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF. **FPT_PHP.1.2** The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.
FPT_PHP.3 **Resistance to Physical Attack**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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. Hierarchical to: No other components Dependencies: No dependencies Management: a) management of the automatic responses to physical tampering Audit:	**FPT_PHP.3.1** The TSF shall resist [**physical manipulation and physical probing (e.g. tampering of the specified physical and technical operating conditions of the IC as voltage supply, clock frequency and temperature out of the valid limits)**] to the [**TSF**] by responding automatically such that the TSP is not violated.

FPT_RVM **Reference Mediation**	
FPT_RVM.1 **Non-Bypassability of the TSP**	PP HPC / PP eHC
FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: ---	**FPT_RVM.1.1** The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
FPT_SEP **Domain Separation**	
FPT_SEP.1 **TSF Domain Separation**	PP 9911 / PP HPC / PP eHC
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. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: ---	**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. **Note** In particular, this SFR concerns the logical channel functionality of the TOE. Each logical channel shall maintain its own channel context whereat each channel context shall have no access to the other channel contexts existing.
FPT_TDC **Inter-TSF TSF Data Consistency**	

FPT_TDC.1 **Inter-TSF Basic TSF Data Consistency**	PP 9911
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. Hierarchical to: No other components Dependencies: No dependencies Management: --- Audit: a) Minimal: Successful use of TSF data consistency mechanisms b) Basic: Use of the TSF data consistency mechanisms c) Basic: Identification of which TSF data have been interpreted d) Basic: Detection of modified TSF data	**FPT_TDC.1.1** The TSF shall provide the capability to consistently interpret [- **PINs and their reference format** - **Keys resp. key pairs** - **Authentication tokens (with their input data for session keys and SSC, if applicable)** - **Imported CV certificates with public RSA keys** - **Other imported / exported cryptographically secured data**] when shared between the TSF and another trusted IT product. **FPT_TDC.1.2** The TSF shall use [- **Rules for the interpretation of PINs supplied for PIN verification: F2B PIN block format** - **Rules for the interpretation of authentication tokens: /ISO 9796-2/, /PKCS1/, /FIPS 46-3/, /ANSI X9.52/, /ANSI X9.19/, /HPC-SMC1/, /eHC1/** - **Rules for the generation of session keys and SSCs from data within authentication tokens: /ANSI X9.63/, /HPC-SMC1/, /eHC1/** - **Rules for the derivation of individual keys: /ISO 9796-2/** - **Rules for the interpretation of imported CV certificates, their format and their included signature: /ISO 9796-2/, /HPC-SMC1/, /eHC1/** - **Rules for the interpretation of other imported / exported cryptographically secured data (e.g. within the framework of Secure Messaging): /ISO 7816-4/, /PKCS1/, /ISO 9796-2/, /PKCS1/, /FIPS 46-3/, /ANSI X9.52/, /ANSI X9.19/, /HPC-SMC1/, /eHC1/**] when interpreting the TSF data from another trusted IT product.
FPT_TST **TSF Self Test**	
FPT_TST.1 **TSF Testing**	PP 9911 / PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
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 con-*	**FPT_TST.1.1** The TSF shall run a suite of self tests [**during initial start-up, periodically during normal operation**] to demonstrate the correct operation of [**the TSF**].

ditions [assignment: *conditions under which self test should occur*]] to demonstrate the correct operation of [selection: [assignment: *parts of TSF*], *the TSF*]. **FPT_TST.1.2** The TSF shall provide authorised users with the capability to verify the integrity of [selection: [assignment: *parts of TSF data*], *TSF data*]. **FPT_TST.1.3** The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code. Hierarchical to: No other components Dependencies: - FPT_AMT.1 Abstract machine testing Management: a) management of the conditions under which TSF self testing occurs, such as during initial start-up, regular interval, or under specified conditions b) management of the time interval if appropriate Audit: a) Basic: Execution of the TSF self tests and the results of the tests	**Note** During initial start-up means *before* code execution. **Refinements** The TOE's self tests shall include the verification of the integrity of any software code (incl. patches) stored outside of the ROM. Upon detection of a self test error the TSF shall warn the entity connected. After OS testing is completed, all testing-specific commands and actions shall be disabled or removed. It shall not be possible to override these controls and restore them for use. Command associated exclusively with one life cycle state shall never be accessed during another state. **FPT_TST.1.2** The TSF shall provide authorised users with the capability to verify the integrity of [**TSF data**]. **Refinement** In this framework, the OS (i.e. the Smartcard Embedded Software of the TOE (TOE-ES)) itself is understood as „authorised user“. **FPT_TST.1.3** The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code. **Refinement** The integrity check over the executable code stored outside the ROM area is covered by FPT_TST.1.1 and the related refinement. The requirement for checking the integrity of the ROM-code shall concern only the production phase, more precise the initialisation phase of the TOE´s lifecycle. Prior to the initialisation of the TOE, the ROM-code of the TOE shall be verifiable by authorised users as the OS developer. The integrity of the ROM-code shall be provable only during the initialisation process.

FTP **Trusted Path/Channels**	
FTP_ITC **Inter-TSF Trusted Channel**	
FTP_ITC.1 **Inter-TSF Trusted Channel**	PP SSCD Type3 / PP SSCD Type2 / PP HPC / PP eHC
FTP_ITC.1.1 The TSF shall provide a communication channel	**FTP_ITC.1.1** The TSF shall provide a communication channel be-

between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. **FTP_ITC.1.2** The TSF shall permit [selection: *the TSF, the remote trusted IT product*] to initiate communication via the trusted channel. **FTP_ITC.1.3** The TSF shall initiate communication via the trusted channel for [assignment: *list of functions for which a trusted channel is required*]. Hierarchical to: No other components Dependencies: No dependencies Management: a) Configuring the actions that require trusted channel, if supported Audit: a) Minimal: Failure of the trusted channel functions b) Minimal: Identification of the initiator and target of failed trusted channel functions c) Basic: All attempted uses of the trusted channel functions d) Basic: Identification of the initiator and target of all trusted channel functions	tween itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. **FTP_ITC.1.2** The TSF shall permit [**the remote trusted IT product**] to initiate communication via the trusted channel. **FTP_ITC.1.3** The TSF shall initiate communication via the trusted channel for [**all functions requiring a trusted channel as defined by the SFP Access Rule Policy**]. **Application Note** For the communication channel either a trusted channel to the TOE by cryptographic means or a channel to the TOE within a trusted environment can be used. In the latter case the TOE identifies the establishment of a trusted environment by a successful user authentication.
FTP_TRP **Trusted Path**	
FTP_TRP.1 **Trusted Path**	PP SSCD Type3 / PP SSCD Type2
FTP_TRP.1.1 The TSF shall provide a communication path between itself and [selection: *remote, local*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure. **FTP_TRP.1.2** The TSF shall permit [selection: *the TSF, local users, remote users*] to initiate communication via the trusted path. **FTP_TRP.1.3** The TSF shall require the use of the trusted path for [selection: *initial user authentication*, [assignment: *other services for which trusted path is required*]].	**FTP_TRP.1.1** The TSF shall provide a communication path between itself and [**local**] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure. **FTP_TRP.1.2** The TSF shall permit [**local users**] to initiate communication via the trusted path. **FTP_TRP.1.3** The TSF shall require the use of the trusted path for [**initial user authentication**]. **Application Note** For the communication path either a trusted path to

Hierarchical to: No other components Dependencies: No dependencies Management: a) Configuring the actions that require trusted path, if supported Audit: a) Minimal: Failures of the trusted path functions b) Minimal: Identification of the user associated with all trusted path failures, if available c) Basic: All attempted uses of the trusted path functions d) Basic: Identification of the user associated with all trusted path invocations, if available	the TOE by cryptographic means or a path to the TOE within a trusted environment can be used. In the latter case the TOE identifies the establishment of a trusted environment by a successful user authentication.

5.1.2 SOF Claim for TOE Security Functional Requirements

The required level for the Strength of Function of the TOE security functional requirements listed in the preceding chap. 5.1.1 is “SOF-high”. This correlates to the claimed assurance level with its augmentation by the assurance component AVA_VLA.4 (refer to the following chap. 5.1.3).

5.1.3 TOE Security Assurance Requirements

The TOE security assurance level is fixed as

EAL4 augmented by ADV_IMP.2, ATE_DPT.2, AVA_MSU.3 and AVA_VLA.4.

The assurance level with its augmentations is chosen in view of the requirements in the Protection Profiles /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/ and /PP SSCD Type2/.

The following table lists the security assurance requirements (SARs) for the TOE:

SAR	
Class ACM Configuration Management	ACM_AUT.1 Partial CM Automation
	ACM_CAP.4 Generation Support and Acceptance Procedures
	ACM_SCP.2 Problem Tracking CM Coverage

Class ADO **Delivery and Operation**	ADO_DEL.2 Detection of Modification
	ADO_IGS.1 Installation, Generation, and Start-up Procedures
Class ADV **Development**	ADV_FSP.2 Fully Defined External Interfaces
	ADV_HLD.2 Security Enforcing High-Level Design
	ADV_IMP.**2** Implementation of the TSF
	ADV_LLD.1 Descriptive Low-Level Design
	ADV_RCR.1 Informal Correspondence Demonstration
	ADV_SPM.1 Informal TOE Security Policy Model
Class AGD **Guidance Documents**	AGD_ADM.1 Administrator Guidance
	AGD_USR.1 User Guidance
Class ALC **Life Cycle Support**	ALC_DVS.1 Identification of Security Measures
	ALC_LCD.1 Developer Defined Life-Cycle Model
	ALC_TAT.1 Well-defined Development Tools
Class ATE **Tests**	ATE_COV.2 Analysis of Coverage
	ATE_DPT.**2** Testing: Low-Level Design
	ATE_FUN.1 Functional Testing
	ATE_IND.2 Independent Testing – Sample
Class AVA **Vulnerability Assessment**	AVA_MSU.**3** Analysis and Testing for Insecure States
	AVA_SOF.1 Strength of TOE Security Function Evaluation

	AVA_VLA.**4** Highly Resistant

5.1.4 Refinements of the TOE Security Assurance Requirements

All assurance components given in the table of chap. 5.1.3 are used as defined in /CC 2.3 Part3/ and /CEM 2.3 Part2/.

5.2 Security Requirements for the Environment of the TOE

5.2.1 Security Requirements for the IT-Environment

There are no security requirements for the IT-Environment of the TOE defined.

5.2.2 Security Requirements for the Non-IT-Environment

There are no security requirements for the Non-IT-Environment of the TOE defined.

6 TOE Summary Specification

6.1 TOE Security Functions

6.1.1 TOE Security Functions / TOE-IC

For the definition of the TOE Security Functions (TSF) related to the TOE-IC refer to the Security Target /ST-ICPhilips/, chap. 6.1.

The TSFs defined for the TOE-IC cover the following functions which are relevant for the TOE: F.RNG, F.HW_DES, F.OPC, F.PHY, F.LOG, F.COMP, F.MEM_ACC, F.SFR_ACC.

6.1.2 TOE Security Functions / TOE-ES

The following section gives a survey of the TSFs of the TOE´s Smartcard Embedded Software.

TOE Security Functions / TOE-ES	
Access Control	
F.ACS	**Security Attribute Based Access Control**
	The TSF enforces the SFP Access Rule Policy as defined in chap. 5.1.1.2. The access control is realised by usage of access rules as security attributes. Access to a DF, an EF, a key, a password or other user data is only possible if the related access rule is satisfied.
Identification and Authentication	
F.IA_AKEY	**Key Based User / TOE Authentication Based on Asymmetric Cryptography**
	The TSF provides the functionality of a key based external and internal authentication on the base of asymmetric cryptography. By an external authentication, users of the TOE can be authenticated with regard to the TOE. Vice versa, by an internal authentication, the TOE itself can be authenticated with regard to the external world. Both authentication mechanisms base on a challenge-response procedure using random numbers. The TSF enforces the following different internal and external authentication mechanisms: - Internal authentication without session key agreement according to /ISO 9796-2/, /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3

	- External authentication without session key agreement according to /ISO 9796-2/, /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3 - Internal authentication including one step of session key and send sequence counter agreement according to /ISO 9796-2/, /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3, /eHC2/, chap. 3.6 - External authentication including one step of session key and send sequence counter agreement according to /ISO 9796-2/, /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3, /eHC2/, chap. 3.6 - Internal authentication according to /HPC-SMC1/, chap. 11, Annex E.6, /eHC1/, chap. 10, Annex E.6 Note: Each external authentication process requires a preceding Get Challenge – operation. The private and public keys necessary on the card´s side for authentication purposes are either generated on-card (with support by the TSF F.RSA_KEYGEN) or imported during the initialisation, personalisation or end-usage phase of the TOE. In particular, the import of public keys can be performed in the form of CV certificates what is connected with the verification of the respective CV certificate under usage of the TSF F.VER_DIGSIG. In each case, the keys involved on the card´s side in the authentication processes have to be explicitly referenced prior to their usage. The access to the keys necessary for the authentication processes is controlled by the specific SFP which is defined for the respective application using the authentication keys. The execution of the specific SFP is task of the TSF F.ACS for access control. In the case of a successful external authentication attempt the TSF sets a corresponding actual security state for key based user authentication. The TSF makes use of asymmetric cryptography with generation and verification of RSA digital signatures resp. RSA encryption and decryption and is therefore directly connected with the TSF F.CRYPTO. Depending on the type of authentication mechanism, the combination of a successful internal and external authentication process can include the generation of session keys (incl. send sequence counter). Depending on the type of authentication mechanism, the TSF stores the generated session keys volatile and on demand as well persistently on the card. The generated keys can be used for securing the following data exchange between the TOE and the external world (in the current or a later session) with the objective of data confidentiality and data integrity and authenticity (Secure Messaging). In addition, as well depending on the type of authentication mechanism, the generated keys can be used further on for authentication processes based on symmetric cryptography.
F.IA_SKEY	**Key Based User / TOE Authentication Based on Symmetric Cryptography**
	The TSF provides the functionality of a key based external and internal authentication on the base of symmetric cryptography. By an external authentication, users of the TOE can be authenticated with regard to the TOE. Vice versa, by an internal authentication, the TOE itself can be authenticated with regard to the external world. Both authentication mechanisms base on a challenge-response procedure using random numbers. The TSF enforces the following different internal and external authentication mechanisms: - Internal authentication with / without individual key derivation and without session key generation according to /HPC-SMC1/, chap. 11, Annex E.4, /eHC1/, chap.10,

