Security Target
BSI-DSZ-CC-0170

Version 1.2

February 20th, 2002

Evaluation of the Philips P8WE6004V0D Secure 8-bit Smart Card Controller

Developed and provided by

Philips Semiconductors, Business Unit Identification

**According to the
Common Criteria for Information Technology
Evaluation (CC) at Level EAL5 augmented**

by

**Philips Semiconductors GmbH
Stresemannallee 101
22505 Hamburg**

Document Information

Document History

Version	Date	Changes	Remarks
Version 1.2	20.02.2002	Public version	

Latest version is: Version 1.2 (February 20th, 2002)

Document Invariants

Name	Value (to be edited)	Test Output (to copy)
file name and length	Automatically	st_6004_v1_2.doc (409088 Byte)
latest version	Version 1.2	Version 1.2
date of this version	February 20th, 2002	February 20th, 2002
classification	Security Document – Strictly Confidential	Security Document – Strictly Confidential
product (short)	Philips P8WE6004V0D smart card controller	Philips P8WE6004V0D smart card controller
product (long)	Philips P8WE6004V0D Secure 8-bit Smart Card Controller	Philips P8WE6004V0D Secure 8-bit Smart Card Controller
developer (long)	Philips Semiconductors, Business Unit Identification	Philips Semiconductors, Business Unit Identification
developer (short)	Philips	Philips
registration number	BSI-DSZ-CC-0170	BSI-DSZ-CC-0170
list of authors	Hans-Gerd Albertsen	Hans-Gerd Albertsen
certific. body (short)	BSI	BSI
certific. body (long)	Bundesamt für Sicherheit in der Informationstechnik (BSI)	Bundesamt für Sicherheit in der Informationstechnik (BSI)

Table of Contents

1 ST Introduction

The chapter *ST Introduction* is divided into the following sections:

ST Identification
ST Overview
CC Conformance

1.1 ST Identification

This Security Target (st_6004_v1_1.doc, Version 1.2, February 20th, 2002) refers to the "Philips P8WE6004V0D Secure 8-bit Smart Card Controller" (TOE) for a Common Criteria evaluation.

1.2 ST Overview

The TOE is the hardware of the microcontroller chip Philips P8WE6004V0D smart card controller produced by Philips and the test software located in the Test-ROM of the microcontroller. The TOE includes the Datasheet of the Philips P8WE6004V0D smart card controller, which contains a description of the architecture, the secure configuration of the chip by the user and the instruction set.
The Philips P8WE6004V0D smart card controller supports the usage for a wide range of security applications within the information technology. The TOE is embedded in a micro-module or another sealed package. The micro-module is embedded into a credit card sized plastic card.
The E2PROM makes the TOE ideal for application requiring non-volatile data storage, including smart cards and portable data banks. Security functions protect data in the on-chip ROM, E2PROM and RAM. In particular when being used in the banking and finance market or in electronic commerce applications the smart card must provide security. Hence the TOE shall

- maintain the integrity and the confidentiality of data stored in the memory of the TOE and
- maintain the integrity, the correct operation and the confidentiality of security functions (security mechanisms and associated functions) provided by the TOE.

This is ensured by the construction of the TOE and by security functions provided by the TOE. Usually the smart card is assigned to a single individual only but may store and process secrets of the system too. So, the TOE must meet security requirements to be applied to security modules. The "Philips P8WE6004V0D Secure 8-bit Smart Card Controller" (TOE) mainly provides a hardware platform for a smart card with

- functions to calculate the Data Encryption Algorithm (DEA) resistant to Differential Power Analysis (DPA) attacks,
- a random number generator and

- mode control regarding a test mode and a user mode.

In addition several security features independently implemented in hardware or controlled by software will be provided to ensure proper operations and integrity and confidentiality of stored data. This includes for example measures for memory protection and sensors to allow operations only under specified conditions.

Regarding the life cycle of the smartcard the development and the production phase of the IC with its dedicated software as described for the Target of Evaluation (TOE) is part of the evaluation. This is based on

- the description of the security environment of the TOE including the assets to be protected and the threats to be countered by the TOE and by the operational environment during the development, production and user phases,
- the description of the security objectives for the TOE and for its environment in terms of integrity and confidentiality of application data and programs, protection of the TOE and associated documentation during the development and production phases and
- the specification of the security requirements which includes the TOE security functional requirements and the TOE security assurance requirements.

1.3 CC Conformance

The Evaluation is based upon

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 2.1, August 1999 and ISO 15408-1:1999

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 2.1, August 1999 and ISO 15408-2:1999

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 2.1, August 1999 and ISO 15408-3:1999

For the evaluation the following methodology will be used

[4] Common Methodology for Information Technology Security Evaluation CEM-99/045 Part 2: Evaluation Methodology, Version 1.0, August 1999

The chosen level of assurance is

EAL 5 augmented. The minimum strength level for the TOE security functions is SOF-high (Strength of Functions High).

This security Target claims the following CC conformances:

Part 2 extended, Part 3 augmented.

The Security Target is conformant to the PP/9806 Version 2.0 (see section 7 for details).

The level of evaluation and the functionality of the TOE are chosen in order to allow the confirmation that the TOE is suitable for use within devices compliant with the German Digital Signature Law.

2 TOE Description

The chapter TOE Description is divided into the following sections:

Product Type
Smart Card Product Life-Cycle
TOE Environment
TOE Logical Phases
TOE Intended usage
General IT features of the TOE

2.1 Product Type

The TOE named Philips P8WE6004V0D smart card controller is a single chip secured 8-Bit microcontroller with firmware, manufactured in an advanced CMOS process by Philips. The device is developed for most high-end safeguarded applications, and is designed for embedding into chip cards according to ISO 7816. Each security measure is designed to act as an integral part of the complete system in order to strengthen the design as a whole. The security measures can be divided into hardware controlled security measures that do not allow for software guided exceptions and security measures that shall be controlled by software. In addition the die is embedded into a module which in turn is embedded into a smart card. Note that Philips delivers wafers. Module production and module embedding (card production) takes place at the Card Manufacturer.
The module and card embedding of the TOE provide external security mechanisms because they make it harder for an attacker to access parts of the TOE for physical manipulation.

The following table lists the TOE components.

Type	Name	Release	Date	Form of delivery
Hardware	Philips P8WE6004V0D Secure 8-bit Smart Card Controller	V0D	07.11.01 (GDS2 File)	Wafer (dice include reference C017D)
Software	Test ROM Software *(the IC dedicated software)*	zz036	04.04.01	Test-ROM on the chip
Document	Guidance, Delivery and Operation Manual	V 1.2	29.01.02	Printed document

Type	Name	Release	Date	Form of delivery
Document	Data Sheet	1.0 Doc. No.: 054610	06.03.01	Printed document

Table 1: Components of the TOE

The CPU of the Philips P8WE6004V0D smart card controller is a derivation of the 80C51 family and has the same instruction set. The following figure 1 shows the components of the microcontroller.

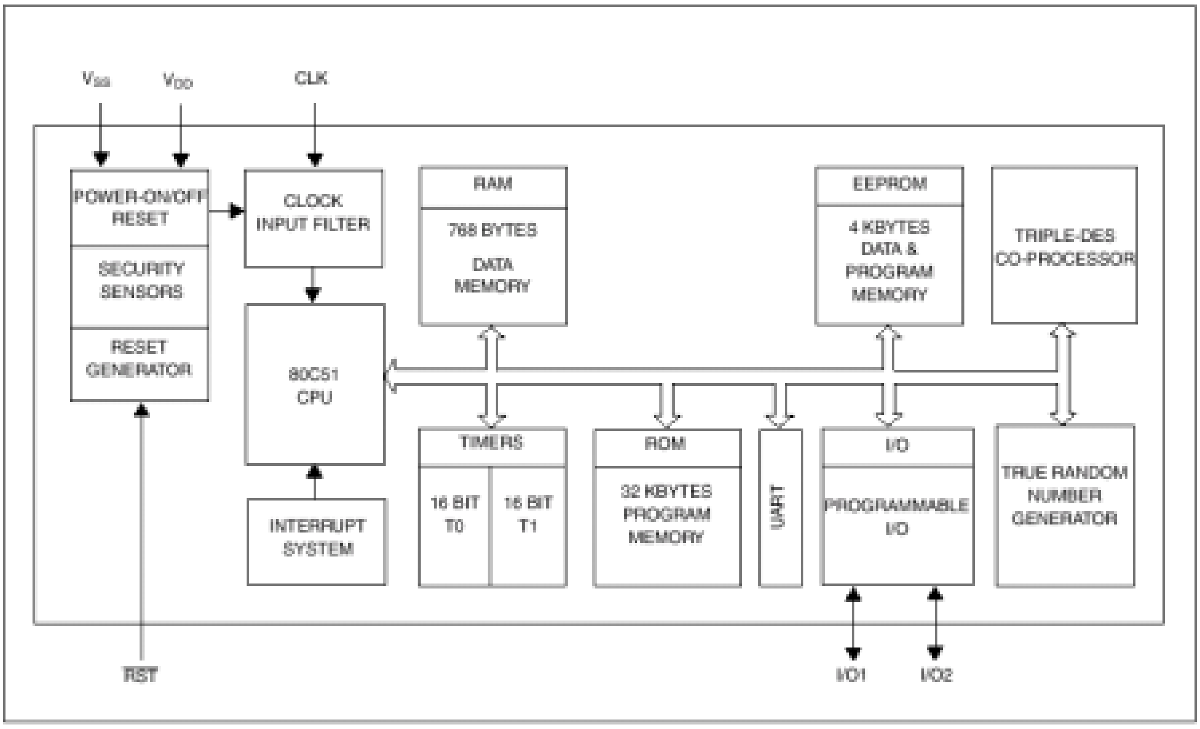

Fig 1: Overview diagram Philips P8WE6004V0D smart card controller

The device includes ROM, RAM and EEPROM memory. The EEPROM can be accessed as data memory as well as program memory. The Triple-DES co-processor supplies single DES and Triple-DES operations. Only Triple-DES according to ISO standard 8732 (1988) is used in the evaluation and therefore part of the evaluated configuration. The random generator provides true random numbers without pseudo random calculation.

The on-chip hardware is software controlled via Special Function Registers. These registers are correlated to the activities of the CPU, Interrupt, I/O, EEPROM, Timers, and UART. The communication with the Philips P8WE6004V0D smart card controller can be performed through a serial interface I/Os according to ISO standard 7816-3. Two 16-bit timers and five vectorized interrupts provide further functionality for I/O, timers and EEPROM.

