PUBLIC

Common Criteria

Information Technology

Security Evaluation

Project S3FT9MD/MC

Security Target Lite of

Samsung

16-bit RISC Microcontroller

for Smart Card

Version 1.1

22th November 2013

SAMSUNG ELECTRONICS

REVISION HISTORY

UPDATES:

Version	Date	Modification
1.0	18th July 2013	– Creation
1.1	22th November 2013	– Update the table 1
		–
		–
		–
		–
		–
		–

WRITERS:

Written by	Title
Sung-Kyoung Kim	Senior Engineer

APPROVAL:

Written by	Title
JungHyun KIM	Principal Engineer

DISTRIBUTION:

Name	Company/Occupation	Copy
Hubert Pujol	ANSSI	1/4
Olivier ROUSIERE	LETI	2/4
Elisabeth CROCHON	LETI	3/4
JungHyun KIM	Samsung Electronics	4/4

CONTENTS

1 ST INTRODUCTION

This introductory chapter contains the following sections:

1.1 Security Target and TOE Reference

1.2 TOE Overview and TOE Description

1.3 Interfaces of the TOE

1.4 TOE Intended Usage

1.1 Security Target and TOE Reference

The Security Target Lite version is 1.1 and dated 22th November 2013

The Security Target Lite is based on

[5] Eurosmart Security IC Platform Protection Profile, Version 1.0, June 2007, BSI-PP-0035.

The Protection Profile and the Security Target are built on *Common Criteria version 3.1*.

- Title: Security Target Lite of S3FT9MD/S3FT9MC 16-Bit RISC Microcontroller for Smart Cards
- Target of Evaluation: S3FT9MD/S3FT9MC
- **TOE reference: S3FT9MD/S3FT9MC_rev0_SW10-43-50_GU14-11-13-11-15-14-30"**
- Provided by: Samsung Electronics Co., Ltd.
- Common Criteria version :

[1] Common Criteria, Part 1: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-001

[2] Common Criteria, Part 2: Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-002

[3] Common Criteria, Part 3: Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-004

1.2 TOE Overview and TOE Description

1.2.1 Introduction

The Target of Evaluation (TOE), the S3FT9MD/S3FT9MC is a smartcard integrated circuit which is composed of a processing unit, security components, contactless and contact based I/O ports, hardware circuit for testing purpose during the manufacturing process and volatile and non-volatile memories (hardware). The TOE also includes any IC Designer/Manufacturer proprietary IC Dedicated Software as long as it physically exists in the smartcard integrated circuit after being delivered by the IC Manufacturer. Such software (also known as IC firmware) is used for testing purpose during the manufacturing process but also provides additional services to facilitate the usage of the hardware and/or to provide additional services, including an [6]AIS31 compliant random number generation library and an [6]AIS31 compliant random number generator. All other software is called Smartcard Embedded Software and is not part of the TOE.

The only difference between S3FT9MD and S3FT9MC is at the FLASH memory size in a logical meaning, say, S3FT9MD (160KB) and S3FT9MC (136KB), which means that all 2 microcontrollers have the same layout.

1.2.2 TOE Definition

The S3FT9MD/S3FT9MC single-chip CMOS micro-controller is designed and packaged specifically for "Smart Card" applications.

The CalmRISC16 CPU architecture of the S3FT9MD/S3FT9MC microcontroller follows the Harvard style, that is, it has separate program memory and data memory. Both instruction and data can be fetched simultaneously without causing a stall, using separate paths for memory access.

The main security features of the S3FT9MD/S3FT9MC integrated circuit are:

- Security sensors or detectors or filters
- Dedicated tamper-resistant design based on synthesizable glue logic and secure topology
- Dedicated hardware mechanisms against side-channel attacks
- Secure DES and AES Symmetric Cryptography support
- One Hardware True Random Number Generator (DTRNG) that meet P2 class of BSI-AIS31 (German Metric).
- The IC Dedicated Software includes:
  - A DTRNG library built around Hardware DTRNG together with a DTRNG application note that meet the "standard" level of ANSSI requirements (French metric) as well as P2 class of BSI-AIS31 (German Metric).

*Note that only the Triple DES algorithm belongs to the TOE, not the Single DES.

The TOE consists of the following Hardware and Software:

TOE Hardware

- 160Kbytes(S3FT9MD), 136Kbytes(S3FT9MC) FLASH / 4K bytes RAM / 32Kbytes ROM / 768 Bytes Flash special area / 512Bytes DMA RAM
- 16-bit Central Processing Unit (CPU)
- Internal Voltage Regulator (IVR)
- Detectors & Security Logic
- Filters
- True random number generator (DTRNG) and **Bilateral Pseudo Random Number Generator (BPRNG)**
- Memory Protection Unit (MPU)
- Triple DES cryptographic coprocessor with 112 or 168 bits key size(version 2.5)
- AES cryptographic coprocessor with 128 bits, 192bits and 256bits key size(version 4.1)
- Hardware UART for contact and contactless I/O modes with DMA RAM
- Address & data buses
- Internal Clock
- Timers

- Power on Reset
- Error Correcting Code (ECC)

TOE Software

The TOE software comprises the following components:

- A Digital True Random Number Generator (DTRNG) library that fulfills the requirements of *AIS 31*, Class P2 as well as the "standard" level of ANSSI (French metric).
- Secure Boot Loader can download the encrypted user code with AES

The TOE configuration is summarized in table 1 below:

Item Type	Item	Version	Date	Form of delivery
Hardware	S3FT9MD/S3FT9MC 16-Bit RISC Microcontroller for Smart Card	0	-	Wafer or Module
Software	Test ROM Code	1.0	-	- Included in S3FT9MD/MC Test ROM - Test ROM code is not part of the TOE.
Software	Secure Boot loader code (S3FT9xx_Bootloader_20130521_1700_MD_4_3.zip)	4.3	2013.05.21	Included in S3FT9MD/MC in ROM
Software	DTRNG FRO library (S3FT9XX_DTRNG_lib_v5.0_LETI_delivery_20130928.zip)	5.0	2013.09.28	Software Library. This library is delivered as object file and is optionally integrated into user NVM code.
Document	DTRNG FRO Application Note (S3FT9XX_DTRNG_FRO_AN_V1.4.pdf)	1.4	2013.11.20	Softcopy
Document	Hardware User's manual (S3FT9XX_UM_REV1.10.pdf)	1.1	2013.09.30	Softcopy
Document	Security Application Note (SAN_S3FT9MD_MF_MH_v1.3.pdf)	1.3	2013.11.20	Softcopy
Document	Chip Delivery Specification (S3FT9MD_DV11.pdf)	1.1	2013.11	Softcopy
Document	Boot Loader Specification (S3FT9xx_80nm_BootloaderSpecification_v1.5.pdf)	1.5	2012.12.21	Softcopy
Document	Architecture Reference: SecuCalm CPU Core	14		Softcopy
Document	Errata for UM and B/L specification (TN03_80nm FSID Devices_for_Bootloader_User's_Manual_Guide_20131122.pdf)	3.0	2013.11.22	Softcopy

Table 1. TOE Configuration

TEST mode and NORMAL mode

TEST mode	NORMAL mode
TEST mode of the TOE provides full access to all security registers and memory area available in	In NORMAL mode of the TOE, TOE can no longer go back to TEST mode domain again since

TOE's specification to verify full functionalities	traceability data are written in the non-volatile memory of the TOE which cannot be altered after it has been written to. The NORMAL mode consists of PRIVILEGE mode and USER mode.

Table 2. Test and Normal Modes basic description

PRIVILEGE mode and USER mode

PRIVILEGE mode	USER mode
Protected mode for the operating system. All the control registers including security related special registers can be read or written only if CPU runs in this mode. Memory Protection Unit can be configured in this Privilege Mode. When the CPU enters an interrupt service routine, it goes into Privilege Mode. Switching to User Mode will be done automatically when it returns from interrupt service routine. But the only way to switch from User Mode to Privilege Mode is via interrupts including SWI instructions. The FE, IE, TE and PM bits can be modified only when PM = 1 (privilege mode).	This mode cannot access all control registers. Interrupts including SWI is only way to switch from User Mode to Privilege Mode. When the program returns from an interrupt service routine, it goes back to User Mode again.

Table 3. Privilege and User Modes basic description

1.2.3 TOE Features

CPU

- 16-bit SecuCalm core

Memory

- Program Memory (ROM)
- Test ROM
- 160Kbytes(S3FT9MD), 136Kbytes(S3FT9MC) Data/Program Memory (FLASH)
- 768 bytes Data Memory(FLASH)
- Data Memory (RAM)
- DMA RAM

Triple DES

- Built-in hardware Triple DES accelerator
- Circuit for resistance against SPA, DPA and safe error attacks

AES

- Built-in hardware AES accelerator

Filters

- Filters

Interrupts

- Two interrupt sources and vectors (FIQ,IRQ)
- Source for FIQ: Invalid memory access
- Sources for IRQ:
- SIO Falling edge
- 16-bit Timer
- Watchdog Timer
- Contact UART Tx/Rx
- Contactless Type Tx/Rx
- Software Interrupts

Serial I/O Interface

- T=0 and 1 (ISO 7816-3)
- Type A and Type B contactless interfaces compliant with the ISO 14443 standard

Reset and Power Down Mode

- Power-on reset and external reset
- Stop mode

Random Number Generator

- A Digital True random number generator (DTRNG) **: AIS31 compliant**
- A **Bilateral Pseudo Random Number Generator (BPRNG): no compliance to any specific metric, but BPRNG is used by the chip for internal use**

Memory Protection Unit

The MPU allow the CPU to access memories through channels. Each channel can allow the access to a contiguous range of address.

The following channels are provided:

- 3 NVM Program Memory channels: allow program fetch in NVM memories
- 1 RAM Program Memory channel
- 2 NVM Data Memory channels: allow data access in NVM memories
- 3 RAM Data Memory channels
- Fixed Data Memory channels for each special memory block: SFLASH (FLASH products only)

Memory Encryption and Bus Scrambling

- Scrambling and encryption

Timers

- 16-Bit Timer with 8 Bit prescaler

- 20-bit Watchdog Timer

ECC
- ECC on Flash

Clock Sources
- External clock
- Internal clock

Operating Voltage Range
- 1.62 V - 5.5 V

Operating Temperature
- - 25°C to 85°C

Package
- Wafer
- 8-pin COB (compliant with ISO 7816)

1.2.4 TOE Life cycle

The complex development and manufacturing processes of a Composite Product can be separated into seven distinct phases. The phases 2 and 3 of the Composite Product life cycle cover the IC development and production:

- IC Development (Phase 2):
  - IC design,
  - IC Dedicated Software development,

- the IC Manufacturing (Phase 3):
  - integration and photomask fabrication,
  - IC production,
  - IC testing,
  - preparation and
  - Pre-personalisation if necessary

The Composite Product life cycle phase 4 can be included in the evaluation of the IC as an option:

- the IC Packaging (Phase 4):
  - Security IC packaging (and testing),
  - Pre-personalisation if necessary.