	Annex E.4, /ISO 9796-2/ - External authentication with / without individual key derivation and without session key generation according to /HPC-SMC1/, chap. 11, Annex E.4, /eHC1/, chap.10, Annex E.4, /ISO 9796-2/ - Mutual authentication with / without individual key derivation and without session key generation according /HPC-SMC1/, chap. 11, Annex E.4, /eHC1/, chap.10, Annex E.4, /ISO 9796-2/ - Internal authentication with / without individual key derivation and including the first step of session key and send sequence counter generation according to /HPC-SMC1/, chap. 11, Annex E.4, /eHC1/, chap.10, Annex E.4, /eHC2/, chap. 3.7, /ANSI X9.63/, /ISO 9796-2/ - External authentication with / without individual key derivation and including the last step of session key and send sequence counter generation according to /HPC-SMC1/, chap. 11, Annex E.4, /eHC1/, chap.10, Annex E.4, /eHC2/, chap. 3.7, /ANSI X9.63/, /ISO 9796-2/ - Mutual authentication with / without individual key derivation and including session key and send sequence counter generation according to /HPC-SMC1/, chap. 11, Annex E.4, /eHC1/, chap.10, Annex E.4, /eHC2/, chap. 3.7, /ANSI X9.63/, /ISO 9796-2/ Note: Each external authentication process requires a preceding Get Challenge – operation. The symmetric keys necessary on the card´s side for the authentication mechanisms can either be generated on-card by a derivation process for deriving individual keys before the main authentication process starts. This key derivation process is performed by the TSF F.CRYPTO. Alternatively, symmetric keys imported during the initialisation, personalisation or end-usage phase of the TOE or agreed within a preceding authentication process can be used. The access to the keys necessary for the authentication processes is controlled by the specific SFP which is defined for the respective application using the authentication keys. The execution of the specific SFP is task of the TSF F.ACS for access control. In the case of a successful external authentication attempt the TSF sets a corresponding actual security state for key based user authentication. The TSF makes use of symmetric cryptography with DES based encryption, decryption, MAC generation resp. MAC verification. Hence, the TSF F.IA_SKEY is directly connected with the TSF F.CRYPTO. Depending on the type of authentication mechanism, the combination of a successful internal and external authentication process can include the generation of session keys (incl. send sequence counter). Depending on the type of authentication mechanism, the TSF stores the generated session keys volatile and on demand as well persistently on the card. The generated keys can be used for securing the following data exchange between the TOE and the external world (in the current or a later session) with the objective of data confidentiality and data integrity and authenticity (Secure Messaging). In addition, as well depending on the type of authentication mechanism, the generated keys can be used further on for authentication processes based on symmetric cryptography.
F.IA_PWD	**Password Based User Authentication**
	Users of the TOE can be authenticated (towards the TOE) by means of a card holder authentication process. For the card holder authentication process, the TSF compares

	the card holder verification information, here a password (PIN), provided by a subject with a corresponding secret reference value stored permanently on the card. The TSF uses for the authentication process the password referenced by the external world. The access to the relevant password resp. its reference value is controlled by the specific SFP which is defined for the respective application using the password. The execution of the specific SFP is task of the TSF F.ACS for access control. The card holder authentication process can be performed by usage of the command Verify or Change Reference Data (whereat the latter command makes a password change possible). Each password used for authentication purposes is connected with an own error usage counter and an own usage counter. Furthermore, each password is connected with an own resetting code (PUK) whereat the resetting code itself is connected with an own usage counter (but no error usage counter). The number of applications of a password for authentication purposes with the command Verify is limited by its usage counter. The TSF allows at maximum for a number of authentication attempts with a password as restricted by its usage counter. The value for the usage counter can be predefined as infinite, i.e. the password can be used without any limit. A password with an expired usage counter cannot be longer used for authentication purposes with the command Verify (but with the command Change Reference Data). In the case of a password with a finite usage counter, each authentication attempt with the command Verify decrements the usage counter of the password, independently whether the authentication attempt succeeds or fails. A successful authentication attempt with the command Change Reference Data re-initialises the usage counter to its predefined initial value. The TSF detects for a password when a predefined number of consecutive unsuccessful authentication attempts occurs related to the card holder authentication process. Each consecutive unsuccessful comparison of the presented password with the reference value stored on the card is recorded by the TSF in order to limit the number of further authentication attempts with this password. In the case of a successful authentication attempt a corresponding actual security state for the password is set and the error usage counter of the password is re-initialised to its predefined initial value. If an authentication attempt with the password fails, the corresponding actual security state is reset and the error usage counter of the password is decreased. When the defined maximum number of unsuccessful authentication attempts has been met or surpassed, the TSF blocks the corresponding password for any further authentication attempt. A password with an expired error usage counter can be unblocked by usage of the related resetting code, provided that the usage counters of the password and of the resetting code are not expired. Otherwise, there is no way to unblock the password so that this password is invalid for each further authentication attempt. The unblocking of a blocked password can be performed by usage of the command Reset Retry Counter only. In the case of a successful authentication attempt with the resetting code related to the blocked password, the expired error usage counter is re-initialised to its initial value (as well as for the usage counter of the password) and hence, the password can be used further on for authentication attempts. The number of applications of a resetting code for authentication purposes is limited by its usage counter. The TSF allows at maximum for a number of authentication attempts with the resetting code as restricted by its usage counter. Each unblocking attempt with the

	command Reset Retry Counter decrements the usage counter of the resetting code, independently whether the authentication attempt with the resetting code succeeds or fails. The unblocking process for a blocked password can be combined with a change of this password. However, even if the command Reset Retry Counter resp. the authentication with the resetting code succeeds, the actual security state for the password will not be set. For security reasons, a password shall be connected with an error usage counter with a sufficiently small value as initial value. Furthermore, the usage of the related resetting code itself shall be limited by an usage counter with a sufficiently small initial value. In general, a security state set due to a successful authentication attempt can be valid for several following TOE commands. However, as well, it is possible to restrict the validity of such an authentication state to one single following TOE command, i.e. after the next command has accessed this security state it will be reset by the TSF. The TSF does not check the quality of passwords or resetting codes used. The sufficient quality of passwords and resetting codes lies in the responsibility of the external world only. The transfer of passwords and resetting codes to the TOE can be executed in unsecured mode, i.e. without usage of Secure Messaging, or alternatively in secured mode, i.e. with usage of Secure Messaging. In the latter case, the TSFs F.EX_CONF and F.EX_INT are involved.
Integrity of Stored Data	
F.DATA_INT	**Stored Data Integrity Monitoring and Action**
	The TSF monitors data stored within the TOE for integrity errors. This concerns all - DFs - EFs - Passwords incl. related attributes - Cryptographic keys incl. related attributes - Security critical data stored within the card and channel context (session keys incl. attributes, status information as actual security states for key and password based authentication, hash values, further security relevant card and channel information) The monitoring is based on the following attributes: - Checksum (CRC) attached to the header of a file - Checksum (CRC) attached to the data body of a file - Checksums (CRC) attached to each secret (password, cryptographic key) and its related attributes stored in the EEPROM - Checksums (CRC) attached to card and channel context related security critical information Each access of the TOE to a DF, to an EF, to a secret (password or cryptographic key incl. its related attributes) or to security critical card resp. channel context data the TSF is secured with an integrity check on base of the mentioned attributes. Upon detection of a data integrity error, the TSF informs the user about this fault (output of a warning). If the checksum of the header of a file has been detected as corrupted, the data con-

	tained in the affected file are no longer accessible. If the data contained in a file are not of integrity, the affected data will be treated in the following way: - For the Read access, the affected data will be exported, but the data export will be connected with a warning. - For the Update access, the integrity error of the affected data will be ignored, and the data imported by the command will be stored and a new checksum will be computed. - For all remaining access modes, the affected data will not be used for data processing. If a secret (password, cryptographic key) and its related attributes are corrupted, the secret and its related data will not be processed. If security critical card or channel context data are not of integrity, the Smartcard Embedded Software immediately jumps into an endless-loop (re-activation by reset possible).
Data Exchange	
F.EX_CONF	**Confidentiality of Data Exchange**
	The TSF provides the capability to ensure that secret data which is exchanged between the TOE and the external world remains confidential during transmission. For this purpose, encryption based on symmetric cryptography is applied to the secret data. The TSF ensures that the user and the user data's access condition have indicated confidentiality for the data exchange. Securing the data transfer with regard to data confidentiality is done by Secure Messaging according to the standard ISO/IEC 7816-4. The cryptographic key used for securing the data transfer is either a symmetric session or static key. In case of a session key, the key is negotiated during a preceding mutual authentication process (based on a random challenge and response procedure) between the TOE and the external world (realised by the TSFs F.IA_SKEY, F.IA_AKEY, F.CRYPTO). For encryption and decryption, the TSF makes use of the TSF F.CRYPTO for DES functionality.
F.EX_INT	**Integrity and Authenticity of Data Exchange**
	The TSF provides the capability to ensure that data which is exchanged between the TOE and the external world remains integer and authentic during transmission. For this purpose, cryptographic checksums based on symmetric cryptography are applied to the data. The TSF ensures that the user and the user data's access condition have indicated integrity and authenticity for the data exchange. Securing the data transfer with regard to data integrity and authenticity is done by Secure Messaging according to the standard ISO/IEC 7816-4. The cryptographic key used for securing the data transfer is either a symmetric session or static key. In case of a session key, the key is negotiated during a preceding mutual authentication process (based on a random challenge and response procedure) between

	the TOE and the external world (realised by the TSFs F.IA_SKEY, F.IA_AKEY, F.CRYPTO). For checksum securing and verification, the TSF makes use of the TSF F.CRYPTO for DES functionality.
Object Reuse	
F.RIP	**Residual Information Protection**
	The TSF ensures that any previous information content of a resource is explicitly erased upon the deallocation of the resource used for any of the following components: - All volatile and non-volatile memory areas used for operations in which security relevant material (as e.g. cryptographic data, passwords or other security critical data) is involved. Explicit erasure is defined as physical erasure.
Protection	
F.FAIL_PROT	**Hardware and Software Failure Protection**
	The TSF preserves a secure operation state of the TOE when the following types of failures and attacks occur: - HW and/or SW induced reset - Power supply cut-off - Power supply variations - Unexpected abortion of the execution of the TSF due to external or internal events (in particular, break of a transaction before completion) - System breakdown - Internal HW and/or SW failure - Manipulation of executable code - Corruption of status information (as e.g. card status information, object life cycle state, actual security state related to key and password based authentication, ...) - Environmental stress - Input of inconsistent or improper data - Tampering - Manipulation resp. insufficient quality of the HW-RNG The TSF makes use of HW and SW based security features and corresponding mechanisms to monitor and detect induced HW and SW failures and tampering attacks. In particular, the TSF is supported by the IC specific TSFs F.OPC and F.PHY. Upon the detection of a failure of the above mentioned type the TSF reacts in such a way that the TSP is not violated. The TOE changes immediately to a locked state and cannot be used any longer within the actual session. Depending on the type of the detected attack to the underlying IC (incl. its Dedicated Software) or to the Smartcard Embedded Software code the TOE will be irreversible locked resp. can be reactivated by a reset.
F.SIDE_CHAN	**Side Channel Analysis Control**

	The TSF provides suitable HW and SW based mechanisms to prevent attacks by side channel analysis like Simple Power Analysis (SPA), Differential Power Analysis (DPA), Differential Fault Analysis (DFA) and Timing analysis (TA). The TSF ensures that all countermeasures available are used in such a way that they support each other. In particular, the TSF is supported by the TSF F.LOG of the underlying IC and its Dedicated Support Software. The TSF acts in such a manner that all security critical operations of the TOE, in particular the TOE´s cryptographic operations, are suitably secured by these HW and SW countermeasures. The TSF guarantees that information on IC power consumption, information on command execution time and information on electromagnetic emanations do not lead to useful information on processed security critical data as secret cryptographic keys or passwords. In particular, the IC contacts as Vcc, I/O and GND or the IC surface do not make it possible for an attacker to gain access to security critical data as secret cryptographic keys or passwords. The TSF enforces the installation of a secure session before any cryptographic operation is started. In particular, the installation of a secure session does not only concern the core cryptographic operation itself. All preparing security relevant actions performed prior to the core cryptographic operation as e.g. the generation of session keys, the process of loading keys into the dedicated IC cryptographic modules and the data preparation as reformatting or padding are involved as well. Furthermore, the secure session covers all security relevant actions which follow the core cryptographic operation as e.g. the post-processing of the output data.
F.SELFTEST	**Self Test**
	The TSF covers different types of self tests whereat each self test consists of a check of a dedicated integrity attribute related to (parts of) the TOE´s code resp. data. The TSF integrates self tests with the following objectives: The TSF provides the capability of conducting a self test during initial start-up, i.e. after each reset, to demonstrate the correct operation of its TSFs. This self test is performed automatically by the TOE and consists of the verification of the integrity of any software code stored in the EEPROM area. Furthermore, the TSF provides authorised users - here the Smartcard Embedded Software of the TOE (TOE-ES) itself - with the capability to verify the integrity of TSF data during run-time. The self test is performed automatically by the TOE and is supported by the TSF F.DATA_INT. Additionally, the TSF provides authorised users with the capability to verify the integrity of stored TSF executable code. This concerns only the production phase, more precise the initialisation phase of the TOE (phase 5 of the product´s life cycle). Prior to the initialisation of the TOE, the ROM-code of the TOE can be verified on demand by the Smartcard Embedded Software developer. The integrity of the whole EEPROM-code is checked automatically by the TOE during the storage of the initialisation file in the framework of the TOE´s initialisation. These self tests are supported by the TSF F.CRYPTO (SHA-1 hash value calculation, MAC verification). The TSF supports all other TSFs defined for the Smartcard Embedded Software (TOE-ES).
Cryptographic Operations	