The Philips P8WE6004V0D smart card controller operates with a single 3V or 5V nominal power supply at a nominal maximum external clock frequency of 8 MHz. The controller provides an internal clock to perform security algorithms. The instruction set contains 255 different instructions, therefore the instruction has a length of one byte which can be followed by parameters of one or two additional bytes. The controller provides three power saving modes with reduced activity: the IDLE Mode and the SLEEP Mode, which includes the CLOCK STOP Mode.

The Philips P8WE6004V0D smart card controller chip will be implemented in a credit card sized plastic card (micro-module embedded into the plastic card) or another sealed package. The chip provides a hardware computing platform to run smart card applications executed by a smart card operating system. Smart card applications will be used to store secret data and calculate cryptographic functions. The secret data can be used as input for the calculation of authentication data and the encryption of data and keys.

The smart card operating system stored in the Customer-ROM and the application stored in the Customer-ROM and/or in the EEPROM are not a part of the TOE. The code in the Test-ROM of the TOE is used by the manufacturer of the smart card to check the chip function. This test code is disabled before the operational use of the smart card.

The smart card applications use the security functions of the operating system based on the security features of the TOE. Mostly independent of the smart card operation system, the TOE provides a symmetric block cipher algorithm and a random number generator to perform cryptographic operations in a secure way. The block cipher algorithm may be used as a cryptographic primitive for encryption of user data and for authentication of user data and entities. The random number generator may be used by the software of the TOE environment for the generation of cryptographic parameters on the smart card and especially of keys. These strong keys should be used when the cryptographic primitives of the TOE are invoked by the software of the environment.

The TOE protects the secret data stored in and operated with the TOE against physical tampering. Within the composition of this TOE, the operating system, and the smart card application the security functionality is only partly provided by the TOE and causes dependencies between the TOE security functions and the functions provided by the operating system or the smart card application on top.

2.1.1 Hardware Description

The microcontroller chip includes:

- CPU 80C51 with
  - Interrupt module,
  - two 16-bit Timers,
  - I/O-Interface,
  - UART,
- memory consisting of
  - 256 + 512 Bytes RAM,
  - 32 kBytes User ROM,
  - 4 Kbytes EEPROM,
- Triple-DES Co-processor,
- Random generator,
- Power module with security sensors and security logic.

Philips will deliver the TOE at the end of phase 3 after the production test in form of wafers. The IC Packaging (phase 4) and the following phases are under control of the customer of Philips. The chip shall be dealt in an appropriate packaging when being used in the field at phase 7.

2.1.2 Software Description

The TOE includes IC dedicated software (called firmware in the following) developed by Philips and embedded in the Test-ROM. The smart card embedded software (called application software in the following; not being part of the TOE) is stored in the Customer-ROM area (User-ROM). The firmware includes the Test Operating System, test routines for the various blocks of the circuitry, control flags for the status of the security area and shutdown functions to ensure that security relevant test operations cannot be executed illegally. The application software depends on the usage of the smartcard and does normally include an Operating System and an application.

2.1.3 Documentation

The Data Sheet of the Philips P8WE6004V0D smart card controller is also part of the TOE. It contains a functional description needed to develop software, guidelines for the use of security features and the instruction set of the TOE. The guidance documentation describes aspects of the program interface and the use of programming measures to improve the security. The provided documentation can be used by the application software developer to develop the Smartcard Embedded Software.

2.1.4 Interface of the TOE

In the user mode the electrical interface of the TOE are the pads to connect the lines power supply, reset input, clock input, ground, Input1/Output1.

In the user mode the logical interface of the TOE depends on the Smartcard Embedded Software. This logical interface can change based on the actual phase beginning with phase 4 since the TOE is switched to the user mode at the end of phase 3.

The internal interface of the TOE for the Smartcard Embedded Software and the developer of the Smartcard Embedded Software is the set of instructions, the bits in the special function registers that are related to the user mode and described in the data sheet as well as the address map for the access of memories by the CPU.

Note: In the test mode (use before delivery of the TOE at the end of phase 3) the interface is the set of test functions based on the test operating system in the Test-ROM and provided at the electrical interface.

In case of an attack where the attacker performs manipulations on the chip surface, the surface can be seen as an interface of the TOE, too.

2.2 Smart Card Product Life-Cycle

The life-cycle of the Philips P8WE6004V0D smart card controller corresponds to the life-cycle definition of the Protection Profile. The developer of the TOE is responsible for phases two and three, development and production of the TOE. Therefore these phases will be in the scope of those assurance requirements dealing with developer security and configuration management (classes ACM and ALC). All further phases are user phases.

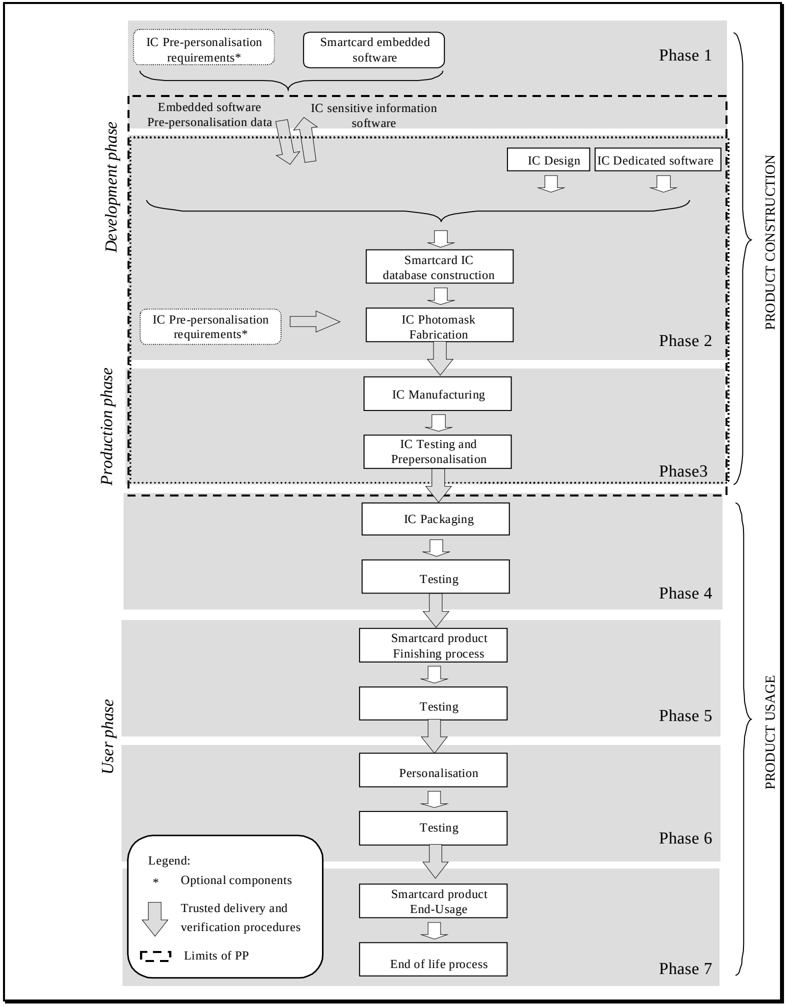

Fig 2: Smartcard Product Life Cycle

Note: Regarding the work flow and the process organisation of Philips the IC Photomask Production is shifted to phase 3 in this Security Target. The complete handling of photomasks (ordering, version control, usage, storage and destruction) is done by the wafer fab.

2.3 TOE Environment

2.3.1 TOE Development Environment

The development of the TOE includes the development of the hardware and of the software in the Test-ROM. The following steps are relevant for the development:

- Concept and specification of the microcontroller,
- HW-Design including the design of the circuit and the layout,
- Development of the firmware,
- Handling of customer specific data (ROM-code and EEPROM-data) as well as
- Preparation of documentation for the customer (software developer).
- During the design and the layout process only people involved in the specific development project for a IC have access to sensitive data. The trustworthiness of used components is ensured with simulation and verification tools that use test pattern generated by Philips Semiconductors, Business Unit Identification. Different people are responsible for the design data and for customer related data. The security measures installed within Philips Semiconductors, Business Unit Identification ensure a secure computer system and provide appropriate storage equipment for the different development tasks.
- The verified layout data is provided from the development to the wafer fab.

2.3.2 TOE Production Environment

The wafer fab provides the layout data of the different photomasks to the manufacturer of the photomasks. The photomasks are generated off-site. The photomasks are verified against the design data of the development before the usage. The accountability and the traceability is ensured among the wafer fab and the photomask provider.
The production of the wafers can be split into two different steps regarding the requirements of the different user. The wafer can be produced with the customer independent mask set and stored in a wafer stock until the wafer is completed with the user specific masks. The computer tracking ensures the control of the complete process including the storage of the semi-finished wafers.
The test process of the wafers is performed within another site of Philips. The delivery process between the involved Philips sites provide accountability and traceability of the produced wafers. Non-functional ICs are marked on the wafer but will be delivered on the wafer to the packaging process.

2.3.3 TOE User Environment

The TOE user environment is the environment of phases 4 to 7. At phases 4, 5 and 6, the TOE user environment must be a controlled environment.
In the End-user environment (phase 7) smartcards are used in a wide range of applications to assure authorised conditional access. Examples of such are Pay-TV, Banking Cards, Portable communication SIM cards, Health cards, Transportation cards.
The end-user environment therefore covers a wide spectrum of very different functions, thus making it difficult to avoid and monitor any abuse of the TOE.
Phases 4 to 7 of the smart card life cycle are not part of the TOE construction process in the sense of this Security Target. Information about those phases are just included to describe how the TOE is used after its construction. Nevertheless the security features of the smartcard IC hardware that are independent of the software are active from the end of phase 3 and cannot be disabled thought the application software in the phases 4 to 7.

2.4 TOE Logical Phases

The TOE is able to control two different logical phases. After production the chip is in the test mode that means under the control of the test software. At the end of the production test the chip will be switched into the user mode so that the chip is under the control of the application software.

2.5 TOE Intended usage

Regarding to phase 7, the combination of the smartcard hardware and the application software is used by the end-user. The method of use of the product in this phase depends on the application. During the other phases of the product construction and the product usage there are several administrator- and user-functions.