In addition, three important stages have to be considered in the Composite Product life cycle:

- Security IC Embedded Software Development (Phase 1),

- the Composite Product finishing process, preparation and shipping to the personalisation line for the Composite Product (Composite Product Integration Phase 5),

- the Composite Product personalisation and testing stage where the User Data is loaded into the Security IC's memory (Personalisation Phase 6),

- the Composite Product usage by its issuers and consumers (Operational Usage Phase 7) which may include loading and other management of applications in the field.

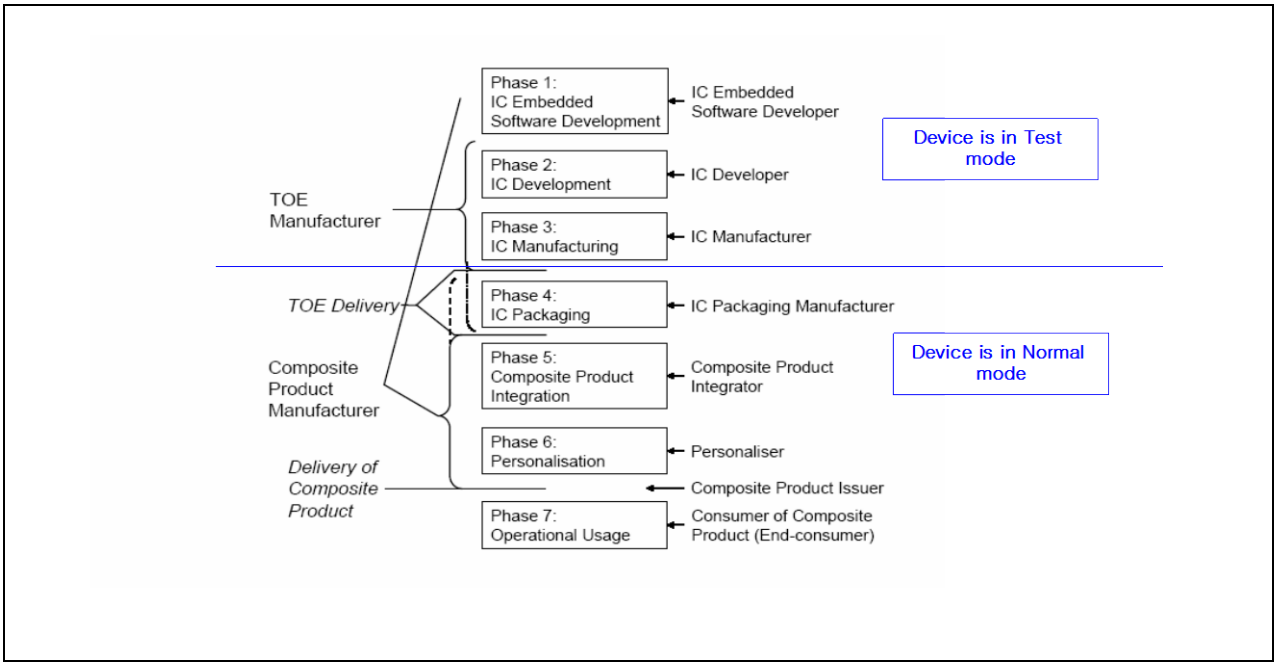


Figure 1: Definition of "TOE Delivery" and responsible Parties

The Security IC Embedded Software is developed outside the TOE development in Phase 1. The TOE is developed in Phase 2 and produced in Phase 3. Then the TOE is delivered in form of wafers. The TOE can also be delivered in form of packaged products. In this case, the development and production of the TOE not only pertain to Phase 2 and 3 but to Phase 4 in addition.

1.3 Interfaces of the TOE

- The physical interface of the TOE with the external environment is the entire surface of the IC
- The electrical interface of the TOE with the external environment is made of the chip's pads including the Vdd, RESETB, XCLK, GND, IO1, L1 and L2 as well as the contactless radio-frequency interface
- The data interface of the TOE is made of the Contact I/O pads and Contactless I/O pads.
- The software interface of the TOE with the hardware consists of Special Function Registers (SFR) and CPU instructions.
- The TRNG interface of the TOE is defined by the DTRNG library interface.

1.4 TOE Intended Usage

The TOE is dedicated to applications such as:

- Banking and finance applications for credit or debit cards, electronic purse (stored value cards) and electronic commerce.
- Network based transaction processing such a mobile phones (GSM SIM cards), pay TV (subscriber and pay-per-view cards), communication highways (Internet access and transaction processing).
- Transport and ticketing applications (access control cards).

- Governmental cards (ID cards, health cards, driving licenses).
- Multimedia applications and Digital Right Management protection.

2 CONFORMANCE CLAIMS

This chapter 2 contains the following sections:

2.1 CC Conformance Claim

2.2 PP Claim

2.3 Package Claim

2.4 Conformance Claim Rationale

2.1 CC Conformance Claim

This Security target claims to be conformant to the Common Criteria version 3.1 R4.

Furthermore it claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in chapter 5.

This *Security Target* has been built with the Common Criteria for Information Technology Security Evaluation; Version 3.1 which comprises

[1] Common Criteria, Part 1: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-001

[2] Common Criteria, Part 2: Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-002

[3] Common Criteria, Part 3: Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-004

has been taken into account.

2.2 PP Claim

This Security Target is strict compliant to the Security IC Platform Protection Profile [5]. The Security IC Platform Protection Profile is registered and certified by the Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0035, Version 1.0, dated 15.06.2007.

This ST does not claim conformance to any other PP.