F.CRYPTO	**Cryptographic Support**
	The TSF provides cryptographic support for the other TSFs using cryptographic mechanisms. The TSF supports: - DES/3DES algorithm according to the standard /FIPS 46-3/ resp. /ANSI X9.52/ with a key length of 56 resp. 112 bit entropie (used for encryption, decryption, MAC generation and verification according to /FIPS 46-3/, /ANSI X9.52/, /ANSI X9.19/, /HPC-SMC1/, chap. 11, 4.1, /eHC1/, chap. 10, 3.1.1) - RSA core algorithm according to the standard /PKCS1/ with key lengths of 1024, 1280, 1536, 1792 resp. 2048 bit modulus lengths (used for RSA encryption, decryption, signature generation and verification, see other TSFs related to RSA based mechanisms) - Random number generation by a deterministic RNG (incl. online-test of the HW-RNG for seeding the SW-RNG) - SHA-1 hash value calculation according to /ALGCAT/, chap. 2 resp. /FIPS 180-2/ - Negotiation of 3DES session keys - Derivation of individual 3DES keys according to the standard /ISO 10118-2/ (including a H2 hash value calculation and DES calculations) The resistance of the TSF against SPA, DPA, DFA and TA is part of the TSF F.SIDE_CHAN. The random number generation is in particular used for RSA and DES key generation and authentication mechanisms. The mechanism for the generation of session keys is directly connected with the TSFs F.IA_AKEY and F.IA_SKEY which realise internal and external authentication processes. Furthermore, the generation of random numbers of high quality, and depending on the authentication type, the SHA-1 hash value calculation of TSF F.CRYPTO are involved. The mechanism for the derivation of individual keys makes use of the SHA-1 hash value calculation and DES based calculations of the TSF F.CRYPTO. The TSF is directly supported by the TSFs of the underlying IC which supply cryptographic functionality.
F.RSA_KEYGEN	**RSA Key Pair Generation**
	The TSF generates RSA key pairs with key lengths of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length for asymmetric cryptography which can be used later on e.g. for digital signatures or authentication purposes. The TSF enforces the key pair generation process and the related key material to meet the following requirements: - The RSA key pair generation process follows a well-designed key generation algorithm of sufficient quality; in particular, the requirements for RSA keys and their generation in /ALGCAT/, chap. 3.1 and 4 as well as in the corresponding European algorithm paper, chap. 4.5.2, 4.6, Annex C.2 and C.3 are taken into account. - Random numbers used in the key pair generation process for the generation of the primes are of high quality to ensure that the new key pair is unpredictable and unique with a high probability.

	- The generation of the random numbers necessary for the primes is performed by usage of a deterministic RNG running on the TOE. - Prime numbers produced in the key pair generation process are unique with a high probability and satisfy the requirements in /ALGCAT/, chap. 3.1 and 4. In particular, the so-called epsilon-condition is considered. - The primes are independently generated. - Sufficiently good primality tests with convincing limits are implemented to guarantee with a high probability for the property of the generated prime candidates to be prime. In particular, the actual version of the significance limit for primality tests is considered. - In the key pair generation process, for the public exponent given by the external world the corresponding private exponent is calculated and converted into its CRT parameters. - For each key length, the generated key pairs show a “good” distribution within the key range; in particular, the generated new key pair is unique with a high probability. - Only cryptographically strong key pairs with the intended key length are generated. In particular, for any generated key pair, the private key cannot be derived from the corresponding public key. - The key pair generation process includes a dedicated check if the generated private and public key match; only valid key pairs are issued. - During the key pair generation process, it is not possible to gain information about the chosen random numbers, about the calculated primes, about other secret values which will be used for the key pair to be generated or about the generated key pair and its parts itself. - During the key pair generation process, it is not possible to gain information about the design of the routines realising the key pair generation. - The key pair generation process includes a physical destruction of the old private key part before the new key pair is generated. The resistance of the TSF against SPA, DPA, DFA and TA is part of the TSF F.SIDE_CHAN. The TSF makes use of the TSF F.CRYPTO for random number generation and RSA signature generation and verification. The public part of the generated key pair can be exported with an authentication attribute which either can be a MAC (generation supported by the TSF F.CRYPTO) or a digital signature (generation supported by the TSF F.GEN_DIGSIG) over the public key data.
F.GEN_DIGSIG	**RSA Generation of Digital Signatures**
	The TSF provides a digital signature functionality based on asymmetric cryptography, particularly based on the RSA algorithm with key lengths of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length. The TSF digital signature function will be used for several purposes with different formats for the digital signature input: - Explicit generation of digital signatures using the signature scheme with appendix according to the standard /PKCS1/, chap. 8.2.1 and with hash algorithm SHA-1 (external SHA-1 hash value calculation), see /HPC-SMC1/, chap. 11, /eHC1/, chap. 10

	- Explicit generation of digital signatures using the signature scheme with appendix according to the standard /ISO 9796-2/ with random number based on the hash algorithm SHA-1 (external SHA-1 hash value calculation), see /HPC-SMC1/, chap. 11, /eHC1/, chap. 10 - Implicit generation of digital signatures within authentication mechanisms for the creation of authentication tokens using the signature scheme with message recovery according to the standard /ISO 9796-2/ based on the hash algorithm SHA-1, see /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3 - Implicit generation of digital signatures within authentication mechanisms for the creation of authentication tokens using the signature scheme with message recovery according to the standard /PKCS1/, chap. 8.2.1 without hash and OID, but with an additional limitation of the length of the input message, see /HPC-SMC1/, chap. 11, Annex E.6, /eHC1/, chap. 10, Annex E.6 The TSF function for generation of a digital signature uses the private key which has been referenced before. The random numbers necessary for the padding of the data within the signature process are generated by using the TSF F.CRYPTO for random number generation. Furthermore, for the signature calculation itself, the TSF makes use of the TSF F.CRYPTO, and the computation of hash values is as well based on the TSF F.CRYPTO. Each private key used for the signature generation function is either generated on-card by usage of the TSF F.RSA_KEYGEN or is generated by the external world and loaded onto the card during the initialisation, personalisation or end-usage phase of the TOE. In the latter case, it is in the responsibility of the external world to guarantee for a sufficient cryptographic strength of the private key and to handle the private key outside the card in a sufficient secure manner. The resistance of the TSF against SPA, DPA, DFA and TA is part of the TSFs F.Log and F.SIDE_CHAN. For each private key - generated on-card or imported with the assumption that the external world meets the requirements on the key handling as defined before - the TSF digital signature function works in such a manner that the private key cannot be derived from the signature and the signature cannot be generated by other individuals not possessing that secret. Furthermore, the TSF digital signature function works in such a manner that no information about the private key can be disclosed during the generation of the digital signature.
F.VER_DIGSIG	**RSA Verification of Digital Signatures**
	The TSF provides a functionality to verify digital signatures based on asymmetric cryptography, particularly based on the RSA algorithm with key lengths of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length. The TSF function to verify a digital signature will be used for several purposes with different formats for the digital signature input: - Implicit verification of digital signatures within authentication mechanisms for the verification of authentication tokens using the signature scheme with message recovery according to the standard /ISO 9796-2/ based on the hash algorithm SHA-1, see /HPC-SMC1/, chap. 11, Annex E.2, E.3, /eHC1/, chap. 10, Annex E.2, E.3 - Implicit verification of digital signatures within the verification and unwrapping of imported CV certificates using the signature scheme with message recovery according to the standard /ISO 9796-2/ based on the hash algorithm SHA-1, see /HPC-SMC1/, Annex B, /eHC1/, chap. 10, Annex B The TSF function to verify a digital signature uses the public key which has been referenced before.

	For the verification mechanism itself, the TSF makes directly use of the TSF F.CRYPTO, and the computation of hash values is as well based on the TSF F.CRYPTO. Each public key used for the function to verify a digital signature is either generated on-card by usage of the TSF F.RSA_KEYGEN or is generated by the external world and loaded onto the card during the initialisation, personalisation or end-usage phase of the TOE. In particular, loading via a CV certificate by a suitable preceding operation is possible.
F.RSA_ENC	**RSA Encryption**
	The TSF provides a functionality to encrypt data based on asymmetric cryptography, particularly based on the RSA algorithm with key lengths of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length. The TSF encryption function will be used for several purposes with different formats for the encryption input: - Explicit encryption of a plain text using the “encryption scheme” with formatted plain message according to the standard /PKCS1/, chap. 7.2.1 and with hash algorithm SHA-1, see /HPC-SMC1/, chap. 11, 4.1, /eHC1/, chap. 10, 3.1.1 - Implicit encryption within authentication mechanisms for the generation of authentication tokens using the “encryption primitive” according to the standard /PKCS1/, chap. 5.1.1 The TSF encryption function uses the public key which has been referenced before. For the encryption mechanism itself, the TSF makes directly use of the TSF F.CRYPTO. Each public key used for the encryption function is either generated on-card by usage of the TSF F.RSA_KEYGEN or is generated by the external world and loaded onto the card during the initialisation, personalisation or end-usage phase of the TOE. In particular, loading via a CV certificate by a suitable preceding operation is possible.
F.RSA_DEC	**RSA Decryption**
	The TSF provides a functionality to decrypt data based on asymmetric cryptography, particularly based on the RSA algorithm with key lengths of 1024, 1280, 1536, 1792 resp. 2048 bit modulus length. The TSF decryption function will be used for several purposes with different formats for the data supplied within the cryptogram: - Explicit decryption of a cryptogram using the “decryption scheme” with formatted input according to the standard /PKCS1/, chap. 7.2.2 and with hash algorithm SHA-1, see /HPC-SMC1/, chap. 11, 4.1, /eHC1/, chap. 10, 3.1.1 - Implicit decryption within authentication mechanisms for the verification of authentication tokens using the “decryption primitive” according to the standard /PKCS1/, chap. 5.1.2 The TSF decryption function uses the private key which has been referenced before. For the decryption mechanism itself, the TSF makes directly use of the TSF F.CRYPTO. Each private key used for the decryption function is either generated on-card by usage of the TSF F.RSA_KEYGEN or is generated by the external world and loaded onto the card during the initialisation, personalisation or end-usage phase of the TOE. In the latter case, it is in the responsibility of the external world to guarantee for a sufficient cryptographic

	strength of the private key and to handle the private key outside the card in a sufficient secure manner. The resistance of the TSF against SPA, DPA, DFA and TA is part of the TSFs F.Log and F.SIDE_CHAN. For each private key - generated on-card or imported with the assumption that the external world meets the requirements on the key handling as defined before - the TSF decryption function works in such a manner that the private key cannot be derived from the cryptogram and the cryptogram cannot be deciphered by other individuals not possessing that secret. Furthermore, the TSF decryption function works in such a manner that no information about the private key may be disclosed during the decipherment of the cryptogram.

6.2 SOF Claim for TOE Security Functions

According to Common Criteria, /CC 2.3 Part1/ and /CC 2.3 Part3/, all TOE Security Functions (TSF) which are relevant for the assurance requirement AVA_SOF.1 are identified in this section.

For the TSFs explicitly defined for the underlying IC, information on the SOF claim can be found in /ST-ICPhilips/.