Phase 1: The Smartcard Embedded Software developer develops software for the smartcard, including a smartcard operating system and/or application specific software parts. By using the software interface of the TOE (in user mode) as defined in section 2.1.4 he is the user of the smartcard hardware with the hardware features.

Phase 2: The IC designer is responsible for the design of the chip that is developed within this phase. In parallel the IC designer develops the IC dedicated software for the production test of the chip that is included in the Test-ROM. Therefore the IC designer takes the role of the administrator during this phase.

Phase 3: The function of the administrator is split into two: The IC manufacturer is responsible for the IC production itself. Regarding the production test after the manufacturing process the test engineer is the administrator.

Note: The definition of the user roles regarding the TOE for the phases 4 to 7 is provided here as additional information and is not in the scope of the evaluation. However the operation manuals address some of the user roles defined for phase 1 and the following phases.

Phase 4: the IC packaging manufacturer (administrator),
· the Smartcard Embedded Software developer (user),
· the system integrators such as the terminal software developer (user).

Phase 5: the smartcard product manufacturer (administrator),
· the Smartcard Embedded Software developer (user),
· the system integrators such as the terminal software developer (user).

Phase 6: the personaliser (administrator),
· the smartcard issuer (administrator),
· the Smartcard Embedded Software developer (user),
· the system integrators such as the terminal software developer (user).

Phase 7: the smartcard issuer (administrator),
· the smartcard end-user (user),
· the Smartcard Embedded Software developer (user),
· the system integrators such as the terminal software developer (user).

The Smartcard Embedded Software developer and the system integrators such as the terminal software developer are listed in Phases 4-7 because they may use samples of the TOE in these phases for their testing purposes. It is not intended that they are able to change the behaviour of the smartcard in another way than a user.
The IC manufacturer and the smartcard product manufacturer may receive ICs from different phases for analysis purpose, if problems should occur during the smartcard usage.

2.6 General IT features of the TOE

The TOE IT functionality consist of

- tamper resistant data storage
- basic cryptographic functions (DES co-processor)
- physical random number generator
- data communication

3 Security Environment

The chapter Security Environment is divided into the following sections:

Definition of Subjects, Objects and Access Modes

Assets

Assumptions

Threats

Organisational Security Policies

3.1 Definition of Subjects, Objects and Access Modes

In this section definitions are given for:

- the subjects, which are able to use the TOE,
- the objects that are protected by the TOE,
- the Access Modes and Methods, which describe, how the subjects can act on the objects based on the technical opportunities.

Note, that in order to make the analysis of potential attack scenarios possible, even access methods are defined and discussed, which are effectively prevented by the TOE.

These definitions describe the TSF Scope of Control (TSC) and are used in later chapters (e. g. to define operations for some of the Security Functional Requirements).

3.1.1 Subjects

Subjects relevant for considering the security of the TOE are:

(S1) the administrator

The test engineer of the IC manufacturer using the Test Mode ROM to test the correct function of the IC according to the manufacturing specification. Regarding to the assumptions of the environment during the manufacturing process no user is considered as attacker. Nevertheless the tamper resistance measures are already active. This administrator is defined only for phase 3, after this phase the administrator is not able to act as administrator.

(S2) the end-user

This definition includes all subjects that can get access to the TOE within phase 4 to 7 when the TOE is switched in the user mode. This might be the legal owner of the smart card or another person who succeeds in getting access to the smart card with the final TOE. It also includes the users that are defined as administrator for the different personalisation steps of phases 4 to 7. These persons may perform direct attacks on the smartcard containing the TOE to access the objects or processed data circumventing the access provided by the application software.

(S3) the test software

This means the IC dedicated software, when it is executed by the TOEs CPU and therefore able to act as a subject. This is only possible within phase 3. The test software is developed by the IC manufacturer and implemented in the TOE's Test-ROM as object O1.

(S4) the application software

This means the software built by the application developer and implemented in the TOE's User-ROM and E^{2}PROM as object O2. The application software can be used at first when the chip is switched to the user mode.

3.1.2 Objects

This section defines the objects and access methods relevant for considering the usage of the TOE based on the technical opportunities of the subjects.

(O1) Test software while being stored in the test area read-only memory (Test-ROM). This object may be (i) executed (i. e. A5 starting S3) or (ii) might be manipulated A4.

(O2) Application software while being stored in the user area of the read-only memory (User-ROM) and the electrically erasable PROM (E^{2}PROM). This object may be (i) executed (i. e. A5 starting the subject S4), (ii) written A7 by the test software S3 or the application software S4 itself, or (iii) might be manipulated A4.
Note that the test software S3 and the application software S4 can modify (write A7) only the E^{2}PROM part of the application software.

(O3) Application data while being stored in the electrically erasable PROM (E^{2}PROM) or the random-access memory (RAM) of the TOE. This object may be (i) read A6 by the test software S3[1] and by the application software S4, (ii) written A7 by the test software S3 and by the application software S4, or (iii) might be manipulated A4. Note that the application data are described by O3 during storage and by O6 during processing.

(O4) Random numbers generated by the random number generator (RNG) of the TOE. This object may (i) be read A6 by the test software S3 and the application software S4, and (ii) might be measured A1, affected A2, influenced A3 or manipulated A4.

(O5) Configuration data while being stored in the one time programmable memory (OTP) of the TOE. This object may be (i) read A6 by the test software S3 and the application software S4, (ii) set A8 by the test software S3 and the application software S4 and (iii) might be manipulated A4.
Note that due to the TOE design the OTP may be modified by means of the software running on the TOE only in one direction. This is described by the access method set A8.

(O6) Information of the data O3 and O4 while being processed by the CPU and the DES co-processor of the TOE. This object may be (i) processed A9 by the test software S3 or by the application software S4, and (ii) measured A1, affected A2 or manipulated A4.

(O7) Mode-switch as special bytes in the security area defining the TOE mode as test mode or user mode. The mode-switch (i) will be set A8 by the test software S3 to change the test mode into the user mode and (ii) might be manipulated A4.

(O8) Sensor-switch as special bytes in the security area enabling or disabling the sensors protecting the TOE against affecting A2. The test-program S3 will (i) enable A10 the sensor-switch activating the sensors or (ii) disable A11 the sensor-switch deactivating the sensors. In addition the sensor-switch might be manipulated A4.

[1] In fact there exist no test functions, which allow users or test personnel to read data from the memory of the TOE.

3.1.3 Access Modes and Methods

(A1) Measure O4 or O6: Measuring the dynamic signals generated by the TOE and gaining information about the object O4 while being generated by the random number generator and the object O6 while being processed by the CPU or the DES co-processor.
Note that signals to be measured may be generated only by the random number generator, by the CPU or by the DES co-processor processing data.

(A2) Affect O4 or O6: Affect means manipulation of the operational conditions of the TOE out of the specified range. Affect O4 or O6 means to run the software on the CPU or to use the random number generator or the DES co-processor while the voltage, clock or temperature as TOE's operational conditions are out of the specified range. Malfunctions of the CPU, random number generator, or the DES co-processor may violate the integrity of the stored objects O2 (E^2PROM part), O3, O5, O7 or O8.

(A3) Influence O4: Influence means to change the environmental conditions of the TOE as voltage, clock or temperature within the specified range to violate the integrity of the random number generator (O4).
Note that the random number generator uses analogue signal processing that may be influenced. It is guaranteed that digital signal processing of the CPU and the DES co-processor are stable within the specified range of the voltage, clock or temperature as TOE's operational conditions as voltage, clock or temperature.

(A4) Manipulate O1, O2, O3, O4, O5, O6, O7 or O8: Specific hardware manipulations, such as connecting to the bus lines, measuring or changing the content of a memory cell and modification of the internal circuitry. These manipulations include to modify the contents of stored data in the ROM (O1, O2), E^2PROM, RAM (O3), OTP O5, O7 or O8 by means of changing the environmental conditions of the TOE out of the specified range of the TOE's operational conditions, when no software is running on the TOE.

(A5) Execute O1 or O2: Execute the object O1 (as test software S3) or the object O2 (as the application software S4) means in the context of the security policy (i) to write O2, (ii) to read or to write O3, (iii) to read O4, (iv) to read or to set O5, or (v) to process O6

(A6) Read O3, O4, or O5: Read means to read the data from the storage media (E^2PROM, RAM O3, or OTP O5,) or the SFR (O4), and to be able output the information as the test software S3 or the application software S4 running on the TOE.

(A7) Write O2 or O3: Write means to modify the data in the storage media (E^2PROM, RAM) by the software running on the TOE (i. e. S3 or S4). The E^2PROM contains parts of O2 and O3. The RAM contains parts of O3.

(A8) Set O5 or O7: Set the object O5 means to write logical 1 into a bit of the OTP by means of the running software (i. e. the test software S3 or the application software S4). Set the object O7 means to change the mode- switch that the TOE detects the user mode after setting.
Note that reset (i. e. write logical 0) a bit of the OTP is not possible due to the TOE design.

(A9) Process O6: The information O6 is processed in the CPU and the DES co-processor by the test software S3 or by the application software S4.

(A10) Enabling O8: Enabling the sensor-switch activates the sensors protecting the TOE against affecting A2.

(A11) Disabling O8: Disabling the sensor-switch deactivates the sensors protecting the TOE against affecting A2.

The Table 2 provides an overview of the access methods based on the technical opportunities of the subjects. The access methods A1, A2, A3, A4 and A5 are performed form outside the TOE and ascribed to the subjects S1 and S2. The access methods A6, A7, A8, A9, A10 and A11 are performed by software running on the TOE and ascribed to the subjects S3 and S4.

	O1	O2	O3	O4	O5	O6	O7	O8
S1	A4, A5	A4, A5	A4	A1, A2, A3, A4	A4	A1, A2, A4	A4	A4
S2	A4, A5	A4, A5	A4	A1, A2, A3, A4	A4	A1, A2, A4	A4	A4
S3	-	A7*[1]	A6, A7	A6	A6, A8	A9	A8	A10, A11
S4	-	A7*[2]	A6, A7	A6	A6, A8	A9	A8*[3]	A10, A11

Table 2: All access methods, which have to be considered

*[1] The test software is able to store data in the EEPROM after the tests are finished. These data can be a part of the application software. The customer is responsible for the data stored in the EEPROM at the end of the test process of phase 3.

*[2] The application software stored in the ROM will be able to store additional data in the EEPROM that can be executed. The software developer is responsible for this security relevant item. This is therefore not in the scope of this evaluation.