2.3 Package Claim

The assurance level for this Security Target is EAL5 augmented with AVA_VAN.5 and ALC_DVS.2.

2.4 Conformance Claim Rationale

This security target claims strict conformance only to one PP, the Security IC Platform Protection Profile [5].

The Evaluation Assurance Level (EAL) of the PP [5] is EAL 4 augmented with the assurance components ALC_DVS.2 and AVA_VAN.5. The Assurance Requirements of the TOE obtain the Evaluation Assurance Level 5 augmented with the assurance components ALC_DVS.2 and AVA_VAN.5 for the TOE.

The Target of Evaluation (TOE) is a complete solution implementing a security integrated circuit (security IC) as defined in the PP [5] section 1.3.1, so the TOE is consistent with the TOE type in the PP [5].

The security problem definition of this security target is consistent with the statement of the security problem definition in the PP [5], as the security target claimed strict conformance to the PP [5]. Additional threats, organisational security policies and assumptions are introduced in chapter 3 of this ST, a rationale is given in chapter 4.4.

The security objectives of this security target are consistent with the statement of the security objectives in the PP [5], as the security target claimed strict conformance to the PP [5]. Additional security objectives are added in chapter 4.1 of this ST, a rationale is given in chapter 4.4.

The security requirements of this security target are consistent with the statement of the security requirements in the PP [5], as the security target claimed strict conformance to the PP [5]. Additional security requirements are added in chapter 6.1 of this ST, a rationale is given in chapter 6.3. All assignments and selections of the security functional requirements are done in the PP [5] and in this security target section 6.1.

3 SECURITY PROBLEM DEFINITION

This chapter 3 contains the following sections:

3.1 Description of Assets

3.2 Threats

3.3 Organizational Security Policies

3.4 Assumptions

3.1 Description of Assets

Assets regarding the Threats

The assets (related to standard functionality) to be protected are

- the User Data,
- the Security IC Embedded Software,
- the security services provided by the TOE for the Security IC Embedded Software.

The user (consumer) of the TOE places value upon the assets related to high-level security concerns:

SC1 integrity of User Data and of the Security IC Embedded Software (while being executed/processed and while being stored in the TOE's memories),

SC2 confidentiality of User Data and of the Security IC Embedded Software (while being processed and while being stored in the TOE's memories)

SC3 correct operation of the security services provided by the TOE for the Security IC Embedded Software.

The Security IC may not distinguish between User Data which are public known or kept confidential. Therefore the security IC shall protect the confidentiality and integrity of the User Data, unless the Security IC Embedded Software chooses to disclose or modify it.

In particular integrity of the Security IC Embedded Software means that it is correctly being executed which includes the correct operation of the TOE's functionality. Though the Security IC Embedded Software (normally stored in the ROM) will in many cases not contain secret data or algorithms, it must be protected from being disclosed, since for instance knowledge of specific implementation details may assist an attacker.

The Protection Profile requires the TOE to provide one security service: the generation of random numbers by means of a physical Random Number Generator. The Security Target may require additional security services. It is essential that the TOE ensures the correct operation of all security services provided by the TOE for the Security IC Embedded Software.

According to the Protection Profile there is the following high-level security concern related to security service:

SC4 deficiency of random numbers.

To be able to protect these assets the TOE shall protect its security functionality. Therefore critical information about the TOE shall be protected. Critical information includes:

- logical design data, physical design data, IC Dedicated Software, and configuration data,
- Initialisation Data and Pre-personalisation Data, specific development aids, test and characterisation related data, material for software development support, and photomasks.

Such information and the ability to perform manipulations assist in threatening the above assets.

Note that there are many ways to manipulate or disclose the User Data: (i) An attacker may manipulate the Security IC Embedded Software or the TOE. (ii) An attacker may cause malfunctions of the TOE or abuse Test Features provided by the TOE. Such attacks usually require design information of the TOE to be obtained. They pertain to all information about (i) the circuitry of the IC (hardware including the physical memories), (ii) the IC Dedicated Software with the parts IC Dedicated Test Software (if any) and IC Dedicated Support Software (if any), and (iii) the configuration data for the security functionality. The knowledge of this information enables or supports attacks on the assets. Therefore the TOE Manufacturer must ensure that the development and production of the TOE is secure so that no information is unintentionally made available for the operational phase of the TOE.

The TOE Manufacturer must apply protection to support the security of the TOE. This not only pertains to the TOE but also to all information and material exchanged with the developer of the Security IC Embedded Software. This covers the Security IC Embedded Software itself if provided by the developer of the Security IC Embedded Software or any authentication data required to enable the download of software. This includes the delivery (exchange) procedures for Phase 1 and the Phases after TOE Delivery as far as they can be controlled by the TOE Manufacturer. These aspects enforce the usage of the supporting documents and the refinements of SAR defined in the protection profile.

The information and material produced and/or processed by the TOE Manufacturer in the TOE development and production environment (Phases 2 up to TOE Delivery) can be grouped as follows:

- logical design data,
- physical design data,
- IC Dedicated Software, Security IC Embedded Software, Initialisation Data and Pre-personalisation Data,
- specific development aids,
- test and characterisation related data,
- material for software development support, and
- photomasks and products in any form

as long as they are generated, stored, or processed by the TOE Manufacturer.

3.2 Threats

The following explanations help to understand the focus of the threats and objectives defined below. For example, certain attacks are only one step towards a disclosure of assets; others may directly lead to a compromise of the application security.

- Manipulation of data (which may comprise any data, including code, stored in or processed by the Security IC) means that an attacker is able to alter a meaningful block of data. This should be considered for the threats T.Malfunction, T.Phys-Manipulation and T.Abuse-Func.