The TSFs related to the complete product using mechanisms which can be analysed for their permutational or probabilistic properties and which contribute to AVA_SOF.1 are the following:

TOE Security Function	SOF Claim	Description / Explanation
F.ACS	Not applicable	The TSF is not realised by permutational or probabilistic mechanisms.
F.IA_AKEY	SOF high	The TSF implements under usage of the TSFs F.CRYPTO, parts for RSA operations, hash value calculation and random number generation, and of the TSFs F.GEN_DIGSIG, F.VER_DIGSIG, F.ENC and F.DEC cryptographic mechanisms for authentication. The TSF is realised by permutational and probabilistic mechanisms.
F.IA_SKEY	SOF-high	The TSF implements under usage of the TSFs F.CRYPTO, parts for DES operations and random number generation, cryptographic mechanisms for authentication. The TSF is realised by permutational and probabilistic mechanisms.
F.IA_PWD	SOF high	The TSF includes a probabilistic password mechanism for the authentication of the user.
F.DATA_INT	Not applicable	In general, the mechanisms for generating and checking CRC-checksums can be analysed with permutational or probabilistic methods. But these mechanisms are not relevant for AVA_SOF.1 as the securing of data areas by CRC-checksums

		is only intended to secure against *accidental* data modification.
F.EX_CONF	Not applicable	The TSF includes cryptographic mechanisms using DES functionality from the TSF F.CRYPTO. Refer to the explanations for F.CRYPTO concerning the SOF claim resp. valuation of DES based encryption / decryption functions.
F.EX_INT	Not applicable	The TSF includes cryptographic mechanisms using DES functionality from the TSF F.CRYPTO. Refer to the explanations for F.CRYPTO concerning the SOF claim resp. valuation of DES based MAC generation / MAC verification functions.
F.RIP	Not applicable	The TSF is not realised by permutational or probabilistic mechanisms.
F.FAIL_PROT	Not applicable	The TSF is not realised by permutational or probabilistic mechanisms.
F.SIDE_CHAN	Not applicable	The TSF is not realised by permutational or probabilistic mechanisms.
F.SELFTEST	Not applicable	The TSF is not realised by permutational or probabilistic mechanisms, except for the functionality supported by the TSFs F.DATA_INT and F.CRYPTO (→ refer to the SOF claim for these TSFs).
F.CRYPTO	SOF high	The TSF includes cryptographic algorithms SHA-1, RSA with key lengths 1024, 1280, 1536, 1792 and 2048 bit modulus length as well as random number generation by usage of a deterministic RNG of quality class K4. These algorithms and key lengths defined for the TSF comply with the requirements in /ALGCAT/, chap. 2, 3.1, 4 for qualified electronic signatures and fulfill therefore the requirements for SOF high. The TSF part concerning DES functionality (used for encryption, decryption, MAC generation and MAC verification) are as well assigned to the SOF claim as permutational and probabilistic mechanisms are involved. The negotiation of session keys and the derivation of individual keys is not considered to part for the SOF analysis.
F.RSA_KEYGEN	SOF high	The TSF includes permutational and probabilistic mechanisms for the key generation process itself as well as for the integrated random number generation and key check. In particular, functionality from the TSF F.CRYPTO (random number generation, RSA signature generation and verification) is used by this TSF.
F.GEN_DIGSIG	SOF high	The TSF implements under usage of the TSF F.CRYPTO, parts for RSA operations and random number generation, cryptographic mechanisms for signature generation. The TSF is realised by permutational and probabilistic mechanisms, in particular the quality of the implemented security mechanisms against leakage can be analysed using permutational or probabilistic methods.
F.VER_DIGSIG	Not applicable	The implementation of the TSF uses only public keys and

		needs not to be considered with regard to high attack potential so that securing of the implementations against Simple Power Analysis (SPA), Differential Power Analysis (DPA), Differential Fault Analysis (DFA) and Timing Attacks (TA) is not necessary. Because of this fact, the TSF – although it can be analysed with permutational or probabilistic methods - is not relevant for AVA_SOF.1. Nevertheless, this TSF is secured by appropriate hardware security features.
F.RSA_ENC	Not applicable	The implementation of the TSF uses only public keys and needs not to be considered with regard to high attack potential so that securing of the implementations against Simple Power Analysis (SPA), Differential Power Analysis (DPA), Differential Fault Analysis (DFA) and Timing Attacks (TA) is not necessary. Because of this fact, the TSF – although it can be analysed with permutational or probabilistic methods - is not relevant for AVA_SOF.1. Nevertheless, this TSF is secured by appropriate hardware security features.
F.RSA_DEC	SOF high	The TSF implements under usage of the TSF F.CRYPTO, part for RSA operations, cryptographic mechanisms for decryption. The TSF is realised by permutational and probabilistic mechanisms, in particular the quality of the implemented security mechanisms against leakage can be analysed using permutational or probabilistic methods.

For each of the TOE Security Functions given in the preceding list an explicit claim of “SOF-high” is made.

The TOE´s cryptographic algorithms themselves can also be analysed with permutational or probabilistic methods but this is not in the scope of CC evaluations.

6.3 Assurance Measures

Appropriate assurance measures will be employed by the developer of the TOE to satisfy the security assurance requirements defined in chap. 5.1.3. For the evaluation of the TOE, the developer will provide appropriate documents describing these measures and containing further information supporting the check of the conformance of these measures against the claimed assurance requirements.

For the Smartcard Embedded Software part of the TOE (TOE-ES), the following table gives a mapping between the assurance requirements and the documents containing the relevant information for the respective requirement. All these documents concerning the TOE-ES are provided by the developer of the TOE-ES. The table below contains only the directly related documents, references to further documentation can be taken from the mentioned documents.

Overview of Developer´s TOE-ES related Documents		
Assurance Class	**Family**	**Document containing the relevant information**
ACM Configuration Management	ACM_AUT	- Document Configuration Control System
	ACM_CAP	- Document Life-Cycle Model - Document Configuration Control System
	ACM_SCP	- Document Configuration Control System - Document Life-Cycle Model
ADO Delivery and Operation	ADO_DEL	- Document Life-Cycle Model
	ADO_IGS	- Document Installation, Generation and Start-Up Procedures
ADV Development	ADV_FSP	- Document Functional Specification
	ADV_HLD	- Document High-Level Design - Detailed development documents as system specifications, design specifications, etc.
	ADV_LLD	- Document Low-Level Design - Detailed development documents as system specifications, design specifications, etc.
	ADV_IMP	- Source Code - Detailed development documents as system specifications, design specifications, etc.
	ADV_RCR	- Document Functional Specification - Document High-Level Design - Document Low-Level Design
	ADV_SPM	- Document TOE Security Policy Model
AGD Guidance Documents	AGD_ADM	--- (Part of the User Guidance)
	AGD_USR	- User Guidance for the User of the MICARDO Card
ALC Life Cycle Support	ALC_DVS	- Document Security of the Development Environment
	ALC_LCD	- Document Life-Cycle Model
	ALC_TAT	- Configuration List
ATE Tests	ATE_COV	- Document Test Documentation - Detailed test documentation as system test specifications, test protocols, etc.
	ATE_DPT	- Document Test Documentation - Detailed test documentation as system test specifications, test protocols, etc.

	ATE_FUN	- Document Test Documentation - Detailed test documentation as system test specifications, test protocols, etc.
	ATE_IND	- Samples of the TOE - Source Code
AVA Vulnerability Assessment	AVA_MSU	- Document Analysis of the Guidance Documents
	AVA_SOF	- Document TOE Security Function Evaluation
	AVA_VLA	- Document Vulnerability Analysis

As mentioned, the evaluation of the TOE will be done as composite evaluation on basis of the evaluated IC "Philips SmartMX P5CC036V1D Secure Smart Card Controller" provided by Philips Semiconductors GmbH. Therefore, for the TOE-IC the following documents will be at least provided by the IC developer:

Overview of Developer´s TOE-IC related Documents	
Class	**Documents**
Security Target	Security Target of the IC evaluation, /ST-ICPhilips/
Evaluation Report	Evaluation Technical Report Lite (ETR Lite) of the IC evaluation, /ETRLite-ICPhilips/
Configuration List	Configuration List for composite evaluation with ORGA, /ConfListPhilips/
User Guidances	User Guidance for the IC, /UG-ICPhilips/
	Data Sheet for the IC, /DS-ICPhilips/
	Instruction Set for the IC, /IS-ICPhilips/

7 PP Claims

Not applicable. Refer to chap. 1.3.

8 Rationale

The following chapters cover the security objectives rationale, the security requirements rationale and the TOE summary specification rationale.

8.1 Security Objectives Rationale

According to the requirements of Common Criteria, /CC 2.3 Part1/ and /CC 2.3 Part3/, the security objectives rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them. In detail, the security objectives rationale demonstrates that the stated security objectives for the TOE and its environment are suitable to counter the identified threats to security and to cover all of the identified Organisational Security Policies and assumptions. Vice versa, the security objective rationale shows that each security objective of the TOE and its environment at least counters one threat or is correlated to one Organisational Security Policy or assumption.

8.1.1 Threats - Security Objectives

8.1.1.1 Threats of the TOE-IC

The threats of the TOE-IC as defined in chap. 3.3.1 are covered completely by the security objectives for the TOE-IC in chap. 4.1.1. The mapping of the threats of the TOE-IC to the relevant security objectives is done within the CC evaluation of the IC resp. within the associated Security Target.

8.1.1.2 General Threats of the TOE-ES

The general threats of the TOE-ES as defined in chap. 3.3.2 are covered completely by the general security objectives for the TOE-ES and the general security objectives for the environment of the TOE as listed in chap. 4.1.2 and 4.2.1. The mapping of the general threats of the TOE-ES to the relevant security objectives is done within the Protection Profile /PP9911/, chap. 8.2.2.

For the TOE-ES, the assumptions A.Plat-Appl, A.Process-Card, A.Check-Init, A.Resp-Appl and A.Key-Function for the TOE-IC (refer to /ST-ICPhilips/) have been redefined suitably as security objectives O.Plat-Appl, O.Process-Card, O.Check-Init, O.Resp-Appl and O.Key-Function for the TOE-ES resp. for the environment of the TOE. The following supplements hold concerning these additional security objectives for the TOE-ES resp. the environment of the TOE:

O.Plat-Appl

As the Smartcard Embedded Software (TOE-ES) is designed in such a manner that the requirements from the TOE-IC guidance documents (hardware data sheet, application notes etc.) and the findings of the TOE-IC evaluation reports relevant for the Smartcard Embedded Software are met, this security objective contributes to the defense of the threats T.CLON*, T.DIS_ES2, T.T_ES, T.T_CMD, T.MOD_LOAD, T.MOD_EXE, T.MOD_SHARE and T.MOD_SOFT*.

O.Process-Card

This security objective guarantees for secure delivery procedures for the TOE or parts of it by the TOE Manufacturer during the phases 4 to 6 of the product life-cycle with the goal to maintain confidentiality and integrity of the TOE and to prevent any possible copy, modification, retention, theft or unauthorised use. It therefore counters the threats T.CLON*, T.DIS_DEL1, T.DIS_DEL2, T.MOD_DEL1, T.MOD_DEL2 and T.T_ES.

O.Check-Init

The security objective O.Check-Init provides the capability for the external world to check the identity of the TOE-IC by specific IC data. This security objective supplements the security objective O.Identification of the TOE-IC and therefore, the mapping to the relevant threats, assumptions or organisational policies is covered by the CC evaluation of the IC.

O.Resp-Appl

As the Smartcard Embedded Software (TOE-ES) is designed in such a manner that security relevant user data are treated by the TOE as required by the security needs of the respective application context this security objective contributes to the defense of the threats T.DIS_ES2, T.T_CMD, T.MOD_LOAD, T.MOD_EXE, T.MOD_SHARE and T.MOD_SOFT*.

O.Key-Function

As the Smartcard Embedded Software (TOE-ES) is designed in such a manner that key-dependent functions and their implementation are not susceptible to leakage attacks, this security objective contributes to the defense of the threat T.DIS_ES2.

8.1.1.3 Specific Threats of the TOE-ES

The security objective O.KEYGEN addresses directly the specific threat T.KEYGEN.

8.1.2 Assumptions - Security Objectives

The assumptions for the environment of the TOE as defined in chap. 3.2 except the assumption A.PERS are covered completely by the general security objectives for the environment

of the TOE as listed in chap. 4.2.1. The mapping of these assumptions for the environment of the TOE to the relevant security objectives is done within the Protection Profile /PP9911/, chap. 8.2.3. Furthermore, the additional security objective O.PERS for the environment of the TOE covers directly the additional assumption A.PERS, as its definition shows.

8.1.3 Organisational Security Policies - Security Objectives

The security objective O.Process-Card requires the developer of the TOE to implement measures as assumed in the Organisational Security Policy P.Process-Card, thus the security objective is covered by the mentioned Organisational Security Policy.

Furthermore, the Organisational Security Policy P.Design-Software obviously covers the security objectives O.Plat-Appl, O.Resp-Appl, O.Check-Init and O.Key-Function as their definition shows.

8.2 Security Requirements Rationale

According to the requirements of Common Criteria, /CC 2.3 Part1/ and /CC 2.3 Part3/, the security requirements rationale demonstrates that the set of security requirements of the TOE is suitable to meet and is traceable to the security objectives for the TOE and its environment. In detail, the following will be demonstrated:

- the combination of the individual functional and assurance requirements components for the TOE and its IT environment together meet the stated security objectives
- the set of security requirements together form a mutually supportive and internally consistent whole
- the choice of security requirements is justified, whereby any of the following conditions is specifically justified:
  - choice of additional requirements not contained in Parts 2 or 3
  - choice of additional assurance requirements not included in EAL 4
  - non-satisfaction of dependencies
- the selected strength of function level for the ST is consistent with the security objectives for the TOE

8.2.1 Security Functional Requirements Rationale

The following section demonstrates that the set and combination of the defined security functional requirements (SFRs) and security assurance requirements (SARs) for the TOE is suitable to satisfy the identified security objectives for the TOE and its environment. Furthermore, this section shows that each of these SARs and SFRs contributes to at least one of the security objectives for the TOE and its environment.

8.2.1.1 Security Objectives for the TOE-IC - Security Functional Requirements

The security objectives for the TOE-IC of chap. 4.1.1 are related to the SARs and SFRs for the TOE defined in chap. 5.1.3 and 5.1.1.1. The mapping of the security objectives for the TOE-IC to the relevant SARs and SFRs is done within the CC evaluation of the IC resp. within the associated Security Target.

8.2.1.2 General Security Objectives for the TOE-ES - Security Functional Requirements

The general security objectives for the TOE-ES of chap. 4.1.2 except O.Plat-Appl, O.Resp-Appl, O.Check-Init and O.Key-Function are related to the SARs and SFRs for the TOE as defined in chap. 5.1.3 and 5.1.1.2. The mapping of these general security objectives for the TOE-ES to the relevant SARs and SFRs is done within the Protection Profile /PP9911/, chap.