*[3] The application software is able to set bits in one part of OTP area that is not write protected but can not be erased based on the hardware design.

3.2 Assets

No additional assets to those already defined in the protection profile were identified.

The objects defined in the preceding section are specific instances of some of these assets.

3.3 Assumptions

Since this Security Target claims conformance to the protection profile PP/9806, the assumptions defined in section 3.2 of the protection profile are valid for this Security Target.

This section defines specific additional assumptions for this ST.

Assumption on phase 1:

A.SOFT_FUNC The embedded software running on the TOE in phases 4 to 7 checks the integrity of the user data stored on the TOE and responses appropriately in case of loss of integrity due to errors. The software shall react adequately to the output signals of the protection mechanisms provided by the TOE.

Note, that this is an assumption for Phase 1, because the application software developer is assumed to implement these properties correctly during phase 1. It can be seen as a special case of

A.SOFT_ARCHI (see PP/9806, section 3.2.1), however it is stated on its own for additional clarity.

Assumption on phase 4 to 6:
A.SCRAP_MARKED All ICs marked as non functional by Philips on the wafers delivered to the IC packaging manufacturer will be destroyed ("scraped") under controlled conditions.

This assumption can be seen as a special case of the assumptions on the delivery process (see PP/9806, section 3.2.2), however it is stated on its own for additional clarity.

3.4 Threats

Since this Security Target claims conformance to the protection profile PP/9806, the threats defined in section 3.3 of the protection profile are valid for this Security Target.

No additional threats where identified, which are not already defined in the protection profile.

3.5 Organisational Security Policies

As stated in the protection profile, specifications of organisational security policies depend essentially on the applications in which the TOE is incorporated. Therefore, no organisational security policy is defined in this Security Target. All security objectives are derived solely from the threats.

4 Security Objectives

The chapter *Security Objectives* is divided into the following sections:

TOE Security Objectives

Security Objectives for the Environment

4.1 TOE Security Objectives

The security objectives defined in section 4.1 of the protection profile are valid for the TOE. In the following the security objectives defined in the PP/9806 are reorganised into specific security objectives

Note: The TOE security objectives have to be supported by the environment. For example O.CLON can be supported by the TOE itself only in so far that the information needed for cloning cannot be derived from the TOE itself. That this information cannot be stolen from the development and production sites cannot be guaranteed by technical mechanisms of the TOE itself. Similarly O.DIS_MEMORY and O.MOD_MEMORY have to be supported by the application software since the TOE cannot prevent a badly designed application software from corrupting or disclosing its own sensitive data.
O.CLON can be seen as a combination of O.DIS_MEMORY, O.TAMPER, O.DIS_MECHANISM.

This note is not specific for this Security Target but obviously holds already for the general situation described in the protection profile.

The following specific security objectives (SO1) – (SO5) reorganise the objectives listed in the PP/9806 and include additional objectives regarding control- and crypto-functions. The Security Policy used by some Security Functional Requirements (see chapter 5.1.1.1) will be based on these security objectives.

(SO1) Mode Control (Introduces Mode Control and covers special aspects of O.OPERATE by providing test functionality in Test Mode and special aspects of O.DIS_MEMORY and O.MOD_MEMORY by preventing the use of test functions in User Mode)

The TOE shall provide test functionality and allow the administrator S1 to execute the test software in the test mode.
The TOE shall allow the end-user S2 to execute the application software in the user mode.
The TOE shall prevent that an end-user S2 executes the test software in the user mode and doing so:

(a) to read out the application data independently of the read access provided by the application software.

(b) to write the application data with chosen values independently of the write access provided by the application software,

(c) to set the configuration data independently of the application software,
(d) to read random numbers independently of the application software,
(e) to process information of the application data and the random numbers independently of the application software.

(SO2) Operating Conditions Control (Covers O.OPERATE)
The TOE allows the execution of the application software in specified range of the operational condition of the TOE. The TOE shall prevent malfunctions during execution of the application software in the user mode forced by the operating parameters voltage supplied, frequency of the clock or temperature of the chip, which are out of the specified range of the operational condition. If the TOE is configured to control the operating parameters and one of these parameters are out of the specified range then the TOE shall reset the actual running program.

(SO3) Tamper Proof (Covers O.FLAW, O.TAMPER and thereby also O.DIS_MECHANISM, O.DIS_MEMORY, O.MOD_MEMORY)
The construction of the TOE shall resist manipulation to withstand attacks trying to manipulate the hardware and disable security enforcing functions of the TOE in the user mode.
The construction of the TOE shall prevent that an attacker is able to manipulate (i) the test software in the ROM, (ii) the application software in the ROM and the E^2PROM, (iii) the application data in the E^2PROM and RAM, (iv) the configuration data in the OTP and (v) the mode-switch when the chip is in the user mode.
The construction of the TOE shall avert the analysis of the design in the user mode to prevent cloning of the TOE hardware.

(SO4) Non-existence of Information Leakage (Covers aspects of O.FLAW and adds support against disclosure of user data during the correct processing)
The TOE must not contain flaws in the design, implementation or operation that cause Information Leakage. The TOE's hardware construction and test-software design shall ensure that it does not disclose data processed by means of the CPU, the DES co-processor or the RNG. This means that during the user mode no static or dynamic signals are generated that can be measured and analysed by an attacker using the physical external interface or that an attacker is able to disclose data based on a flaw using the physical external interface or the surface of the smartcard IC.

(SO5) Cryptographic support (Additional Objective, introduces support by the Random Number Generator and the DEA co-processor)
(a) Bytes read from the TOE's random number generator in the user mode shall be unpredictable.
(b) The TOE shall provide a tamper resistant Triple-DES implementation.

4.2 Security Objectives for the Environment

Since this Security Target claims conformance to the protection profile PP/9806, the objectives defined in section 4.2 of the protection profile are valid for this Security Target.

Note that according to the note in the preceding subsection the application software, which is a part of the environment, has to support all security objectives of the TOE.

The following additional objective for the environment is identified (which is relevant for the phases 2 - 6:

OE.SCRAP_DEFECT Appropriate methods are established to destroy all defect ICs and pre-finished smart cards. The non functional IC, which marked on the delivered wafers, and the defect smart cards identified according to O.TEST_OPERATE will be scraped under controlled conditions by the responsible manufacturer of that phase.

Note that this objective can be seen as a special aspect of O.TEST_OPERATE. It is also described as part of the text of PP/9806, section 2.3.2. However, it was added as an explicit objective for clarity.

5 IT Security Requirements

The chapter *IT Security Requirements* is divided into the following sections and subsections*:*

TOE Security Requirements

TOE Security Functional Requirements

TOE Security Assurance Requirements

Security Requirements for the Environment

5.1 TOE Security Requirements

5.1.1 TOE Security Functional Requirements

5.1.1.1 Definition of Security Policies used in Security Functional Requirements

Some of the SFRs defined for the TOE require access policies to be defined. This is done in this section.

The security policy informally defined by the security objectives (SO1) to (SO5) in section 4.1 can be described semiformally as follows:

SO1 Mode Control

The TOE shall allow the access-sets (S1, A5, O1) and (S3, A7, O2) in the test mode as well as (S2, A5, O2) in the user mode.

The TOE shall prevent access-sets (S1, A5, O2) in the test mode as well as (S2, A5, O1) and (S4, A8, O7).

The external security measure A.SOFT_ARCHI prevents (S4, A7, O2) in the user mode.

SO2 Operating Conditions Control

If the sensor-switch O8 is enabled A10 then for the test mode as well as for the user mode

(a) the TOE shall prevent the access-sets (S1, A2, O4), (S1, A2, O6), (S2, A2, O4) and (S2, A2, O6) and

(b) if A2 is detected the TOE shall prevent the access sets (S2, A5, O1), (S2, A5,O2), (S3, A6, O3), (S3, A6, O4), (S3, A6, O5), (S3, A7, O2), (S3, A7, O3), (S3, A8, O5), (S3, A9, O6), (S4, A6, O3), (S4, A6, O4), (S4, A6, O5), (S4, A7, O2), (S4, A7, O3), (S4, A8, O5) and (S4, A9, O6).

SO3 Tamper Proof

The TOE shall prevent access-sets (S1, A4, O1), (S1, A4, O2), (S1, A4, O3), (S1, A4, O4), (S1, A4, O5), (S1, A4, O6), (S1, A4, O7), (S1, A4, O8), (S2, A4, O1), (S2, A4, O2), (S2, A4, O3), (S2, A4, O4), (S2, A4, O5), (S2, A4, O6), (S2, A4, O7) and (S2, A4, O8) in the user mode as well as in the test mode.

SO4 Non-existence of Information Leakage

The TOE shall prevent the access-sets (S1, A1, O4) and (S1, A1, O6) in the test mode as well as (S2, A1, O4) and (S2, A1, O6) in the user mode.

SO5 Cryptographic support

The TOE shall allow the access-sets (S3, A6, O4), (S4, A6, O4), (S3, A9, O6) and (S4, A9, O6) where S3 has permission only in the test mode and S4 has permission only in the user mode.

The TOE shall prevent the access-sets (S1, A1, O4), (S1, A2, O4), (S1, A3, O4), (S1, A4, O4), (S2, A1, O4), (S2, A2, O4), (S2, A3, O4) and (S2, A4, O4) as well as the access-sets (S1, A1, O6) (S1, A2, O6), (S1, A4, O6), (S2, A1, O6), (S2, A2, O6) and (S2, A4, O6) in the test mode and in the user mode.

The security objectives imply the following security policies:

SP1 Tamper Protection Security Policy

SP1.1 The TOE shall prevent any access-method measuring A1, influencing A3 and manipulating A4 for all objects which may be accessed by these methods.

The SP1 is described by the following list of access-sets prevented by the TOE: (S1, A1, O4), (S1, A1, O6), (S1, A3, O4), (S1, A4, O1), (S1, A4, O2), (S1, A4, O3), (S1, A4, O4), (S1, A4, O5), (S1, A4, O6), (S1, A4, O7), (S1, A4, O8), (S2, A1, O4), (S2, A1, O6), (S2, A3, O4), (S2, A4, O1), (S2, A4, O2), (S2, A4, O3), (S2, A4, O4), (S2, A4, O5), (S2, A4, O6), (S2, A4, O7) and (S2, A4, O8).