- Manipulation of the TOE means that an attacker is able to deliberately deactivate or otherwise change the behaviour of a specific function in a manner which enables exploitation. This should be considered for the threat T.Malfunction, T.Phys-Manipulation and T.Abuse-Func.

- Disclosure of data (which may comprise any data, including code, stored in or processed by the Security IC) means that an attacker is realistically[1] able to determine a meaningful block of data.

[1] taking into account the assumed attack potential (and for instance the probability of errors)

This should be considered for the threats T.Leak-Inherent, T.Phys-Probing, T.Leak-Forced and T.Abuse-Func.

The cloning of the functional behaviour of the Security IC on its physical and command interface is the highest level security concern in the application context.

The cloning of that functional behaviour requires to (i) develop a functional equivalent of the Security IC 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 Security IC using the input from the previous steps.

The Security IC is a platform for the Security IC Embedded Software which ensures that especially the critical User Data are stored and processed in a secure way (refer to below). The Security IC Embedded Software must also ensure that critical User Data are treated as required in the application context. In addition, the personalisation process supported by the Security IC Embedded Software (and perhaps by the Security IC in addition) must be secure. This last step is beyond the scope of the Protection Profile. As a result the threat "cloning of the functional behaviour of the Security IC on its physical and command interface" is averted by the combination of measures which split into those being evaluated according to the Protection Profile (Security IC) and those being subject to the evaluation of the Security IC Embedded Software or Security IC and the corresponding personalisation process. Therefore, functional cloning is indirectly covered by the security concerns and threats described below.

The high-level security concerns are refined below by defining threats as required by the Common Criteria (refer to Figure 3). Note that manipulation of the TOE is only a means to threaten User Data or the Security IC Embedded Software and is not a success for the attacker in itself.

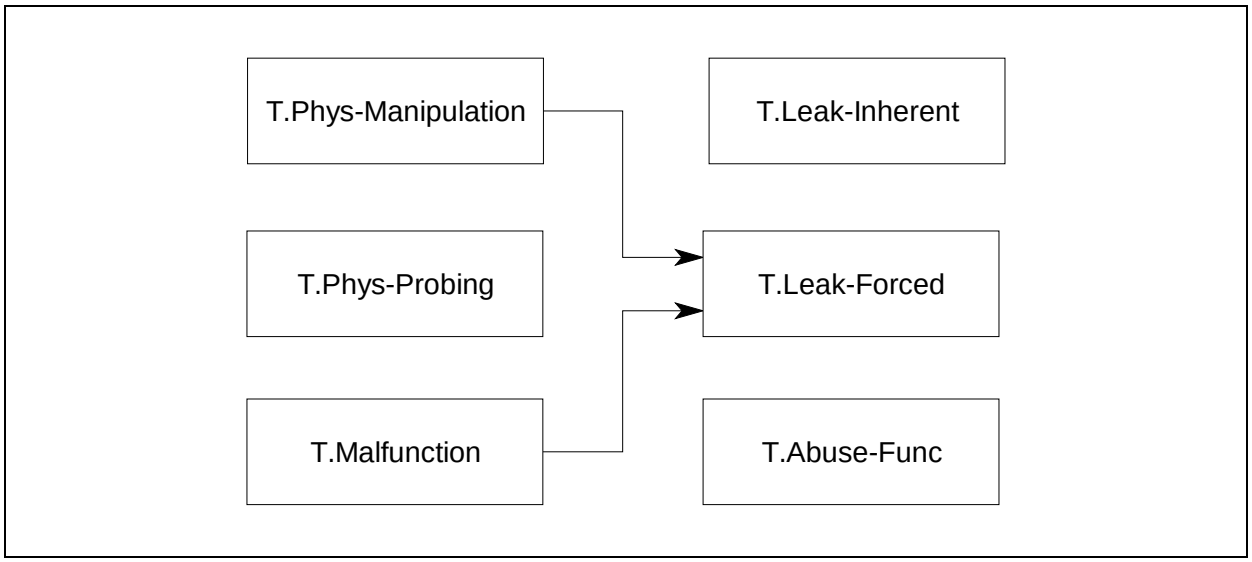


Figure 3: Standard Threats

The high-level security concern related to security service is refined below by defining threats as required by the Common Criteria (refer to Figure 4).

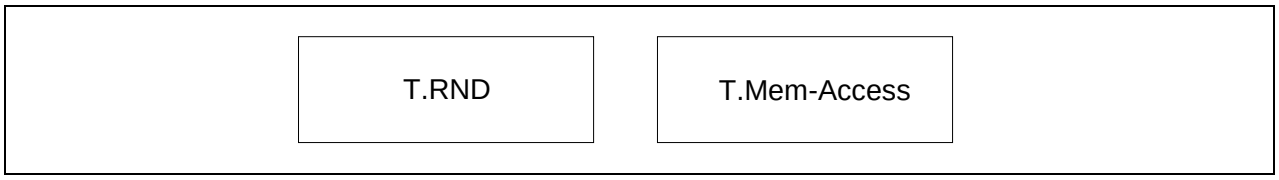


Figure 4: Threats related to security service

The Security IC Embedded Software must contribute to averting the threats: At least it must not undermine the security provided by the TOE.

The above security concerns are derived from considering the end-usage phase (Phase 7) since

- Phase 1 and the Phases from TOE Delivery up to the end of Phase 6 are covered by assumptions and
- the development and production environment starting with Phase 2 up to TOE Delivery are covered by an organisational security policy.