8.3.1 whereat the following supplements related to the SFRs which are defined in chap. 5.1.1.2 in addition to the SFRs in /PP9911/, chap. 5 have to be considered:

Security Functional Requirements / Security Objectives	**O.TAMPER_ES**	**O.OPERATE***	**O.DIS_MECHANISM2**	**O.DIS_MEMORY***	**O.MOD_MEMORY***	**O.FLAW***	**O.CLON***	**O.KEYGEN**
FCS_CKM.1 / RSA-KeyGen								**x**
FCS_CKM.1 / DES-KeyGen-Asym				**x**	**x**			
FCS_CKM.1 / DES-KeyGen-Sym				**x**	**x**			
FCS_CKM.1 / KeyDerivation				**x**	**x**			
FCS_CKM.2					**x**			
FCS_RND.1				**x**	**x**			**x**
FDP_UCT.1				**x**				
FDP_UIT.1					**x**			
FIA_UAU.6	**x**			**x**	**x**			
FMT_LIM.1						**x**		
FMT_LIM.2						**x**		
FMT_SMF.1	**x**	**x**						
FPT_AMT.1		**x**			**x**			
FPT_EMSEC.1	**x**		**x**	**x**			**x**	
FPT_PHP.1	**x**	**x**	**x**	**x**	**x**		**x**	
FPT_RVM.1	**x**	**x**						
FTP_ITC.1				**x**	**x**			
FTP_TRP.1				**x**	**x**			

The cryptographic support functional requirement FCS_CKM.1 / RSA-KeyGen supports the generation of RSA key pairs which match the requirements in O.KEYGEN.

The cryptographic support functional requirements FCS_CKM.1 / DES-KeyGen-Asym, FCS_CKM.1 / DES-KeyGen-Sym and FCS_CKM.1 / KeyDerivation support the establishment of secure communication channels between the TOE and the external world by generating the necessary keys. Hence, they contribute to O.DIS_MEMORY* and O.MOD_MEMORY*.

FCS_CKM.2 covers the secured export of public keys which are generated oncard and contributes to the objective O.MOD_MEMORY*.

The cryptographic support functional requirement FCS_RND.1 supports the generation of RSA key pairs which match the requirements in O.KEYGEN. Furthermore, it supports as well the generation and derivation of session keys for secure communication channels between the TOE and the external world and therefore contributes to O.DIS_MEMORY* and O.MOD_MEMORY*.

Sensitive information can be exchanged between the TOE and the external world with respect to confidentiality. The TSF control function FDP_UCT.1 contributes to the realization of O.DIS_MEMORY* as it serves for confidential data transfer.

Sensitive information can be exchanged between the TOE and the external world with respect to integrity. The TSF control function FDP_UIT.1 contributes to the realization of O.MOD_MEMORY* as it serves for integer data transfer.

The identification and authentication functional requirement FIA_UAU.6 prevents unauthorized access to stored memory, and thus contributes to security objectives O.TAMPER_ES, O.DIS_MEMORY* and O.MOD_MEMORY*.

The FMT_LIM.1 functional requirement directly contributes to the objective O.FLAW*.

The FMT_LIM.2 functional requirement as well contributes directly to the objective O.FLAW*.

The FMT_SMF.1 functional requirement meets O.TAMPER_ES and O.OPERATE* objectives.

FPT_AMT.1 functional requirement meets O.MOD_MEMORY* and partially O.OPERATE*. The suite of self tests may run only during initial start-up of the TOE, aiming at the integrity of executable code and/or sensitive memory content. Each test yields a global answer depending of the result of the test. This test has to be defined, but it is clear that a correct authentication process or a correct cryptographic operation demonstrate the correct operation of the TSF during execution of commands.

The FPT_EMSEC.1 functional requirement serves for the objectives O.TAMPER_ES, O.DIS_MEMORY*, O.DIS_MECHANISM2 and O.CLON*.

The FPT_PHP.1 functional requirement meets O.TAMPER_ES, O.DIS_MEMORY*, O.MOD_MEMORY* and O.CLON*. FPT_PHP.1 also meets O.OPERATE* and O.DIS_MECHANISM2.

The FPT_RVM.1 functional requirement meets O.TAMPER_ES and O.OPERATE* objectives.

Sensitive information can be securely exchanged between the TOE and the external world. The TSF control function FTP_ITC.1 contributes to the realization of O.DIS_MEMORY* and O.MOD_MEMORY* as it covers the set-up of a trusted communication channel.

Sensitive information can be securely exchanged between the TOE and the external world. The TSF control function FTP_TRP.1 contributes to the realization of O.DIS_MEMORY* and O.MOD_MEMORY* as it covers the set-up of a trusted communication path.

For the TOE-ES, as mentioned above, the assumptions A.Plat-Appl, A.Resp-Appl, A.Check-Init and A.Key-Function of the TOE-IC (refer to /ST-ICPhilips/) have been redefined suitably as security objectives for the TOE-ES. The following supplements hold concerning these additional security objectives for the TOE-ES:

O.Plat-Appl, O.Resp-Appl

The design of the TOE-ES in such a manner, that the requirements from the TOE-IC guidance documents (hardware data sheet, application notes etc.), from the findings of the TOE-IC evaluation reports relevant for the Smartcard Embedded Software and from the requirements of the MICARDO Card specification are met (O.Plat-Appl, O.Resp-Appl), is covered by the SARs for the whole TOE. In particular, the components of the class ADV with its design documentation and implementation representation (refer to chap. 5.1.3) contribute to the fulfillment of the security objectives O.Plat-Appl and O.Resp-Appl.

O.Key-Function

The design of the TOE-ES in such a manner, that the key-dependent functions are implemented in the TOE-ES in such a way that they are not susceptible to leakage attacks (O.Key-Function) is covered by the SARs for the whole TOE. In particular, the components of the class ADV with its design documentation and implementation representation and the components of the class AVA for vulnerability analysis (refer to chap. 5.1.3) contribute to the fulfillment of the security objective O.Key-Function.

O.Check-Init

The security objective O.Check-Init provides the capability for the external world to check the initialisation data brought into the TOE during the IC manufacturing. This security objective supplements the security objective O.Identification of the TOE-IC and therefore, the mapping to the relevant SFRs and SARs is covered by the CC evaluation of the IC. Furthermore, this security objective is covered by the SARs for the whole TOE, in particular by the components of the class ADV with its design documentation and implementation representation (refer to chap. 5.1.3).

8.2.1.3 Specific Security Objectives for the TOE-ES - Security Functional Requirements

Refer to chap. 8.2.1.2 where the specific security objective O.KEYGEN is already considered.

8.2.2 Security Functional Requirements Dependencies

The following section demonstrates that all dependencies between the identified security functional requirements included in this ST are satisfied.

8.2.2.1 SFRs of the TOE-IC

The dependencies under the SFRs for the TOE-IC of chap. 5.1.1.1 are considered in the scope of the CC evaluation of the IC resp. within the associated Security Target.

8.2.2.2 SFRs of the TOE-ES

The table below gives an overview of all SFRs defined for the TOE-ES (refer to chap. 5.1.1.2) and their dependencies. For each SFR, an information is provided about which dependency is relevant and wether and by which other SFRs of this ST the dependency is satisfied. Hereby, if there exist according to the definitions in /CC 2.3 Part2/ alternative dependencies, only the chosen one is listed. Furthermore, only direct dependencies are considered.

Number	SFR		(Direct) Dependencies	Comment / Line Number
1	**FAU_SAA.1**	**Potential Violation Analysis**	- FAU_GEN.1	See below.
2	**FCS_CKM.1 / RSA-KeyGen**	**Cryptographic Key Generation**	- FCS_CKM.2 - FCS_COP.1 / Exp-RSA-GenDigSig-PKCS1 - FCS_COP.1 / Exp-RSA-GenDigSig-ISO9796-2 - FCS_COP.1 / RSA-Corresp - FCS_COP.1 / Imp-RSA-GenDigSig-PKCS1 - FCS_COP.1 / Imp-RSA-GenDigSig-ISO9796-2 - FCS_COP.1 / Exp-RSA-Dec - FCS_COP.1 / RSA-Dec-Primitive - FCS_COP.1 / Imp-RSA-VerDigSig-ISO9796-2 - FCS_COP.1 / Imp-RSA-VerDigSig-CV - FCS_COP.1 / Exp-RSA-Enc - FCS_COP.1 / RSA-Enc-Primitive - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	6, 11-20, 8, 53
3	**FCS_CKM.1 / DES-KeyGen-Asym**	**Cryptographic Key Generation**	- FCS_COP.1 / Imp-DES-Enc-MACGen - FCS_COP.1 / Imp-DES-MACVer-Dec - FCS_COP.1 / Imp-DES-MACVer-Dec-Enc-MACGen - FCS_COP.1 / DES-Enc-Dec - FCS_COP.1 / DES-MACGen-MACVer - FCS_CKM.4 / DES-KeyErasure - FMT_MSA.2	22-26, 9, 53
4	**FCS_CKM.1 / DES-KeyGen-Sym**	**Cryptographic Key Generation**	- FCS_COP.1 / Imp-DES-Enc-MACGen - FCS_COP.1 / Imp-DES-MACVer-Dec - FCS_COP.1 / Imp-DES-MACVer-	22-26, 9, 53

			Dec-Enc-MACGen - FCS_COP.1 / DES-Enc-Dec - FCS_COP.1 / DES-MACGen-MACVer - FCS_CKM.4 / DES-KeyErasure - FMT_MSA.2	
5	**FCS_CKM.1 / KeyDerivation**	**Cryptographic Key Generation**	- FCS_COP.1 / Imp-DES-Enc-MACGen - FCS_COP.1 / Imp-DES-MACVer-Dec - FCS_COP.1 / Imp-DES-MACVer-Dec-Enc-MACGen - FCS_COP.1 / DES-Enc-Dec - FCS_COP.1 / DES-MACGen-MACVer - FCS_CKM.4 / DES-KeyErasure - FMT_MSA.2	22-26, 9, 53
6	**FCS_CKM.2**	**Cryptographic Key Distribution**	- FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	2, 8, 53
7	**FCS_CKM.3**	**Cryptographic Key Access**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.1 / DES-KeyGen_Asym - FCS_CKM.1 / DES-KeyGen-Sym - FCS_CKM.4 / RSA-PrKey-Erasure - FCS_CKM.4 / DES-Key-Erasure - FCS_CKM.4 / RSA-PubKey-Erasure - FMT_MSA.2	33, 2-10, 53
8	**FCS_CKM.4 / RSA-PrKey-Erasure**	**Cryptographic Key Destruction**	- FCS_CKM.1 / RSA-KeyGen - FMT_MSA.2	2, 53
9	**FCS_CKM.4 / DES-Key-Erasure**	**Cryptographic Key Destruction**	- FDP_ITC.1 - FCS_CKM.1 / DES-KeyGen-Asym - FCS_CKM.1 / DES-KeyGen-Sym - FMT_MSA.2	3, 4, 53
10	**FCS_CKM.4 / RSA-PubKey-Erasure**	**Cryptographic Key Destruction**	- FDP_ITC.1 - FMT_MSA.2	33, 53
11	**FCS_COP.1 / Exp-RSA-GenDigSig-PKCS1**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	33, 2, 8, 53

12	**FCS_COP.1 / Exp-RSA-GenDigSig-ISO9796-2**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	33, 2, 8, 53
13	**FCS_COP.1 / RSA-Corresp**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	33, 2, 8, 53
14	**FCS_COP.1 / Imp-RSA-GenDigSig-PKCS1**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	33, 2, 8, 53
15	**FCS_COP.1 / Imp-RSA-GenDigSig-ISO9796-2**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	33, 2, 8, 53
16	**FCS_COP.1 / Exp-RSA-Dec**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	33, 2, 8, 53
17	**FCS_COP.1 / RSA-Dec-Primitive**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PrKey-Erasure - FMT_MSA.2	33, 2, 8, 53
18	**FCS_COP.1 / Imp-RSA-VerDigSig-ISO9796-2**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PubKey-Erasure - FMT_MSA.2	33, 2, 10, 53
19	**FCS_COP.1 / Imp-RSA-VerDigSig-CV**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PubKey-Erasure - FMT_MSA.2	33, 2, 10, 53
20	**FCS_COP.1 / Exp-RSA-Enc**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PubKey-Erasure - FMT_MSA.2	33, 2, 10, 53
21	**FCS_COP.1 / RSA-Enc-Primitive**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / RSA-KeyGen - FCS_CKM.4 / RSA-PubKey-Erasure	33, 2, 10, 53