SP1.2 The TOE shall prevent successful affecting A2 (i) in the Test mode (S1, A2, O4), (S1, A2, O6) if the sensor-switch is enabled, and (ii) in the User mode (S2, A2, O4), (S2, A2, O6).[2]

If attempted affecting A2 is detected the TOE shall prevent the access sets (S2, A5, O1) and (S2, A5,O2). Moreover, it shall close the running program (this prevents all other access-sets listed in SO2 (b) above).

SP2 Test Mode Security Policy

In the test mode the administrator S1 is permitted to execute A5 the test software O1. The test software as subject S3 is permitted to access the objects O2, O3, O4, O5, O6, O7 and O8 as described by Table 3. The TOE shall prevent the access-sets described by Table 4.

SP3 User Mode Security Policy

In the user mode the end-user S2 is permitted to execute A5 the application software O2. The application software as subject S4 is permitted to access the objects O3, O4, O5, O6, O7 and O8 as described by Table 5. The TOE shall prevent the access-sets described by Table 6.

	O1	O2	O3	O4	O5	O6	O7	O8
S1	A5		X*	X*	X*	X*	X*	X*
S2			X*	X*	X*	X*	X*	X*

[2] Note that due to the assumption A.DEV_ORG the integrity of the TOE will be ensured by the test environment if the TOE is used out of the specified range of the operational condition.

S3		A7	A6, A7	A6	A6, A8	A9	A8	A10, A11
S4								

Table 3: Allowed access according to the Test Mode Security Policy

X* This combination is not applicable regarding the regular access modes A5 to A11 that can be controlled by the chip.

	O1	O2	O3	O4	O5	O6	O7	O8
S1		A5	X*	X*	X*	X*	X*	X*
S2	A5	A5	X*	X*	X*	X*	X*	X*
S3								
S4		A7	A6, A7	A6	A6, A8	A9	A8	A10, A11

Table 4: Prevented access according to the Test Mode Security Policy

X* This combination is not applicable regarding the regular access modes A5 to A11 that can be controlled by the chip.

	O1	O2	O3	O4	O5	O6	O7	O8
S1			X*	X*	X*	X*	X*	X*
S2		A5	X*	X*	X*	X*	X*	X*
S3								
S4		A7*[1]	A6, A7	A6	A6, A8	A9		

Table 5: Allowed access according to the User Mode Security Policy

X* This combination is not applicable regarding the regular access modes A5 to A11 that can be controlled by the chip.

*[1] Note that this is a security relevant option based on the decision of the software developer.

	O1	O2	O3	O4	O5	O6	O7	O8
S1	A5	A5	X*	X*	X*	X*	X*	X*
S2	A5		X*	X*	X*	X*	X*	X*
S3		A7	A6, A7	A6	A6, A8	A9	A8	A10, A11
S4							A8	A10, A11

Table 6: Prevented access according to the User Mode Security Policy

X* This combination is not applicable regarding the regular access modes A5 to A11 that can be controlled by the chip.

5.1.1.2 TOE Security Functional Requirements already contained in the PP/9806 and applicable to Phase 3 only

The SFRs already specified in the protection profile PP/9806, which are applicable to phase 3 only, are listed in this subsection. The assignment and selection operations, which were left open in the PP, are completed here. Additionally for every SFR a refinement is stated, which serves as a short explanation of the relevance for the specific TOE.

FIA_UAU.2 User authentication before any action
FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies: FIA_UID.1 Timing of identification

(No operations necessary.)

Refinement: For the TOE this means that the test engineer S1 will have be authenticated before execution of test functions by the test ROM. The end-user S2 will be authenticated implicitly by the TOE as anybody, who is not authenticated as a test engineer. S2 cannot execute security relevant functions of the TOE (S2 stands for anybody including potential attackers).
Note, that the decision, which security relevant functions provided by the user ROM (application software) an end-user is allowed to use, is outside of the control of the TOE. It has to be controlled by suitable functions of the user ROM itself.

FIA_UID.2 User identification before any action
Hierarchical to: FIA_UID.1
FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.
Dependencies: No dependencies.

(No operations necessary.)

Refinement: This identification is done implicitly with help of the flag “mode-switch” which is changed during the TOE life-cycle at the end of phase 3 from test mode into the user mode: The controlled environment within phase 3 limits TOE access to the administrator. Moreover, the administrator S1 is implicitly identified by the TOE during authentication, before test functions can be used. The user mode is active during phase 4 to 7. In this mode everybody is identified implicitly as end-user (which is the same as “anybody”) S2.
From user mode the TOE cannot be switched back into test mode.

FIA_ATD.1 User attribute definition
Hierarchical to: No other components.
FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to

individual users: *administrator or end-user*[3].
Dependencies: No dependencies.

Refinement: The TOE will distinguish only between administrators S1 and end-users S2. No further user attributes besides these role-names are necessary. All properties of these roles are defined statically and implicitly.

FPT_TST.1 TSF testing
Hierarchical to: No other components.
FPT_TST.1.1 The TSF shall run a suite of self tests *at the request of the authorised user S1, at the conditions test mode*[4] to demonstrate the correct operation of the TSF.
FPT_TST.1.2 The TSF shall provide authorised users with the capability to verify the integrity of TSF data.
FPT_TST.1.3 The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.
Dependencies: FPT_AMT.1 Abstract machine testing

Refinement: At the end of phase 3 the test engineer S1 can initiate a suite of self tests demonstrating correct operation of the TSF and checking integrity of the IC dedicated software (the test ROM content) and data. This is possible only in test mode.

FDP_SDI.1 Stored data integrity monitoring
Hierarchical to: No other components.
FDP_SDI.1.1 The TSF shall monitor user data stored within the TSC for *integrity errors*[5] on all objects, based on the following attributes: *all user data*[6].
Dependencies: No dependencies.

Refinement: The production testing at the end of phase 3 includes functions, which allow to check the integrity of the user ROM. Moreover the test functions used to write user data into the EEPROM of the TOE check the correctness of these data after writing. All user data are checked, therefore no specific attributes have to be defined.

5.1.1.3 TOE Security Functional Requirements already contained in the PP/9806 and applicable to Phases 3 to 7

The SFRs already specified in the protection profile PP/9806, which are applicable to phase 3 to 7, are listed in this subsection. The assignment and selection operations, which were left open in

[3] [assignment: list of security attributes]

[4] [selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self test should occur]]

[5] [assignment: integrity errors]

[6] [assignment: user data attributes]

the PP, are completed here. Additionally for every SFR a refinement is stated, which serves as a short explanation of the relevance for the specific TOE.

FMT_MOF.1 Management of security functions behaviour
Hierarchical to: No other components.
FMT_MOF.1.1 The TSF shall restrict the ability to *disable*[7] the functions *sensors*[8] to the *administrator*[9].
Dependencies: FMT_SMR.1 Security roles

Refinement: The administrator S1 can only disable the sensors in test mode. They are enabled automatically when the TOE is switched to user mode. In user mode the sensors are permanently active and cannot be disabled by S1 or S2.

FMT_MSA.1 Management of security attributes
Hierarchical to: No other components.
FMT_MSA.1.1 The TSF shall enforce the *Mode-Control-Policy*[10] to restrict the ability to *modify*[11] the security attributes *mode-switch (*O7*)*[12] to *administrator S1*[13].
Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control][14]
FMT_SMR.1 Security roles

Refinement: The Mode-Control-Policy is an access control policy defined as the combination of the Test Mode Security Policy and the User Mode Security Policy as defined in section 5.1.1.1. Note that the policy in fact doesn't allow the administrator to modify the mode-switch directly (access type A4). The administrator can only start the test software (access type A5 to O1) and can then call a test function, by which the test software (as S3) modifies (access A8) the mode-switch (O7)

FMT_SMR.1 Security roles
Hierarchical to: No other components.
FMT_SMR.1.1 The TSF shall maintain the roles *the administrator S1* and *the end-user S2*[15].
FMT_SMR.1.2 The TSF shall be able to associate users with roles.

[7] [selection: determine the behaviour of, disable, enable, modify the behaviour of]

[8] [assignment: list of functions]

[9] [assignment: the authorised identified roles]

[10] [assignment: *access control SFP, information flow control SF*P]

[11] [selection: *change_default, query, modify, delete,* [assignment: *other operation*s]]

[12] [assignment: *list of security attribute*s]

[13] [assignment: *the authorised identified role*s]

[14] Subset FDP_ACC.1 is selected in this Security Target and is included in FDP_ACC.2. Therefore FDP_IFC.1 and the dependency FDP_IFF.1 is not described in this Security Target.

[15] [assignment: *the authorised identified role*s]

Dependencies: FIA_UID.1 Timing of identification

Refinement: The roles administrator S1 and end user S2 are fixed roles defined by the behaviour of the TOE in test mode and user mode. Users are associated with roles by using different operating modes in the TOE life-cycle (test mode phase 3 and user mode in phase 4 to 7) and implicitly by authentication in test mode: Every user, who can authenticate himself in test mode by using the correct password associated with a test function, is automatically associated with the role administrator, while every user is associated with the role end-user in user mode.

FMT_MSA.3 Static attribute initialisation
Hierarchical to: No other components.
FMT_MSA.3.1 The TSF shall enforce the *Mode-Control-Policy*[16] to provide *fixed*[17] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 The TSF shall allow the *administrator S1* [18] to specify alternative initial values to override the default values when an object or information is created.
Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

Refinement: The Mode-Control-Policy is an access control policy defined as the combination of the Test Mode Security Policy and the User Mode Security Policy as defined in section 5.1.1.1. The only security attribute used to enforce the SFP is the global security attribute “mode-switch”. According to the policy this mode-switch is set to “Test Mode” as a default, when the TOE is produced.
This value can be seen as a “permissive” value, since testing is possible. On the other hand the value can also be seen as a “restrictive” value, since the subject anybody [S2, same as end-user] cannot use security functions. In any way it is the default value required by the policy and therefore the authors used the word “fixed” as “other property” in this selection.

FDP_ACC.2 Complete access control
Hierarchical to: FDP_ACC.1
FDP_ACC.2.1 The TSF shall enforce the *Mode-Control-Policy and the Tamper Protection Security Policy* [19] on *the subjects S1 - S4 and the objects O1 - O8* [20] and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2 The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.
Dependencies: FDP_ACF.1 Security attribute based access control

[16] [assignment: *access control SFP, information flow control SFP*]

[17] [selection: *restrictive, permissive, other property*]

[18] [assignment: *the authorised identified roles*]

[19] [assignment: access control SFP]

[20] [assignment: list of subjects and objects]

Refinement: The Mode-Control-Policy is an access control policy defined as the combination of the Test Mode Security Policy and the User Mode Security Policy as defined in section 5.1.1.1. The Tamper Protection Security Policy is also defined in section 5.1.1.1. The subjects, objects and access methods in the TSC are defined in section 3.1.