The TOE's countermeasures are designed to avert the threats described below. Nevertheless, they may be effective in earlier phases (Phases 4 to 6).

The TOE is exposed to different types of influences or interactions with its outer world. Some of them may result from using the TOE only but others may also indicate an attack. The different types of influences or interactions are visualised in Figure 5. Due to the intended usage of the TOE all interactions are considered as possible.

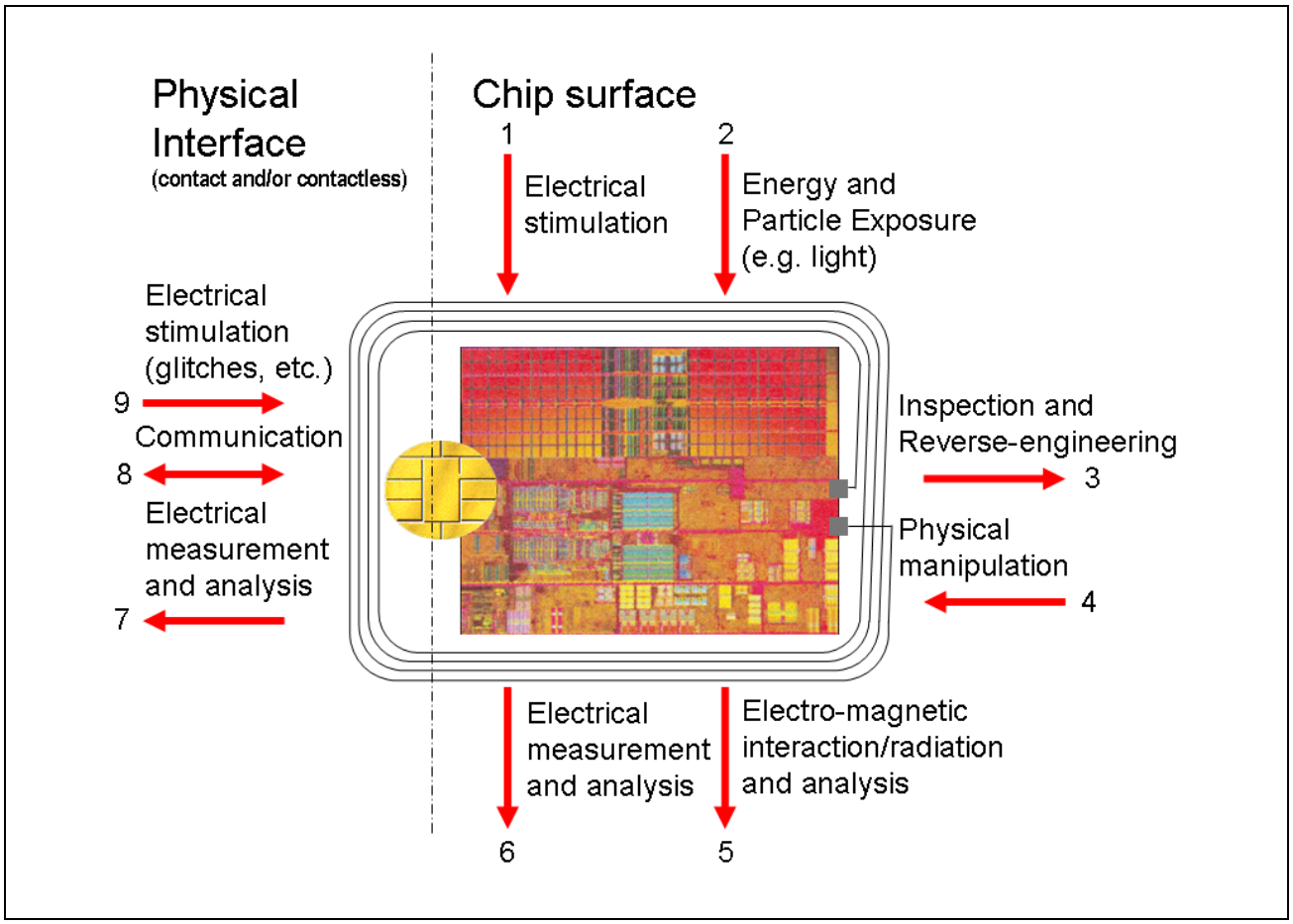


Figure 5: Interactions between the TOE and its outer world

An interaction with the TOE can be done through the physical interfaces (Number 7 – 9 in Figure 5) which are realised using contacts interface. Influences or interactions with the TOE also occur through the chip surface (Number 1 – 6 in Figure 5). In Number 1 and 6 galvanic contacts are used. In Number 2 and 5 the influence (arrow directed to the chip) or the measurement (arrow starts from the chip) does not require a contact. Number 3 and 4 refer to specific situations where the TOE and its functional behaviour is not only influenced but definite changes are made by applying mechanical, chemical and other methods (such as 1, 2). Many attacks require a prior inspection and some reverse-engineering (Number 3). This demonstrates the basic building blocks of attacks. A practical attack will use a combination of these elements.

3.2.1 Standard Threats

The TOE shall avert the threat "Inherent Information Leakage (T.Leak-Inherent)" as specified below.

T.Leak-Inherent	Inherent Information Leakage An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential data as part of the assets.

No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. One example is the Differential Power Analysis (DPA). This leakage may be interpreted as a covert channel transmission but is more closely related to measurement of operating parameters, which may be derived either from direct (contact) measurements (Numbers 6 and 7 in Figure 5) or measurement of emanations (Number 5 in Figure 5) and can then be related to the specific operation being performed.

The TOE shall avert the threat "Physical Probing (T.Phys-Probing)" as specified below.