			- FMT_MSA.2	
22	**FCS_COP.1 / Imp-DES-Enc-MACGen**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / DES-KeyGen_Asym - FCS_CKM.1 / DES-KeyGen-Sym - FCS_CKM.4 / DES-Key-Erasure - FMT_MSA.2	33, 3, 4, 5, 9, 53
23	**FCS_COP.1 / Imp-DES-MACVer-Dec**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / DES-KeyGen_Asym - FCS_CKM.1 / DES-KeyGen-Sym - FCS_CKM.4 / DES-Key-Erasure - FMT_MSA.2	33, 3, 4, 5, 9, 53
24	**FCS_COP.1 / Imp-DES-MACVer-Dec-Enc-MACGen**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / DES-KeyGen_Asym - FCS_CKM.1 / DES-KeyGen-Sym - FCS_CKM.4 / DES-Key-Erasure - FMT_MSA.2	33, 3, 4, 5, 9, 53
25	**FCS_COP.1 / DES-Enc-Dec**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / DES-KeyGen_Asym - FCS_CKM.1 / DES-KeyGen-Sym - FCS_CKM.4 / DES-Key-Erasure - FMT_MSA.2	33, 3, 4, 5, 9, 53
26	**FCS_COP.1 / DES-MACGen-MACVer**	**Cryptographic Operation**	- FDP_ITC.1 - FCS_CKM.1 / DES-KeyGen_Asym - FCS_CKM.1 / DES-KeyGen-Sym - FCS_CKM.4 / DES-Key-Erasure - FMT_MSA.2	33, 3, 4, 5, 9, 53
27	**FCS_COP.1 / SHA-1**	**Cryptographic Operation**	- FDP_ITC.1 - FMT_MSA.2	33, 53
28	**FCS_RND.1**	**Quality Metric for Random Numbers**	none	---
29	**FDP_ACC.2**	**Complete Access Control**	- FDP_ACF.1	30
30	**FDP_ACF.1**	**Security Attribute Based Access Control**	- FDP_ACC.2 - FMT_MSA.3	29 (higher hierarchical element), 54
31	**FDP_DAU.1**	**Basic Data Authentication**	none	---
32	**FDP_ETC.1**	**Export of User Data without Security Attributes**	- FDP_ACC.2	29 (higher hierarchical element)
33	**FDP_ITC.1**	**Import of User Data**	- FDP_ACC.2	29 (higher

		without Security Attributes	- FMT_MSA.3	hierarchical element), 54
34	**FDP_RIP.1**	**Subset Residual Information Protection**	none	---
35	**FDP_SDI.2 / Int-PersData**	**Stored Data Integrity Monitoring and Action**	none	---
36	**FDP_SDI.2 / Int-TempData**	**Stored Data Integrity Monitoring and Action**	none	---
37	**FDP_UCT.1**	**Basic Data Exchange Confidentiality**	- FTP_ITC.1 - FDP_ACC.2	70, 29 (higher hierarchical element)
38	**FDP_UIT.1**	**Data Exchange Integrity**	- FTP_ITC.1 - FDP_ACC.2	70, 29 (higher hierarchical element)
39	**FIA_AFL.1 / PIN**	**Authentication Failure Handling**	- FIA_UAU.1	43
40	**FIA_AFL.1 / PUC**	**Authentication Failure Handling**	- FIA_UAU.1	43
41	**FIA_AFL.1 / KeyUsage**	**Authentication Failure Handling**	- FIA_UAU.1	43
42	**FIA_ATD.1**	**User Attribute Definition**	none	---
43	**FIA_UAU.1**	**Timing of Authentication**	- FIA_UID.1	47
44	**FIA_UAU.3**	**Unforgeable Authentication**	none	---
45	**FIA_UAU.4**	**Single-use Authentication Mechanisms**	none	---
46	**FIA_UAU.6**	**Re-Authenticating**	none	---
47	**FIA_UID.1**	**Timing of Identification**	none	---
48	**FIA_USB.1**	**User-Subject Binding**	- FIA_ATD.1	42
49	**FMT_LIM.1**	**Limited Capabilities**	- FMT_LIM.2	50
50	**FMT_LIM.2**	**Limited Availability**	- FMT_LIM.1	49

51	**FMT_MOF.1**	**Management of Security Functions Behaviour**	- FMT_SMF.1 - FMT_SMR.1	57, 58
52	**FMT_MSA.1**	**Management of Security Attributes**	- FDP_ACC.2 - FMT_SMF.1 - FMT_SMR.1	29 (higher hierarchical element), 57, 58
53	**FMT_MSA.2**	**Secure Security Attributes**	- ADV_SPM.1 - FDP_ACC.2 - FMT_MSA.1 - FMT_SMR.1	29 (higher hierarchical element), 52, 58 Dependency to ADV_SPM.1 given by assurance class, see below.
54	**FMT_MSA.3**	**Static Attribute Initialisation**	- FMT_MSA.1 - FMT_SMR.1	52, 58
55	**FMT_MTD.1 / General**	**Management of TSF Data**	- FMT_SMF.1 - FMT_SMR.1	57, 58
56	**FMT_MTD.1 / Init**	**Management of TSF Data**	- FMT_SMF.1 - FMT_SMR.1	57, 58
57	**FMT_SMF.1**	**Specification of Management Functions**	none	---
58	**FMT_SMR.1**	**Security Roles**	- FIA_UID.1	47
59	**FPR_UNO.1 / Sec-CryptoOp**	**Unobservability**	none	---
60	**FPR_UNO.1 / Sec-PINOp**	**Unobservability**	none	---
61	**FPT_AMT.1**	**Abstract Machine Testing**	none	---
62	**FPT_EMSEC.1**	**TOE Emanation**	none	---
63	**FPT_FLS.1**	**Failure with Preservation of Secure State**	- ADV_SPM.1	Dependency to ADV_SPM.1 given by assurance class, see below.
64	**FPT_PHP.1**	**Passive Detection of Physical Attack**	none	---

65	**FPT_PHP.3**	**Resistance to Physical Attack**	none	---
66	**FPT_RVM.1**	**Non-Bypassability of the TSP**	none	---
67	**FPT_SEP.1**	**TSF Domain Separation**	none	---
68	**FPT_TDC.1**	**Inter-TSF Basic TSF Data Consistency**	none	---
69	**FPT_TST.1**	**TSF Testing**	- FPT_AMT.1	61
70	**FTP_ITC.1**	**Inter-TSF trusted Channel**	none	---
71	**FTP_TRP.1**	**Trusted Path**	none	---

The preceding table shows that the functional component dependencies are satisfied by any functional component defined in this ST except for the following components. Justification for unsupported dependencies:

The **dependency of FAU_SAA.1 with FAU_GEN.1** (Audit Data Generation) is not applicable for the TOE. The FAU_GEN.1 component forces security relevant events to be recorded (due to dependencies with other functional security components), but for smartcards this is not achievable resp. makes no sense due to security reasons as recording of the security relevant event itself could cause a security breach. Nevertheless, the function FAU_SAA.1 is used and the specific audited events are defined in the ST independently of FAU_GEN.1.

The **dependency of FCS_COP.1 / SHA-1** with FCS_CKM.4 Cryptographic key destruction (Cryptographic key destruction) is not applicable as the SFR FCS_COP.1 / SHA-1 does not involve cryptographic keys.

8.2.3 Strength of Function Level Rationale

Due to the requirements for smartcard products intended to be used for high security applications within the Health, Identification and Banking market the level for the strength of the TOE´s security functional requirements is claimed as SOF-high. The TOE is considered as a product with critical security mechanisms which only have to be defeated by attackers possessing a high level of expertise, opportunity and resources, and whereby successful attack is judged beyond normal practicality.

8.2.4 Security Assurance Requirements Rationale

The assurance requirements of this ST defined in chap. 5.1.3 are summarized in the following table:

Assurance Requirements	Name	Type
EAL4	Methodically Designed, Tested and Reviewed	Assurance Level / Class
ADV_IMP.2	Implementation of the TSF	Higher hierarchical component
ATE_DPT.2	Testing: Low-Level Design	Higher hierarchical component
AVA_MSU.3	Analysis and Testing for Insecure States	Higher hierarchical component
AVA_VLA.4	Highly Resistant	Higher hierarchical component

8.2.4.1 Evaluation Assurance Level Rationale

Due to the requirements for smartcard products intended to be used for high security applications within the Health, Identification and Banking market the assurance level for the TOE is chosen as EAL4 augmented by ADV_IMP.2, ATE_DPT.2, AVA_MSU.3 and AVA_VLA.4. Hereby, all assurance components will be used as defined in /CC 2.3 Part3/ and /CEM 2.3 Part2/.

The evaluation assurance level of EAL4 augmented is selected for the TOE since this level provides an adequate and meaningful level of assurance for the TOE, with regard to the security of the development process of the TOE as well as with regard to the TOE´s security and resistance against attacks with high attack potential in its operational use. The chosen assurance level permits the developer to gain maximum assurance from positive security engineering based on good commercial practices and represents a sufficiently high practical level of assurance expected for the security product. Furthermore, to guarantee for a sufficiently secure product, the evaluators should have access especially to the low level design and source code, whereby the lowest assurance level for such access is given with the assurance class EAL4.

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions.

The assurance level EAL4 augmented requires knowledge of the Common Criteria evaluation scheme and process, but does not make use of specialist techniques on the part of the developer.

A more detailed rationale for the chosen augmentations of the evaluation assurance class EAL4 is provided in the following chap. 8.2.4.2.

8.2.4.2 Assurance Augmentations Rationale

The following section gives reason for the choice of the assurance components augmenting the evaluation assurance class EAL4.

Apriori, the assurance components ADV_IMP.2, ATE_DPT.2, AVA_MSU.3 and AVA_VLA.4 are chosen with respect to the common understanding of security requirements for high security smartcards intended to be used in the framework of Health, Identification and Banking applications.

In detail, the following deliberations are of interest:

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.

The assurance component ADV_IMP.2 is a higher hierarchical component to EAL4, which only requires ADV_IMP.1 „Subset of the implementation of the TSF“.

The augmentation by ADV_IMP.2 is chosen for the following reason: It is important for the TOE and its assurance that the evaluator evaluates the implementation representation of the *entire* TSF to determine that the SFRs as defined in the ST are addressed by the representation of the TSF and that the implementation representation is an accurate and complete instantiation of the TOE´s SFRs. This provides a direct correspondence between the TOE´s SFRs and the implementation representation, in addition to the pairwise correspondences required by the ADV_RCR family.

ATE_DPT.2 Testing: Low-Level Design

Testing of the TSFs and their internal structure is done with the objective to counter the risk of missing an error or malicious code in the development of the TOE. Testing that exercises specific internal interfaces can provide assurance not only that the TSF exhibits the desired external security behaviour, but also that this behaviour stems from correctly operating internal mechanisms.

The assurance component ATE_DPT.2 is a higher hierarchical component to EAL4, which only requires ATE_DPT.1 „Testing: high-level design“.

It is important for the TOE and its assurance that testing of the TSFs is not only done on basis of the high-level description of the internal workings of the TSF (level of the subsystems) in order to demonstrate the absence of any flaws and to provide assurance that the TSF subsystems have been correctly realised. Moreover, the testing of the TSFs shall cover tests on the modules of the TSFs providing a low-level description of the internal workings of the TSF with the goal to demonstrate the absence of any flaws and to provide assurance that the TSF modules have been correctly realised. The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and *low-level design*.

AVA_MSU.3 Analysis and Testing for Insecure States

Misuse investigates whether the TOE can be configured or used in a manner that is insecure but that an administrator or user of the TOE would reasonably believe to be secure.

The assurance component AVA_MSU.3 is a higher hierarchical component to EAL4, which only requires AVA_MSU.2 „Validation of analysis".

The augmentation by AVA_MSU.3 is chosen according to the requirements in the protection profiles /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/ and /PP SSCD Type2/. Due to the nature of the TOE´s intended application, the TOE may be issued to users and may not be directly under the control of trained and dedicated administrators. As a result, it is imperative that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In AVA_MSU.3, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met, and this analysis is validated and confirmed through testing by the evaluator.

AVA_VLA.4 Highly Resistant

According to the definition of the TOE, it must be shown to be highly resistant to penetration attacks. This is due to the fact that the TOE can be placed in a hostile environment.

This assurance requirement is achieved by the assurance component AVA_VLA.4. 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 and is presumed to have a high level of technical sophistication.

The assurance component AVA_VLA.4 is a higher hierarchical component to EAL4, which only requires AVA_VLA.2 „Independent vulnerability analysis".

The augmentation by AVA_VLA.4 is chosen according to the requirements in the protection profiles /PP9911/, /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/ and /PP SSCD Type2/. For AVA_VLA.4, a systematical vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. Hereby, the analysis shall provide a justification that the analysis completely addresses the TOE deliverables. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks performed by attackers possessing a high attack potential.

8.2.5 Security Assurance Requirements Dependencies

The security assurance requirements specified by this ST are drawn from the assurance class EAL4 with its augmentation by the higher hierarchical components ADV_IMP.2, ATE_DPT.2, AVA_MSU.3 and AVA_VLA.4.

EAL4 is asserted to be a known set of assurance components for which all dependencies are satisfied. For the components of the augmentation the following deliberation shows that all further dependencies resulting from the augmentation are satisfied:

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, and so these dependencies are satisfied.

ATE_DPT.2 has dependencies with ADV_HLD.2 „Security enforcing high-level design", ADV_LLD.1 „Descriptive low-level design" and ATE_FUN.1 „Functional testing". All these dependencies are satisfied by EAL4.

AVA_MSU.3 has dependencies with ADO_IGS.1 "Installation, generation, and start-up procedures", ADV_FSP.1 "Informal functional specification", AGD_ADM.1 "Administrator guidance" and AGD_USR.1 "User guidance". All these dependencies are satisfied by EAL4.

AVA_VLA.4 has dependencies with ADV_FSP.1 „Informal functional specification", ADV_HLD.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" and AGD_USR.1 „User Guidance". All these dependencies are satisfied by EAL4.

8.2.6 Security Requirements – Mutual Support and Internal Consistency

The following part of the security requirements rationale shows that the set of security requirements for the TOE consisting of the security assurance requirements (SARs) and the security functional requirements (SFRs) together forms a mutually supportive and internally consistent whole.

The analysis of the TOE´s security requirements with regard to their mutual support and internal consistency demonstrates:

- The assurance class EAL4 is an established set of mutually supportive and internally consistent assurance requirements.
- The dependency analysis for the additional assurance components in chap. 8.2.5 shows that the assurance requirements are mutually supportive and internally consistent as all (additional) dependencies are satisfied and no inconsistency appears.
- The dependency analysis in chap. 8.2.2 for the security functional requirements of the TOE-IC and the TOE-ES shows that the basis for mutual support and internal consistency between all defined functional requirements is satisfied. All dependencies between the chosen functional components are analysed, and non-dissolved dependencies are appropriately explained.

  The mutual support and internal consistency of the functional requirements is shown for the TOE-IC relevant SFRs in the scope of the CC evaluation of the TOE-IC resp. in the correlated ST and for the TOE-ES relevant SFRs in chap. 8.2.1.2 and 8.2.1.3 within the mapping of the security objectives to the SFRs.