FDP_ACF.1 Security attribute based access control
Hierarchical to: No other components.
FDP_ACF.1.1 The TSF shall enforce the *Mode-Control-Policy and the Tamper Protection Security Policy* [21] to objects based on *mode-switch* [22].
FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *(i) in the Test Mode as described in Table 3 (ii) in the User Mode as described in Table 5* [23].
FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:
(i) The TOE allows the administrator S1 to change the mode-switch from the Test mode into the User Mode.
(ii) The TOE prevents the administrator S1 and the end-user S2 to change the mode-switch from the User mode into the Test Mode. [24].
FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the (i) *rules defined for the Tamper Protection Security Policy in section 5.1.1.1, (ii) in the Test Mode as described in Table 4, (iii) in the User Mode as described in Table 6*[25].
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

Refinement: The Mode-Control-Policy is an access control policy defined as the combination of the Test Mode Security Policy and the User Mode Security Policy as defined in section 5.1.1.1. The Tamper Protection Security Policy is also defined in section 5.1.1.1. The subjects, objects and access methods in the TSC are defined in section 3.1.

FDP_IFC.1 Subset information flow control
Hierarchical to: No other components.
FDP_IFC.1.1 The TSF shall enforce the *SFP_{chip}* [26] on *S3 and S4* [27].

[21] [assignment: access control SFP]

[22] [assignment: security attributes, named groups of security attributes]

[23] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

[24] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

[25] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

[26] [assignment: information flow control SFP]

[27] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]

Dependencies: FDP_IFF.1 Simple security attributes

Refinement: The following information flow control $\mathrm{SFP}_{\mathrm{chip}}$ shall be enforced by the TSF:

- The information flow from subject S4 to S3 during the user mode is denied.
- S3 shall be able to store customer specific data in the EEPROM-memory that can be accessed by S4 in the user mode. S3 is only permitted to store user data when the TOE is in the test mode.
- The access of S4 to data and software stored in the EEPROM or in the RAM shall be denied during the test mode performed by S3.

Note: Since the security policies defined in section 5.1.1.1 already include all possibilities for subjects to get access to information (in the form of data objects), the $\mathrm{SFP}_{\mathrm{chip}}$ is only an explicit repetition of parts of these policies.

FDP_IFF.1 Simple security attributes
Hierarchical to: No other components.
FDP_IFF.1.1 The TSF shall enforce the *SFP_{chip}*[28] based on the following types of subject and information security attributes: *S3 or S4 and the mode-switch*[29].
FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: *see SFP_{chip}*[30].
FDP_IFF.1.3 The TSF shall enforce the *no additional information flow control rules*[31].
FDP_IFF.1.4 The TSF shall provide *no additional SFP capabilities*[32].
FDP_IFF.1.5 The TSF shall explicitly authorise an information flow based on the following rules: *see SFP_{chip}*[33].
FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules: *see SFP_{chip}*[34].
Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

Refinement: For the definition of $\mathrm{SFP}_{\mathrm{chip}}$ see the preceding SFR.

[28] [assignment: information flow control SFP]

[29] [assignment: the minimum number and type of security attributes]

[30] [assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]

[31] [assignment: additional information flow control SFP rules]

[32] [assignment: list of additional SFP capabilities]

[33] [assignment: rules, based on security attributes, that explicitly authorise information flows]

[34] [assignment: rules, based on security attributes, that explicitly deny information flows]

FAU_SAA.1 Potential violation analysis
Hierarchical to: No other components.
FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.
FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:
a) Accumulation or combination of
(1) *low frequency of clock input or,*
(2) *high frequency of clock input or,*
(3) *low voltage power supply or*
(4) *high voltage power supply or*
(5) *low temperature or*
(6) *high temperature or*
(7) *high voltage for the write process to the EEPROM*[35].
known to indicate a potential security violation;
b) no other rules.
Dependencies: FAU_GEN.1 Audit data generation

Refinement: --

Note that the PP/9806 states that the dependency of FAU_SAA.1 with FAU_GEN.1 is not applicable to the TOE; the FAU_GEN component forces many security relevant events to be recorded (due to dependencies with other functional security components) and this is not achievable to a smartcard IC considering state-of-the-art implementation. It is then assumed that the function FAU_SAA.1 may still be used and the specific audited events will have to be defined in the ST independently with FAU_GEN.1. Refer to section 8.2.2.

FPR_UNO.1 Unobservability
Hierarchical to: No other components.
FPR_UNO.1.1 The TSF shall ensure that *S1 and S2*[36] are unable to observe the operation *A9*[37] on *O6*[38] by *S3 or S4*[39].
Dependencies: No dependencies.

Refinement: The unobservability scenario “Measuring the external behaviour during the Triple-DES-operation or the CPU-operation” A1 includes:

(1) Differential Power Analysis,
(2) Timing Attacks

[35] [assignment: subset of defined auditable events]

[36] [assignment: list of users and/or subjects]

[37] [assignment: list of operations]

[38] [assignment: list of objects]

[39] [assignment: list of protected users and/or subjects]

FPT_PHP.2 Notification of physical attack
Hierarchical to: FPT_PHP.1
FPT_PHP.2.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.
FPT_PHP.2.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.
FPT_PHP.2.3 For *the power supply block, the internal frequency generation and the chip temperature* [40], the TSF shall monitor the devices and elements and notify a *reset*[41] when physical tampering with the TSF's devices or TSF's elements has occurred.
Dependencies: FMT_MOF.1 Management of security functions behaviour

Refinement: In case of detected affecting, the TOE shall automatic respond preventing the access sets (S1, A5, O1), (S1, A5,O2), (S2, A5, O1), (S2, A5,O2), (S3, A6, O4), (S3, A9, O6), (S4, A6, O4) and (S4, A9, O6).

FPT_PHP.3 Resistance to physical attack
Hierarchical to: No other components.
FPT_PHP.3.1 The TSF shall resist *the physical tampering scenario: Manipulation A4* [42] to the *objects O1 - O8* [43] by responding automatically such that the TSP is not violated.
Dependencies: No dependencies.

Refinement: For the Random Numbers O4 this shall also include the scenario that the attacker tries to modify the quality of the random number generation by tampering attacks.

5.1.1.4 Additional TOE Security Functional Requirements

Additional SFRs, which are specific for this Security Target are specified in this subsection. These SFRs are applicable from phases 3 to 7.

In order to state the IT security functional requirement for the Random Number Generator of the TOE an additional family (FCS_RND) of the Class FCS (cryptographic support) with a security functional component FCS_RND.1 is defined in chapter 9.1 according to Annexes B and C of Common Criteria Part 1 ([1]). This family FCS_RND Generation of random numbers describes the functional requirements for the generation of random numbers (i. e. read A6 the object O4 in the context of this Security Target) used for cryptographic purposes.

FCS_RND.1 Quality metric for random numbers

[40] [assignment: list of TSF devices/elements for which active detection is required]

[41] [assignment: a designated user or role]

[42] [assignment: physical tampering scenarios]

[43] [assignment: list of TSF devices/elements]

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that have an *entropy of at least 7 bit in each byte*[44].

Dependencies: No dependencies.

Refinement: The entropy of the random numbers is measured by the Shannon-Entropy as follows:

$E = -\sum_{i=0}^{255} p_i \cdot \log_2 p_i$, where p_i is the probability that the random number (which takes values between 0 and 255) is equal to i.

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform *encryption and decryption*[45] in accordance with a specified cryptographic algorithm *Triple Data Encryption Algorithm (TDEA)*[46] and cryptographic key sizes *of 112 bit*[47] that meet the following *list of standards*[48]:
U.S. Department of Commerce / National Bureau of Standards
Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25, keying option 2 and according to International Organization for Standardization: Banking – Key Management, International Standard ISO 8732 (1988), Chapter 12.1.3.

Dependencies: [FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction, FMT_MSA.2 Secure security attributes

Refinement: --

5.1.1.5 Explicit SOF claim

Since the assurance level is augmented with AVA_VLA.4 the required level for the Strength of Function of the above listed Security Functional Requirements level is “SOF-high”.

5.1.2 TOE Security Assurance Requirements

The protection profile PP/9806 claims the assurance level EAL 4 augmented by ADV_IMP.2, ALC_DVS.2 and AVA_VLA.4.

[44] [assignment: a defined quality metric]

[45] [assignment: list of crypto-graphic operations]

[46] [assignment: cryptographic algorithm]

[47] [assignment: cryptographic key sizes]

[48] [assignment: list of standards]

This Security Target claims assurance level EAL 5 augmented by ALC_DVS.2, AVA_VLA.4 and AVA_MSU.3.

Since ADV_IMP.2 is part of EAL 5, the claims of the PP are all either contained directly or fulfilled by components hierarchical to them in the level claimed in this Security Target.

5.2 Security Requirements for the Environment

5.2.1 Security Requirements for the IT-Environment

The security objectives for the environment will be ensured by Non-IT security requirements only (see the next subsection and the rationale, section 8.2).

5.2.2 Security Requirements for the Non-IT Environment

This section contains the Security Requirements for the general (Non-IT-) Environment.

Most of the security objectives for the environment defined in PP/9806, section 4.2 and the additional objective OE.SCRAP_DEFECT defined in section 4.2 of this document are of an organisational type. Therefore they require the definition of suitable procedures and methods by the parties involved in the smart card life cycle. These objectives are already stated in the form of requirements (compare e. g. O.DEV_DIS in the PP, which starts “The IC designer must have procedures...”. Therefore it is not necessary to repeat all these requirements here. Instead, the general requirement for the (Non-IT-) environment is to fulfil all requirements already contained in the objectives.

Note: All measures that are related to requirements, which are directed to the IC design and production process, will be included in the documents describing the methods for site security, configuration control and internal delivery.

There is one type of objectives, for which the question has to be considered, if they should be seen as requirements for the IT-environment: These are the objectives, which have implications for the Smartcard Embedded Software, which is a part of the IT-environment. In fact the PP/9806 contains one objective of this type, O.SOFT_MECH. However, in fact this is not a direct technical requirement for the embedded software itself, but a requirement for the developer of this software. Therefore this Security Target doesn’t define Security Functional Requirements in the style of CC, part 2 for this objective.