T.Phys-Probing	Physical Probing An attacker may perform physical probing of the TOE in order (i) to disclose User Data, (ii) to disclose/reconstruct the Security IC Embedded Software or (iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.

Physical probing requires direct interaction with the Security IC internals (Numbers 5 and 6 in Figure 5). Techniques commonly employed in IC failure analysis and IC reverse engineering efforts may be used. Before that hardware security mechanisms and layout characteristics need to be identified (Number 3 in Figure 5). Determination of software design including treatment of User Data may also be a pre-requisite.

This pertains to "measurements" using galvanic contacts or any type of charge interaction whereas manipulations are considered under the threat "Physical Manipulation (T.Phys-Manipulation)". The threats "Inherent Information Leakage (T.Leak-Inherent)" and "Forced Information Leakage (T.Leak-Forced)" may use physical probing but require complex signal processing in addition.

The TOE shall avert the threat "Malfunction due to Environmental Stress (T.Malfunction)" as specified below.

T.Malfunction	Malfunction due to Environmental Stress An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to (i) modify security services of the TOE or (ii) modify functions of the Security IC Embedded Software (iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC

Embedded Software. This may be achieved by operating the Security IC outside the normal operating conditions (Numbers 1, 2 and 9 in Figure 5).

The modification of security services of the TOE may e.g. affect the quality of random numbers provided by the random number generator up to undetected deactivation when the random number generator does not produce random numbers and the Security IC Embedded Software gets constant values. In another case errors are introduced in executing the Security IC Embedded Software. To exploit an attacker needs information about the functional operation, e.g. to introduce a temporary failure within a register used by the Security IC Embedded Software with light or a power glitch.

The TOE shall avert the threat "Physical Manipulation (T.Phys-Manipulation)" as specified below.

T.Phys-Manipulation Physical Manipulation

An attacker may physically modify the Security IC in order to (i) modify User Data, (ii) modify the Security IC Embedded Software, (iii) modify or deactivate security services of the TOE, or (iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software.

The modification may be achieved through techniques commonly employed in IC failure analysis (Numbers 1, 2 and 4 in Figure 5) and IC reverse engineering efforts (Number 3 in Figure 5). The modification may result in the deactivation of a security feature. Before that hardware security mechanisms and layout characteristics need to be identified. Determination of software design including treatment of User Data may also be a pre-requisite. Changes of circuitry or data can be permanent or temporary.

In contrast to malfunctions (refer to T.Malfunction) the attacker requires gathering significant knowledge about the TOE's internal construction here (Number 3 in Figure 5).

The TOE shall avert the threat "Forced Information Leakage (T.Leak-Forced)" as specified below:

T.Leak-Forced Forced Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential data as part of the assets even if the information leakage is not inherent but caused by the attacker.

This threat pertains to attacks where methods described in "Malfunction due to Environmental Stress" (refer to T.Malfunction) and/or "Physical Manipulation" (refer to T.Phys-Manipulation) are used to cause leakage from signals (Numbers 5, 6, 7 and 8 in Figure 5) which normally do not contain significant information about secrets.

The TOE shall avert the threat "Abuse of Functionality (T.Abuse-Func)" as specified below.

T.Abuse-Func Abuse of Functionality

An attacker may use functions of the TOE which may not be used after TOE Delivery in order to (i) disclose or manipulate User Data, (ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or (iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or (iv) enable an attack disclosing or manipulating the User Data or the Security IC Embedded Software.

3.2.2 Threats related to security services

The TOE shall avert the threat "Deficiency of Random Numbers (T.RND)" as specified below.

T.RND Deficiency of Random Numbers

An attacker may predict or obtain information about random numbers generated by the TOE for instance because of a lack of entropy of the random numbers provided.

An attacker may gather information about the produced random numbers which might be a problem because they may be used for instance to generate cryptographic keys.

Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the TOE without specific knowledge about the TOE's generator. Malfunctions or premature ageing are also considered which may assist in getting information about random numbers.

3.2.3 Threats related to additional TOE Specific Functionality

The TOE shall avert the additional threat "Memory Access Violation (T.Mem-Access)" as specified below.

T.Mem-Access Memory Access Violation

Parts of the IC Smartcard Embedded Software may cause security violations by accidentally or deliberately accessing restricted data (which may include code). Any restrictions are defined by the security policy of the specific application context and must be implemented by the Smartcard IC Embedded Software.

Clarification: This threat does not address the proper definition and management of the security rules implemented by the Security IC Embedded Software, this being software design and correctness issue. This threat addresses the reliability of the abstract machine targeted by the software implementation. To avert the threat, the set of access rules provided by this TOE should be undefeated if operated according to the provided guidance. The threat is not realized if the Security IC Embedded Software is designed or implemented to grant access to restricted information. It is realized if an implemented access denial is granted under unexpected conditions or if the execution machinery does not effectively control a controlled access.

Here the attacker is expected to (i) take advantage of flaws in the design and/or the implementation of the TOE memory access rules (refer to T.Abuse-Func but for functions available after TOE delivery), (ii) introduce flaws by forcing operational conditions (refer to T.Malfunction) and/or by physical manipulation (refer to T.Phys-Manipulation). This attacker is expected to have a high level potential of attack.

3.3 Organizational Security Policies

The following Figure 6 shows the policies applied in this Security Target.