  Concerning the SFRs of the TOE-ES, the SFRs have been chosen under consideration of the protection profiles /PP9911/, /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/ and /PP SSCD Type2/. The SFRs drawn from /PP9911/ build as shown in the rationale of the protection profile a mutually supportive and internally consistent whole. The additional SFRs suitably supplement the SFRs of /PP9911/

and do not lead to any inconsistency or any weakness as can be seen from the deliberations in chap. 8.2.1.2 and 8.2.1.3.

- All operations (assignment, selection, iteration and refinement) conducted on the CC functional components lead to a consistent and meaningful whole.

  For the TOE-IC relevant SFRs the evidence is done within the scope of the CC evaluation of the TOE-IC resp. in the correlated ST.

  For the TOE-ES relevant SFRs the following deliberations are important. First, all operations on the chosen SFRs are done with the target to reflect correctly and completely the security functionality provided by the TOE whereat the operations in this ST take the operations already done within the protection profiles /PP-eHC/, /PP-HPC/, /PP-SMC/, /PP SSCD Type3/ and /PP SSCD Type2/ into account. Furthermore, all assignment, selection, iteration and refinement operations are conducted in such a way that they do not contradict each other and build an internally consistent security system. In particular, the iterations of the functional components for cryptographic support, FCS_CKM and FCS_COP, are necessary to differentiate between the different cryptographic algorithms and mechanisms of the TOE. The iteration of the functional component FIA_AFL is necessary to differentiate between the different authentication mechanisms provided by the TOE.

- Inconsistency between functional and assurance requirements can only arise if there are functional-assurance dependencies which are not met, a possibility which has been shown not to arise in chap. 8.2.2. Furthermore, as discussed in chap. 8.2.4, the chosen assurance components are adequate for the functionality of the TOE what underlines that the assurance requirements and security functional requirements support each other and that there are no inconsistencies between these two groups of security requirements.

8.3 TOE Summary Specification Rationale

According to the requirements of Common Criteria, /CC 2.3 Part1/ and /CC 2.3 Part3/, the TOE summary specification rationale demonstrates that the TOE security functions (TSFs) and assurance measures are suitable to meet the TOE security requirements. In detail, the following will be demonstrated:

- the combination of the specified TOE´s IT security functions work together so as to satisfy the TOE security functional requirements
- the strength of the TOE function claims made are valid, or assertions that such claims are unnecessary are valid
- the claim that the stated assurance measures are compliant with the assurance requirements is justified

8.3.1 Security Functions Rationale

The following section demonstrates that the set and combination of the defined TOE security functions (TSFs) is suitable to satisfy the identified TOE security functional requirements (SFRs). Furthermore, this section shows that each of the TSFs is related to at least one security functional requirement.

8.3.1.1 Security Functional Requirements for the TOE-IC – TOE Security Functions

The SFRs for the TOE-IC of chap. 5.1.1.1 are related to the TSFs of the TOE-IC defined in chap. 6.1.1. The mapping of the SFRs for the TOE-IC to the relevant TSFs is done within the CC evaluation of the IC resp. within the associated Security Target.

8.3.1.2 Security Functional Requirements for the TOE-ES – TOE Security Functions

The SFRs for the TOE-ES of chap. 5.1.1.2 are related to the TSFs of the TOE-ES defined in chap. 6.1.2. The mapping of the SFRs for the TOE-ES to the relevant TSFs is done in the following.

The tables below give an overview of which TSFs of the TOE-ES contribute to the satisfaction of the SFRs for the TOE-ES.

Security Functional Requirements / TOE Security Functions	F.ACS	F.IA_AKEY	F.IA_SKEY	F.IA_PWD	F.DATA_INT	F.EX_CONF	F.EX_INT	F.RIP	F.FAIL_PROT	F.SIDE_CHAN	F.SELFTEST	F.CRYPTO	F.RSA_KEYGEN	F.GEN_DIGSIG	F.VER_DIGSIG	F.RSA_ENC	F.RSA_DEC	TSFs of IC
FAU_SAA.1		x	x	x	x	x	x		x		x							
FCS_CKM.1 / RSA-KeyGen	x									x		(x)	x					(x)
FCS_CKM.1 / DES-KeyGen-Asym		x	x							x		x						(x)
FCS_CKM.1 / DES-KeyGen-Sym		x	x							x		x						(x)
FCS_CKM.1 / Key-Derivation		x	x							x		x						(x)
FCS_CKM.2												(x)	x					
FCS_CKM.3	x																	
FCS_CKM.4 / RSA-PrKey-Erasure	(x)							x					x					
FCS_CKM.4 / DES-Key-Erasure	(x)							x										
FCS_CKM.4 / RSA-PubKey-Erasure	(x)							x										
FCS_COP.1 / Exp-RSA-GenDigSig-PKCS1	(x)									x		(x)		x				(x)
FCS_COP.1 / Exp-RSA-GenDigSig-ISO9796-2	(x)									x		(x)		x				(x)
FCS_COP.1 / RSA-Corresp	(x)									x		(x)		x				(x)
FCS_COP.1 / Imp-RSA-GenDigSig-PKCS1	(x)									x		(x)		x				(x)
FCS_COP.1 / Imp-RSA-GenDigSig-ISO9796-2	(x)									x		(x)		x				(x)
FCS_COP.1 / Exp-RSA-Dec	(x)									x		(x)					x	(x)
FCS_COP.1 / RSA-Dec-Primitive	(x)									x		(x)					x	(x)
FCS_COP.1 / Imp-RSA-VerDigSig-ISO9796-2	(x)											(x)			x			
FCS_COP.1 / Imp-RSA-VerDigSig-CV	(x)											(x)			x			
FCS_COP.1 / Exp-RSA-Enc	(x)											(x)				x		
FCS_COP.1 / RSA-Enc-Primitive	(x)											(x)				x		
FCS_COP.1 / Imp-DES-Enc-MACGen	(x)									x		x						(x)
FCS_COP.1 / Imp-DES-MACVer-Dec	(x)									x		x						(x)
FCS_COP.1 / Imp-	(x)									x		x						(x)

DES-MACVer-Dec-Enc-MACGen																		
FCS_COP.1 / DES-Enc-Dec	**(x)**									**x**		**x**						**(x)**
FCS_COP.1 / DES-MACGen-MACVer	**(x)**									**x**		**x**						**(x)**
FCS_COP.1 / SHA-1										**x**		**x**						**(x)**
FCS_RND.1										**x**		**x**						**(x)**
FDP_ACC.2	**x**																	
FDP_ACF.1	**x**																	
FDP_DAU.1												**x**	**x**					
FDP_ETC.1	**x**					**x**	**x**											
FDP_ITC.1	**x**					**x**	**x**											
FDP_RIP.1								**x**										
FDP_SDI.2 / Int-PersData					**x**													
FDP_SDI.2 / Int-TempData					**x**													
FDP_UCT.1						**x**						**x**						**(x)**
FDP_UIT.1							**x**					**x**						**(x)**
FIA_AFL.1 / PIN				**x**														
FIA_AFL.1 / PUC				**x**														
FIA_AFL.1 / KeyUsage		**x**	**x**															
FIA_ATD.1	**x**																	
FIA_UAU.1	**x**																	
FIA_UAU.3		**x**	**x**	**x**														
FIA_UAU.4		**x**	**x**									**x**						
FIA_UAU.6		**x**	**x**															
FIA_UID.1	**x**																	
FIA_USB.1	**x**																	
FMT_LIM.1																		**x**
FMT_LIM.2																		**x**
FMT_MOF.1	**x**																	
FMT_MSA.1	**x**																	
FMT_MSA.2	**x**																	
FMT_MSA.3	**x**																	
FMT_MTD.1 / General	**x**																	
FMT_MTD.1 / Init	**x**																	
FMT_SMF.1	**x**																	
FMT_SMR.1	**x**																	
FPR_UNO.1 / Sec-CryptoOp		**x**	**x**									**x**	**x**	**x**			**x**	
FPR_UNO.1 / Sec-PINOp				**x**														
FPT_AMT.1											**x**							
FPT_EMSEC.1										**x**								**(x)**
FPT_FLS.1									**x**									
FPT_PHP.1																		**(x)**
FPT_PHP.3										**x**								**(x)**
FPT_RVM.1											**x**							
FPT_SEP.1	**x**																	
FPT_TDC.1		**x**	**x**	**x**		**x**	**x**					**x**		**x**	**x**	**x**	**x**	
FPT_TST.1											**x**							
FTP_ITC.1		**x**	**x**			**x**	**x**											
FTP_TRP.1				**x**														

Note:

X directly contributing TSF

(X) supporting TSF

The detailed description and analysis of the TOE Security Functions in chap. 6.1 demonstrate how the defined functions work together and support each other. Furthermore, this description shows that no inconsistencies exist.

The deliberations above support this result. Additionally, for the TSFs of the underlying IC as defined in /ST-ICPhilips/ such analysis is done in the scope of the CC evaluation of the IC resp. within the correlated ST.

In the following, for each SFR of the TOE it will be explained why and how the TSFs listed in the preceding tables meet the respective SFR.

The rationale here is presented in form of tables. The full rationale as given in the TOE´s Security Target is not intended to be published and hence not part of the ST-Lite.

8.3.2 Assurance Measures Rationale

The assurance measures of the developer as mentioned in chap. 6.3 are considered to be suitable and sufficient to meet the CC assurance level EAL4 augmented by ADV_IMP.2, ATE_DPT.2, AVA_MSU.3 and AVA_VLA.4 as claimed in chap. 5.1.3. Especially the deliverables listed in chap. 6.3 are seen to be suitable and sufficient to document the fulfillment of the assurance requirements in detail.

As the development and production process of the TOE is very complex and a great number of assurance measures are implemented by the developer, a detailed description of these measures beyond the information given in chap. 2.2 and 2.3 as well as a detailed mapping of the assurance measures to the assurance requirements is not in the scope of this ST.

8.3.3 TOE Security Functions – Mutual Support and Internal Consistency

The detailed description of the TOE Security Functions in chap. 6.1 demonstrates how the defined functions work together and support each other. Furthermore, this description shows that no inconsistencies exist.

The deliberations in chap. 8.3.1 support this result. Additionally, for the TSFs of the TOE-IC as defined in chap. 6.1.1 such analysis is done in the scope of the IC evaluation resp. within the correlated ST.

8.3.4 Strength of Functions

The selected Strength of Functions level for the TOE´s security functions of SOF-high is consistent with the security objectives for the TOE, as the TOE is considered as a security product with critical security mechanisms which shall be resistant against attacks with high attack potential.

8.4 Extensions

8.4.1 FCS_RND Generation of Random Numbers

Family behaviour

This family describes the functional requirements for random number generation used for cryptographic purposes.

In order to ensure that a random number generator can be employed for different cryptographic purposes, the random number generation must assure that the generated random numbers posses certain properties. Typical properties include assurance that a given quality metric (e.g. minimum entropy) is achieved or that an implementation meets a given standard.

Component levelling

FCS_RND Generation of random numbers	1

FCS_RND.1 Quality Metric for Random Numbers requires that random numbers meet a defined quality metric.

Management: FCS_RND.1

There are no management activities foreseen.

Audit: FCS_RND.1

There are no actions defined to be auditable.

FCS_RND.1 Quality Metric for Random Numbers

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet [assignment: *a defined quality metric*].

Hierarchical to: No other components

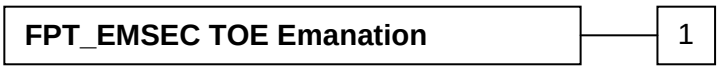
Dependencies: No dependencies.

8.4.2 FPT_EMSEC TOE Emanation

Family behaviour

This family defines requirements to mitigate intelligible emanations.

Component levelling



FPT_EMSEC.1 TOE Emanation has two constituents:

FPT_EMSEC.1.1 Limit of Emissions requires to not emit intelligible emissions enabling access to TSF data or user data.

FPT_EMSEC.1.2 Interface Emanation requires not emit interface emanation enabling access to TSF data or user data.

Management: FPT_EMSEC.1

There are no management activities foreseen.

Audit: FPT_EMSEC.1

There are no actions defined to be auditable.

FPT_EMSEC.1 TOE Emanation

FPT_EMSEC.1.1	The TOE shall not emit [*assignment: types of emissions*] in excess of [*assignment: specified limits*] enabling access to [*assignment: list of types of TSF data*] and [*assignment: list of types of user data*].
FPT_EMSEC.1.2	The TSF shall ensure [assignment: type of users] are unable to use the following interface [assignment: type of connection] to gain access to [assignment: list of types of TSF data] and [assignment*: list of types of user data]*.

Hierarchical to: No other components.

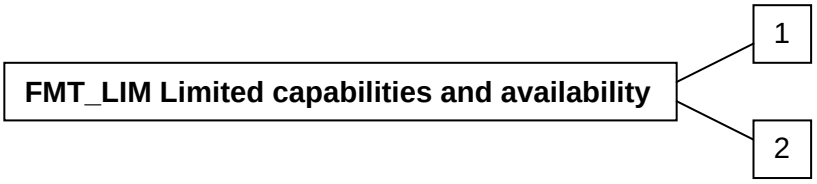
Dependencies: No dependencies.

8.4.3 FMT_LIM Limited Capabilities and Availability

Family behaviour

This family defines requirements that limits the capabilities and availability of functions in a combined manner. Note that FDP_ACF restricts the access to functions whereas the Limited capability of this family requires the functions themselves to be designed in a specific manner.

Component levelling



FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.

FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (refer to Limited capabilities (FMT_LIM.1)). This can be achieved, for instance, by removing or by disabling functions in a specific phase of the TOE's life-cycle.

Management: FMT_LIM.1, FMT_LIM.2

There are no management activities foreseen.

Audit: FMT_LIM.1, FMT_LIM.2

There are no actions defined to be auditable.