A special case of the situation discussed above is the usage of the DEA-coprocessor provided by the TOE (defined as a requirement as SFR FCS_COP.1 in section 5.1.1.4). Since this coprocessor gets the cryptographic keys from the Smartcard Embedded Software, the developer of this software has to provide functions for a secure key management. This includes secure key genera-

tion or secure key import (depending on the application) and secure key destruction, where suitable.
It was considered, that these requirements should be defined as formal SFRs for the IT-environment using the SFRs FDP_ITC.1 resp. FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2 from CC, part 2, which are the SFRs appearing in the dependency list for FCS_COP.1.

However, since the smart card embedded software may vary heavily depending on the application, for which it is written, it wasn't seen as suitable to define formal technical requirements. Instead the following requirement holds (which is in fact a special case of O.SOFT_MECH):

The Smartcard Embedded Software shall be designed and developed in a way, which ensures adequate cryptographic key management. This can be measured by checking that the dependencies for FCS_COP.1 defined in CC, Part2 are either

- fulfilled by (functions of the embedded software fulfilling) the SFRs listed above or
- by other functions which cover the dependencies in a different but sufficient way, where the security needs of the application determine the interpretation of the term "sufficient".

6 TOE Summary Specification

This chapter is divided into the following sections:

TOE Security Functions

Assurance measures

6.1 TOE Security Functions

The IT security functions directly correspond to the TOE security functional requirements defined in chapter 5.1.1.

The TSF F.TEST described below is only applicable in phase 3:

F.TEST: The TSF provides test functions to demonstrate the correct operation of the security functions provided by the TOE. This is performed in the Test Mode by a suite of self tests at the request of the authorised user. The test functions also allow to check the integrity of the ROM content and the functions writing data into the EEPROM check the correctness of the data written.
Note: In User Mode these test functions are not available.

The following security functions are applicable to the phases 4 to 7

Note: Some of the security functions are configured at the end of phase 3 and all security functions are already active during the delivery form phase 3 to phase 4.

F.RNG: The random number generator continuously produces random numbers with a length of one byte. Each byte will at least contain a 7 bit entropy. The TOE implements the F.RNG by means of a physical hardware random number generator working stable within the limits guaranteed by F.OPC (operational conditions).

F.DEA: The TOE provides the Triple Data Encryption Algorithm (TDEA) according to the Data Encryption Standard (DES). F.DEA is a modular basic cryptographic function which provides the TDEA algorithm as defined by FIPS PUB 46 by means of a hardware co-processor and supports the 2-key Triple DEA algorithm according to keying option 2 in FIPS PUB 46-3 and according to the ISO standard 8732 (1988). The two 56 bit keys (112 bit) for the 2-key Triple DES algorithm shall be provided by the application software. For encryption the application software provides 8 byte of the plain text and F.DEA calculates 8 byte cipher text. The output of calculation is read by the application software. For decryption the application software also provides 8 byte of cipher text and F.DEA calculates 8 byte plain text. The output of calculation is read by application software.

F.DEA includes specific implementation features to reduce leakage of confidential data via the external interfaces of the TOE to ensure that attackers are unable to utilise the key and plain text by measuring the external behaviour during the Triple DES Operation. These attacks include DPA, DFA, SPA and TA. The specification of the implementation features is beyond the scope of the TOE Summary Specification.

F.OPC: The function F.OPC has the following sub-functions: A function that filters power supply and clock input (called opercond_filter) and a function that monitors the power supply, the frequency of the clock and the temperature of the chip by means of sensors (called opercond_stop).

The filter function of the power smoothes the power consumption and ensures a minimum length of the clock impuls.

The ranges allowed for these parameters are as defined in the SFR FAU_SAA.1 (see section 5.1.1.3).
If one of these parameters is out of the specified range then the function "opercond_stop" shall reset the TOE including a reset of the CPU and the special function register. The TOE holds the reset until the operation conditions abate to the specified range. In addition the application software cannot change the behaviour of the sensors.

F.PHY: The function F.PHY protects the TOE against manipulation of (i) the hardware, (ii) the test software in the ROM, (iii) the application software in the ROM and the E^2PROM, (iv) the application data in the E^2PROM and RAM, (v) the configuration data in the security row and (vi) the mode-switch. The protection of the TOE comprises different features of the construction which makes a tamper attack more difficult.

F.MODE: The function F.MODE provides access control by means of TOE modes of operation selected by a mode-switch: (i) Test Mode and (ii) User Mode. The F.MODE contains 3 sub-functions:

Identification: In the Test Mode the TOE identifies the administrator. In the User Mode the TOE identifies the user.

Access control: In the Test Mode the TOE (i) allows to execute the test software and (ii) prevents to execute the application software. In the User Mode the TOE (i) allows to execute the application software and (ii) prevents to execute the test software.

Mode switch: The initial TOE mode is the Test Mode. The TOE allows to change the mode-switch from the Test mode into the User Mode. If the mode-switch is changed the TOE will remain in an endless loop waiting for a reset to start a program in the User mode. The TOE prevents to change the mode-switch from the User mode into the Test Mode.

Explicit SOF claim

According to the CEM a Security Target shall identify all mechanisms which can be assessed according to the assurance requirement AVA_SOF.1.

The following mechanisms contributing to these functions were identified, which can be analysed for their permutational or probabilistic properties:

1. The output of the Random Number Generator F.RNG can be analysed with probabilistic methods.
2. The quality of the mechanism contributing to the DPA resistance of F.DEA can be analysed using probabilistic methods on power consumption of the TOE.

Therefore an explicit SOF claim of "high" is made for these mechanisms.

Note, that the cryptographic algorithm of F.DEA can also be analysed with permutational or probabilistic methods but that this is not in the scope of CC evaluations.

6.2 Assurance measures

Appropriate assurance measures will be employed to satisfy the security assurance requirements defined in chapter 5.1.2. The developer will provide documents containing the measures and further information needed to examine conformance of the measures to the assurance requirements. The following table gives a mapping between the assurance requirements and the documents containing the information needed for the respective requirement either directly or referring to further documents containing this information.

Document containing or referring the relevant information	Input evidence according to CC Part 3, which is contained or referred to in the document	Input for assurance component(s) (according to developer actions in CC Part 3)
Functional Specification, Philips	semiformal functional specification	ADV_FSP.3
	correspondence analysis between the TOE summary specification and the functional specification	ADV_RCR.2
Formal Model, Philips	TSP model (formal)	ADV_SPM.3
High Level design, Philips	high-level design (semiformal)	ADV_HLD.3
	correspondence analysis between functional specification and high-level design	ADV_RCR.2
Low Level Design, Philips	low level design	ADV_LLD.1
	architectural description	ADV_INT.1
	correspondence analysis between functional specification and high-level design	ADV_RCR.2
	correspondence analysis between high-level design and implementation representation	ADV_RCR.2
Implementation representation, Philips	implementation representation	ADV_IMP.2
Configuration Management and Life Cycle documentation, Philips	configuration management documentation (the configuration list is provided as a separate document)	ACM_AUT.1, ACM_CAP.4, ACM_SCP.3
	development tools documentation	ALC_TAT.2
	development security documentation	ALC_DVS.2
	life cycle definition documentation	ALC_LCD.2
	parts of the delivery documentation	ADO_DEL.2
Guidance, Delivery and Operation, Philips	administrator guidance	AGD_ADM.1, AVA_MSU.3
	secure installation, generation, and start-up procedures	ADO_IGS.1
	user guidance	AGD_USR.1, AVA_MSU.3
	parts of the delivery documentation	ADO_DEL.2
Vulnerability Assessment, Philips	analysis and disposition of vulnerabilities	AVA_VLA.4
	analysis of guidance documentation	AVA_MSU.3
	covert channel analysis documentation	VLA_CCA.1
	strength of function claims analysis	AVA_SOF.1
Test Documentation, Philips	test documentation	ATE_FUN.1, (ATE_IND.2)
	test coverage analysis	ATE_COV.2
	depth of testing analysis	ATE_DPT.2

7 PP Claims

This Security Target claims conformance to the following protection profile:

[5] Protection Profile, Smartcard Integrated Circuit; Common Criteria for Information Technology Security Evaluation; Version 2.0, September 1998, registered at the French Certification Body under number PP/9806

The short term for this protection profile used in this document is PP/9806.

8 Rationale

The chapter *Rationale* is divided into the following sections:

Security Objectives Rationale
Security Requirements Rationale
TOE Summary Specification Rationale
PP Claims Rationale

8.1 Security Objectives Rationale

The security objectives rationale in section 7.2 of the PP/9806 shows, that the threats and assumptions defined in the protection profile are addressed effectively by the security objectives defined in the PP.

This section deals with the additional items defined in this Security Target.

8.1.1 Threats and objectives

No additional threats are defined in this Security Target. The following discussion shows, how the objectives address the threats defined in the PP.

SO1 “Mode Control” controls the possibility to switch between Test Mode and User Mode. By allowing production testing in phase 3 it supports to check the correct operation of the security functions, thereby supporting O.OPERATE and O.FLAW. By this SO1 addresses in fact all threats against the TOE (because every attack may be easier in case of incorrect function of parts of security functionality).
On the other hand it prevents use of test functions in phases 4 to 7 thereby preventing attacks trying to disclose or modify software, data or design information by the use of test functions. In addition the mode control ensures the activation of the security functions in the User Mode of the TOE.

Therefore it addresses (at least) the threats: T.DIS_SOFT, T.DIS_DSOFT, T.DIS_DESIGN, T.DIS_TEST, T.MOD_SOFT, T.MOD_DSOFT, T.MOD_DESIGN as well as T.T_SAMPLE and T.T_PRODUCT in the case of unauthorised use.

SO2 "Operating Conditions Control" supports the continuous correct operation of security functions. When the correct operation cannot be supported because the operation conditions exceed or fall below the defined threshold the TOE goes into a secure state performing an internal reset.
Therefore it addresses the threats: T.DIS_SOFT, T.DIS_DSOFT, T.MOD_SOFT, T.MOD_DSOFT and T.T_SAMPLE, T.T_PRODUCT in the case of unauthorised use.

SO3 "Tamper Proof" supports to prevent disclosure and modification of software, data and design information thereby addressing (at least) T.DIS_SOFT, T.DIS_DSOFT, T.DIS_DESIGN, T.DIS_TEST, T.MOD_SOFT, T.MOD_DSOFT, T.MOD_DESIGN as well as T.T_SAMPLE and T.T_PRODUCT in the case of unauthorised use.

SO4 "Non-existence of Information Leakage " prevents disclosure of secret data while being processed by the TOE. Thereby it addresses T.DIS_SOFT and T.DIS_DSOFT and in addition T.MOD_SOFT, T.MOD_DSOFT and T.MOD_DESIGN regarding the aspect of flaws in the design.

SO5 "Cryptographic support" provide good random numbers and hardware DEA calculation. This helps the embedded software to generate good cryptographic keys and to use DEA for data confidentiality. Thereby it addresses T.DIS_SOFT and T.DIS_DSOFT.

Note: The threat cloning (T.CLON) of the functional behaviour requires to (i) development of a functional equivalent of the Smartcard Embedded Software or disclosure of the Smartcard Embedded Software, (ii) disclose, interpret and employ the secret User Data stored in the TOE, and (iii) develop and build a functional equivalent of the smartcard using the input from the previous steps. Therefore this is a combination of T.DIS_SOFT, T.DIS_DESIGN and T.MOD_SOFT.

In addition the threat T.DIS_SOFT is defined in the PP as follows:
"Unauthorized disclosure of Smartcard Embedded Software and data such as access control, authentication system, data protection system, memory partitioning, cryptographic programs."
Because this threat covers data processed by the Smartcard Embedded Software this also includes end-user data. The objectives SO4 and SO5 are provided to counter this threat.

The preceding discussion shows that the additional objectives defined in this Security Target integrate with those of the PP to address the threats effectively.

8.1.2 Assumptions and objectives

The additional assumption A.SOFT_FUNC (see section 3.3) is addressed by the objectives O.SOFT_MECH and O.DEV_TOOLS. The guidance documentation will contain descriptions, how the embedded software shall check the integrity of user data and that it shall react adequately to loss of integrity. It will also describe how the software can determine that a reset was caused by the sensors and can therefore react correctly. O.SOFT_MECH ensures that the embedded software developer uses this information adequately and O.DEV_TOOLS supports the correctness of this implementation.
The additional assumption A.SCRAP_MARKED is addressed by the additional objective OE.SCRAP_DEFECT defined in section 4.2. Obviously the assumption is fulfilled if the same holds for the objective.

8.2 Security Requirements Rationale

8.2.1 Security Functional Requirements for the TOE

The section 7.3 of the PP/9806 shows that the security requirements defined in the PP are suitable to meet the security objectives of the TOE defined in the PP. The SFRs stated in section 5.1.1.2 of this Security Target are taken from the PP and suitable operations are described for this SFRs. Obviously the reasoning given in the PP isn't affected by these operations. Therefore it holds unchanged.

The additional security objectives defined in section 4.1 are met by the SFRs defined in this Target in the following way:

SO1 "Mode Control": The Test Mode/ User Mode Security Policy (see the policy definitions in 5.1.1.1) is realised by the SFRs connected with Identification/Authentication/Attribute Management/Access control (FMT_MOF.1, FIA_UAU.2, FIA_UID.2, FIA_ATD.1 in section 5.1.1.2, FMT_MSA.1, FMT_SMR.1, FMT_MSA.3, FDP_ACC.2, FDP_ACF.1, FDP_IFC.1, FDP_IFF.1 in section 5.1.1.3). Testing of the TOE in phase 3 in Test Mode is realised by FPT_TST.1 and FDP_SDI.1 (see section 5.1.1.2).

SO2 "Operating Conditions Control": This is realised by FAU_SAA.1 and FPT_PHP.2 (section 5.1.1.3). Regarding FMT_MOF.1 only an authorised user is allowed to disable the sensors.

SO3 "Tamper Proof": SFRs FDP_ACC.2 and FDP_ACF.1 define the enforcement of the Tamper Protection Security Policy (see section 5.1.1.1), which describes this objective. FPT_PHP.3 also adds to this objective.

SO4 "Non-existence of Information Leakage": SFR FPR_UNO.1 defines the impossibility of DPA and Timing attacks.

SO5 “Cryptographic support”: This is defined by the additional SFRs FCS_RND.1 and FCS_COP.1 (see section 5.1.1.4).

Note that the requirement FCS_RND.1 was not taken from CC, part 2 since no requirements on the quality of random number generators exist in the CC, part 2 at present.

The refinements added to the SFRs describe in detail how the SFRs support the objectives. Note that the Security policies derived from the objectives in section 5.1.1.1 are heavily used in all descriptions. The fact that all SFRs work together to realise these policies gives additional assurance that they build a mutually supportive whole.

8.2.2 Security Functional Requirements Dependencies

The discussion of dependencies given in the PP/9806, section 7.3.2 remains valid for all SFRs already contained in the PP.
For the SFR FAU_GEN.1 the following additional explanation is given:
The TOE ensures a secure state and enforces a reset when the events (1) to (6) defined in the Security Functional Requirement FAU_SAA.1 occur. Therefore the TOE cannot audit these events. Nether the less the TOE provides a flag that is used to indicate whether it is a normal reset or a reset enforced by an event that indicate a potential violation. In addition the event (7) defined in the Security Functional Requirement FAU_SAA.1 is indicated by a separate flag. In addition the dependency of FAU_GEN.1 (FPT_STM.1 Reliable time stamps) cannot be fulfilled because state of the art smartcard ICs do not include a internal clock.

In addition the extra SFRs are discussed here.

The SFR FCS_RND.1 defines no dependencies.

FCS_COP.1 defines the following dependencies:

- [FDP_ITC.1 Import of user data without security attributes or FCS_CKM.1 Cryptographic key generation],
- FCS_CKM.4 Cryptographic key destruction,
- FMT_MSA.2 Secure security attributes.

These dependency requirements all address the appropriate management of cryptographic keys used by the specified cryptographic function. All requirements concerning key management shall be fulfilled by the environment according to the discussion in 5.2.2.

It was decided not to include the functional requirements [FDP_ITC.1 or FCS_CKM.1], FCS_CKM.4 and FMT_MSA.2 *explicitly* as security functional requirements for the IT environment because this would mean a restriction for the realisation of the smart card software, which is not justifiable. The possibility was seen that special smart card applications may be designed that are able to resolve the dependencies without use of all the explicit functional requirements (for example by moving some of the functional responsibilities to organisational measures outside of the smart card). So the more abstract general requirement discussed in 5.2.2 were chosen to give the developers of the smart card software the freedom to choose how to fulfil them.

The same argument holds for further indirect dependencies of FCS_COP.1.1, which exist according to CC Part 2 [2]: FDP_ACC.1, FDP_IFC.1, FDP_ACF.1, FDP_IFF.1, FMT_MSA.3, FCS_CKM.2, FMT_MSA.1, FMT_SMR.1, FIA_UID.1 and ADV_SPM.1 (Note that many of these requirements appear in this Security Target, but not with respect to the cryptographic key management for the DEA coprocessor.).

8.2.3 Strength of function level rationale

The reasoning given for the level "high" already in the PP is also valid for this Security Target.

8.2.4 Security assurance requirements rationale

The assurance level EAL5 was chosen in order to meet high assurance expectations as for example required by the German Digital Signature Act.

The augmentation with the requirements ALC_DVS.2, AVA_VLA.4 and AVA_MSU.3 was chosen to additionally support this. In particular it is expected that attackers with high attack potential will try to attack high-end digital signature applications. Similar reasoning holds for electronic payment systems, where highly skilled hackers might try to compromise the system. These considerations show that these high assurance requirements are adequate.

The assurance components in an EAL level are chosen in a way that they build a mutually supportive and complete set of components. The requirements chosen for augmentation do not add any dependencies, which are not already fulfilled for the corresponding requirements contained in EAL5. Therefore, these components add additional assurance to EAL5 but the mutual support of the requirements is still guaranteed.

8.2.5 Security Requirements are mutually supportive and internally consistent

The discussions of SFRs and assurance components in the preceding sections has shown that mutual support and consistency are given for both groups of requirements. The arguments given for the fact that the assurance components are adequate for the functionality of the TOE also shows that the SFRs and assurance requirements support each other and that there are no inconsistencies between these groups.

8.3 TOE Summary Specification Rationale

8.3.1 TOE security functions

Note: This chapter (8.3.1) was not intended to be published

8.3.2 Assurance measures

Note: This chapter (8.3.2) was not intended to be published.

8.4 PP Claims Rationale

According to section 7 this Security Target claims conformance to the PP/9806 Protection Profile.

The sections of this document, where threats, objectives and security requirements are defined, clearly state, which of these items are taken from the Protection Profile and which are added in this ST. Therefore this is not repeated here. In addition the items added in this Security Target do not contradict to the items included in the Protection Profile. The operations done for the SFRs taken from the PP are also clearly indicated.

The assurance level claimed for this target (EAL5+) is shown in section 5.1.2 to include resp. exceed the requirements claimed by the PP (EAL4+).

These considerations show that the Security Target correctly claims conformance to the PP/9806.

9 Annexes

9.1 Definition of specific IT security functional requirements

To define the IT security functional requirements of the TOE an additional family (FCS_RND) of the Class FCS (cryptographic support) is defined here. This class describes the functional requirements for random number generation used for cryptographic purposes.

FCS_RND Generation of random numbers

Family behaviour
This family defines quality requirements for the generation of random numbers which are intended to be use for cryptographic purposes.
Component levelling

FCS_RND Generation of random numbers
1

FCS_RND.1 Generation of 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
Hierarchical to: No other components.
FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that have an [assignment: a defined quality metric].
Dependencies: No dependencies.

9.2 Abbreviations

IC	Integrated Circuit
O	Object
OTP	One Time Programmable
PP	Protection Profile
S	Subject
SEF	Security Enforcing Function
SF	Security function
SFP	Security Function Policies
SOF	Strength of function
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security functions
TSP	TOE Security Policy

9.3 Further Documents

[6] Data Encryption Standard (DES), FIBS PUB 46, US NBS, 1977, Washington,

[7] FIPS PUB 46-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION DATA ENCRYPTION STANDARD (DES) Reaffirmed 1999 October 25

[8] International Organization for Standardization: Banking – Key Management, International Standard ISO 8732 (1988), Chapter 12.1.3