FMT_LIM.1 Limited capabilities

FMT_LIM.1.1 The TSF shall be designed in a manner that limits their capabilities so that in conjunction with „Limited availability (FMT_LIM.2)" the following policy is enforced [assignment: *Limited capability and availability policy*].

Hierarchical to: No other components.

Dependencies: FMT_LIM.2.

FMT_LIM.2 Limited availability

FMT_LIM.2.1 The TSF shall be designed in a manner that limits their availability so that in conjunction with „Limited capabilities (FMT_LIM.1)" the following policy is enforced [assignment: Limited capability and availability policy].

Hierarchical to: No other components.

Dependencies: FMT_LIM.1.

Reference

I Bibliography

/CC 2.3 Part1/
Title: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model
Identification: CCIMB-2005-08-001
Version: Version 2.3
Date: August 2005
Author: CC Project Sponsoring Organisations CSE, SCSSI, BSI, NLNCSA, CESG, NIST, NSA

/CC 2.3 Part2/
Title: Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements
Identification: CCIMB-2005-08-002
Version: Version 2.3
Date: August 2005
Author: CC Project Sponsoring Organisations CSE, SCSSI, BSI, NLNCSA, CESG, NIST, NSA

/CC 2.3 Part3/
Title: Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements
Identification: CCIMB-2005-08-003
Version: Version 2.3
Date: August 2005
Author: CC Project Sponsoring Organisations CSE, SCSSI, BSI, NLNCSA, CESG, NIST, NSA

/CEM 0.6 Part1/
Title: Common Methodology for Information Technology Security Evaluation, Part 1: Introduction and General Model
Identification: CEM99/045
Version: Draft 0.6
Date: Jan. 1997
Author: CC Project Sponsoring Organisations CSE, SCSSI, BSI, NLNCSA, CESG, NIST, NSA

/CEM 2.3 Part2/
Title: Common Methodology for Information Technology Security Evaluation, Part 2: Evaluation Methodology
Identification: CCIMB-2005-08-004
Version: Version 2.3
Date: August 2005
Author: CC Project Sponsoring Organisations CSE, SCSSI, BSI, NLNCSA, CESG, NIST, NSA

/AIS32/
Title: Übernahme international abgestimmter CC Interpretationen
Identification: AIS 32
Date: 02.07.2001
Publisher: Bundesamt für Sicherheit in der Informationstechnik

/PP9806/
Title: Protection Profile - Smartcard Integrated Circuit
Identification: Registered at the French Certification Body (DCSSI) under the number PP/9806
Version: Version 2.0
Date: Sept. 1998
Author: Motorola Semiconductors, Philips Semiconductors, Service Central de la Securite des Systemes d´Information, Siemens AG Semiconductors, ST Microelectronics, Texas-Instruments Semiconductors

/PP9911/
Title: Protection Profile - Smartcard Integrated Circuit with Embedded Software
Identification: Registered at the French Certification Body (DCSSI) under the number PP/9911
Version: Version 2.0
Date: June 1999
Author: Atmel Smart Card ICs, Bull-SC&T, De la Rue – Card Systems, Eurosmart, Gemplus, Giesecke & Devrient GmbH, Hitachi Europe Ltd, Infineon Technologies AG, Microelectronica Espana, Motorola SPS, NEC Electronics, Oberthur Smart Card, ODS, ORGA Kartensysteme GmbH, Philips Semiconductors Hamburg, Schlumberger Cards Devision, Service Central de la Securite des Systemes d´Information, ST Microelectronics

/BSI-PP-0002/
Title: Smartcard IC Platform Protection Profile
Identification: Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0002
Version: Version 1.0
Date: July 2001
Author: Atmel Smart Card ICs, Hitachi Europe Ltd, Infineon Technologies AG, Philips Semiconductors

/CompPP9806-BSIPP0002/
Title: Assessment on the Substitution of an Evaluation based on PP/9806 by an Evaluation based on BSI-PP-0002-2001
Version: Version 1.1
Date: May 2002
Publisher: Bundesamt für Sicherheit in der Informationstechnik (BSI)

/DS-ICPhilips/
Title: Data Sheet: SmartMX – P5CC036 Secure Smart Card Controller
Version: Revision 3.0
Date: Sept. 21st 2004
Publisher: Philips Semiconductors GmbH

/IS-ICPhilips/
Title: Instruction Set SmartMX-Family, Secure Smart Card Controller, Objective Specification
Version: Revision 1.0
Date: May 9th 2003
Publisher: Philips Semiconductors GmbH

/UG-ICPhilips/
Title: Guidance, Delivery and Operation Manual: Evaluation of the Philips P5CC036V1D Secure Smart Card Controller
Version: Revision 1.0
Date: March 18th 2005
Publisher: Philips Semiconductors GmbH

/ST-ICPhilips/
Title: Security Target - Evaluation of the Philips P5CC036V1D Secure Smart Card Controller
Identification: BSI-DSZ-CC-0293
Version: Version 1.0
Date: March 18th 2005
Publisher: Philips Semiconductors GmbH

/ETRLite-ICPhilips/
Titel: BSI-DSZ-CC-0293: ETR-lite for composition according to AIS 36
Version: Version 1.0
Date: July 6th 2005
Publisher: T-Systems GEI GmbH

/ConfListPhilips/
Title: Customer specific Appendix of the Configuration List for composite evaluation with ORGA (P5CC036V1D)
Version: Version 1.0
Publisher: Philips Semiconductors GmbH

/ISO9796-2/
Title: Information Technology – Security Techniques – Digital Signature Schemes Giving Message Recovery – Part 2: Mechanisms Using a Hash Function
Identification: ISO/IEC 9796-2

Version: First Edition
Date: 1997
Publisher: ISO / IEC

/ISO9798-3/
Title: Information Technology – Security Techniques – Entity Authentication Mechanisms – Part 3: Entity Authentication Using a public key algorithm
Identification: ISO/IEC 9798-3
Version: Second Edition
Date: 1998
Publisher: ISO / IEC

/ISO 7816-4/
Title: Integrated circuit(s) cards with contacts. Part 4: Interindustry commands for interchange
Identification: ISO/IEC 7816-4
Version: First edition
Date: September 1.1995
Publisher: International Organization for Standardization/International Electrotechnical Commission

/ISO 7816-8/
Title: Integrated circuit(s) cards with contacts. Part 8: Interindustry commands for interchange
Identification: ISO/IEC FDIS 7816-8
Date: June 1998
Publisher: International Organization for Standardization/International Electrotechnical Commission

/ISO 7816-9/
Title: Integrated circuit(s) cards with contacts. Part 9: Enhanced interindustry commands
Identification: ISO/IEC 7816-9
Version: First Edition
Date: Sept. 2000
Publisher: International Organization for Standardization/International Electrotechnical Commission

/SHA-1/
Title: Secure Hash Standard (SHS)
Identification: FIPS Publication 180-2
Date: August 2002
Publisher: National Institute of Standards and Technology (NIST)

/FIPS 46-3/
Title: Data Encryption Standard (DES)
Identification: FIPS Publication 46-3

Date: October 1999
Publisher: National Institute of Standards and Technology (NIST)

/ANSI X9.52/
Title: Triple Data Encryption Algorithm Modes of Operation
Identification: ANSI X9.52
Date: 1998
Publisher: American National Standards Institute (ANSI)

/PKCS1/
Title: PKCS #1 v2.1: RSA Cryptography Standard
Date: June 2002
Publisher: RSA Laboratories

/ISO 11770-3/
Title: Information Technology – Security Techniques – Key Management – Part 3: Mechanisms Using Asymmetric Techniques
Identification: ISO/IEC 11770-3
Date: 1996
Publisher: ISO/IEC

/ISO 10118-2/
Title: Information Technology – Security Techniques – Hash Functions – Part 2: Hash Functions Using an n-Bit Block Cipher Algorithm
Identification: ISO/IEC 10118-2
Date: 1994
Publisher: ISO/IEC

/ANSI X9.19/
Title: Financial Institution Retail Message Authentication
Identification: ANSI X9.19
Date: 1996
Publisher: American National Standards Institute (ANSI)

/ANSI X9.63/
Title: Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography
Identification: ANSI X9.63
Date: 2001
Publisher: American National Standards Institute (ANSI)

/eHC1/
Title: Die Spezifikation der elektronischen Gesundheitskarte, Teil 1: Kommandos, Algorithmen und Funktionen der Betriebssystem-Plattform

Version: Version 1.1.0
Date: 07.02.2006
Publisher: gematik mbH

/eHC2/
Title: Die Spezifikation der elektronischen Gesundheitskarte, Teil 2: Anwendungen und anwendungsspezifische Strukturen
Version: Version 1.1.0
Date: 07.02.2006
Publisher: gematik mbH

/HPC-SMC1/
Title: German Health Professional Card and Security Module Card, Part 1: Commands, Algorithms and Functions of the COS Platform
Version: Version 2.1.0
Date: 21.02.2006
Publisher: BundesÄrzteKammer, Kassenärztliche Bundesvereinigung, BundesZahnÄrzteKammer, BundesPsychotherapeutenKammer, Kassenzahnärztliche Bundesvereinigung, Werbe- und Vertriebsgesellschaft Deutscher Apotheker mbH

/HPC-SMC2/
Title: German Health Professional Card and Security Module Card, Part 2: HPC Applications and Functions
Version: Version 2.1.0
Date: 21.02.2006
Publisher: BundesÄrzteKammer, Kassenärztliche Bundesvereinigung, BundesZahnÄrzteKammer, BundesPsychotherapeutenKammer, Kassenzahnärztliche Bundesvereinigung, Werbe- und Vertriebsgesellschaft Deutscher Apotheker mbH

/SigG01/
Title: Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften
Identification: Bundesgesetzblatt Nr. 22, S. 876
Date: 16.05.2001
Publisher: Dtsch. Bundestag

/SigV01/
Title: Verordnung zur elektronischen Signatur
Identification: Bundesgesetzblatt Nr. 509, S. 3074
Date: 16.11.2001
Publisher: Dtsch. Bundestag

/ECDir/

Title: Richtlinie 1999/93/EG des Europäischen Parlaments und des Rates vom 13. Dezember 1999 über gemeinschaftliche Rahmenbedingungen für elektronische Signaturen
Identification: Amtblatt der Europäischen Gemeinschaften, L13/12-L13/20
Date: 19.01.2001
Publisher: Europäisches Parlament und Rat der Europäischen Union

/ALGCAT/
Title: Geeignete Algorithmen zur Erfüllung der Anforderungen nach §17 Abs.1 bis 3 SigG vom 22. Mai 2001 in Verbindung mit Anlage 1 Anschnitt I Nr. 2 SigV vom 22. Nov. 2001
Identification: Bundesanzeiger Nr. 58, S. 1913-1915
Date: 23.03.2006
Publisher: Bundesnetzagentur

/PP-eHC/
Title: Protection Profile – electronic Health Card (eHC) – elektronische Gesundheitskarte (eGK)
Identification: BSI-PP-0020
Version: 1.10
Date: Feb. 16th 2006
Publisher: Bundesamt für Sicherheit in der Informationstechnik (BSI)

/PP-HPC/
Title: Protection Profile – Health Professional Card (HPC) – Heilberufsausweis (HBA)
Identification: BSI-PP-0018
Version: 1.0
Date: Dec. 12th 2005
Publisher: Bundesamt für Sicherheit in der Informationstechnik (BSI)

/PP-SMC/
Title: Protection Profile – Security Module Card (SMC)
Identification: BSI-PP-0019
Version: 1.0
Date: Feb. 1st 2006
Publisher: Bundesamt für Sicherheit in der Informationstechnik (BSI)

/PP SSCD Type3/
Title: Protection Profile – Secure Signature-Creation Device Type 3 “EAL 4+”
Identification: BSI-PP-0006-2002
Version: Version 1.05
Date: July 25th 2001
Publisher: CEN/ISSS – Information Society Standardization System, Workshop on Electronic Signatures

/PP SSCD Type2/

Title:	Protection Profile – Secure Signature-Creation Device Type 2 “EAL 4+”
Identification:	BSI-PP-0005-2002
Version:	Version 1.04
Date:	July 25th 2001
Publisher:	CEN/ISSS – Information Society Standardization System, Workshop on Electronic Signatures

II Summary of abbreviations

A.x	Assumption
AC	Access Condition
AID	Application Identifier
ALW	Always
AM	Access Mode
AR	Access Rule
AS	Application Software
ATR	Answer To Reset
AUT	Key Based Authentication
BS	Basic Software
CC	Common Criteria
CH	Cardholder
CHV	Cardholder Verification
DES	Data Encryption Standard
DF	Dedicated File
DFA	Differential Fault Analysis
DPA	Differential Power Analysis
EAL	Evaluation Assurance Level
EF	Elementary File
EHC	Electronic Health Card
ES	Embedded Software
HPC	Health Professional Card
IC	Integrated Circuit
IFD	Interface Device
MAC	Message Authentication Code
MF	Master File
O.x	Security Objective
OS	Operating System
PAR	Partial Access Rule
P.x	Organisational Security Policy
PIN	Personal Identification Number
PP	Protection Profile
PUC	PIN Unblocking Code
PW	Password
PWD	Password Based Authentication
RSA	Rivest-Shamir-Adleman Algorithm
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SM	Secure Messaging
SMC	Security Module Card

SOF	Strength of Functions
SPA	Simple Power Analysis
SPM	TOE Security Policy Model
SSC	Send Sequence Counter
SSCD	Secure Signature-Creation Device
ST	Security Target
TA	Timing Analysis
T.x	Threat
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Function
TSP	TOE Security Policy

III Glossary

For explanation of technical terms refer to the following documents:

/PP9911/, Annex A

/BSI-PP-0002/, Chap. 8.7