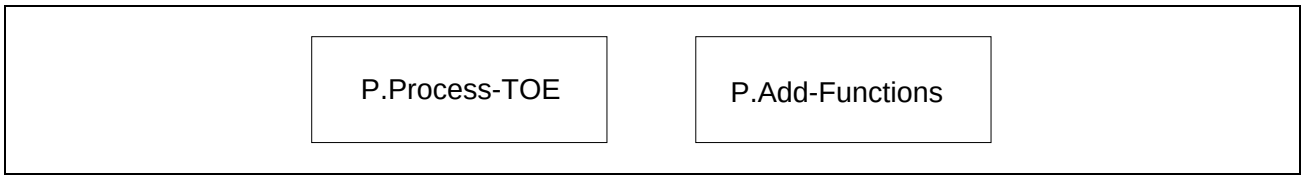


Figure 6: Policies

The IC Developer / Manufacturer must apply the policy "Protection during TOE Development and Production (P.Process-TOE)" as specified below.

P.Process-TOE	Protection during TOE Development and Production An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.

The accurate identification is introduced at the end of the production test in phase 3. Therefore the production environment must support this unique identification.

The information and material produced and/or processed by the TOE Manufacturer in the TOE development and production environment (Phases 2 up to TOE Delivery) can be grouped as follows:

- logical design data,
- physical design data,
- IC Dedicated Software, Security IC Embedded Software, Initialisation Data and Pre-personalisation Data,
- specific development aids,
- test and characterisation related data,
- material for software development support, and
- photomasks and products in any form

as long as they are generated, stored, or processed by the TOE Manufacturer.

The TOE provides specific security functionality which can be used by the Smartcard Embedded Software. In the following specific security functionality is listed which is not derived from threats identified for the TOE's environment because it can only be decided in the context of the smartcard application, against which threats the Smartcard Embedded Software will use the specific security functionality.

The IC Developer / Manufacturer must apply the policy "Additional Specific Security Functionality (P.Add-Functions)" as specified below.

P.Add-Functions	Additional Specific Security Functionality The TOE shall provide the following specific security functionality to the Smartcard Embedded Software: ● Triple Data Encryption Standard (3DES) ● Advanced Encryption Standard (AES)

3.4 Assumptions

The following Figure 6 shows the assumptions applied in this Security Target.

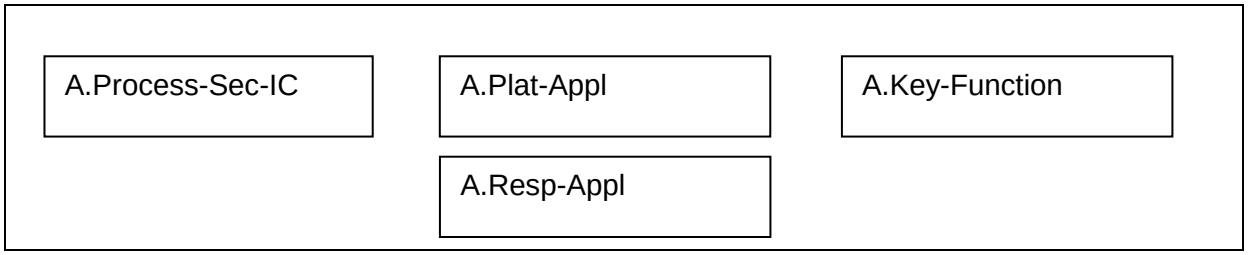


Figure 7: Assumptions

The intended usage of the TOE is twofold, depending on the Life Cycle Phase: (i) The Security IC Embedded Software developer use it as a platform for the Security IC software being developed. The Composite Product Manufacturer (and the consumer) uses it as a part of the Security IC. The Composite Product is used in a terminal which supplies the Security IC (with power and clock) and (at least) mediates the communication with the Security IC Embedded Software.

Before being delivered to the consumer the TOE is packaged. Many attacks require the TOE to be removed from the carrier. Though this extra step adds difficulties for the attacker no specific assumptions are made here regarding the package.

Appropriate "Protection during Packaging, Finishing and Personalisation (A.Process-Sec-IC)" must be ensured after TOE Delivery up to the end of Phase 6, as well as during the delivery to Phase 7 as specified below.

A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that the Phases after TOE Delivery are assumed to be protected appropriately.

The information and material produced and/or processed by the Security IC Embedded Software Developer in Phase 1 and by the Composite Product Manufacturer can be grouped as follows:

- the Security IC Embedded Software including specifications, implementation and related documentation,
- pre-personalisation and personalisation data including specifications of formats and memory areas, test related data,
- the User Data and related documentation, and
- material for software development support

as long as they are not under the control of the TOE Manufacturer. Details must be defined in the Protection Profile or Security Target for the evaluation of the Security IC Embedded Software and/or Security IC.

The developer of the Security IC Embedded Software must ensure the appropriate "Usage of Hardware Platform (A.Plat-Appl)" while developing this software in Phase 1 as specified below.

A.Plat-Appl	Usage of Hardware Platform

The Security IC Embedded Software is designed so that the requirements from the following documents are met: (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the hardware data sheet, and the hardware application notes, and (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report.

Note that particular requirements for the Security IC Embedded Software are often not clear before considering a specific attack scenario during vulnerability analysis of the Security IC (AVA_VAN). A summary of such results is provided in the document "ETR for composite evaluation" (ETR-COMP). This document can be provided for the evaluation of the composite product. The ETR-COMP may also include guidance for additional tests being required for the combination of hardware and software. The TOE evaluation must be completed before evaluation of the Security IC Embedded Software can be completed. The TOE evaluation can be conducted before and independent from the evaluation of the Security IC Embedded Software.

The developer of the Security IC Embedded Software must ensure the appropriate “Treatment of User Data (A.Resp-Appl)” while developing this software in Phase 1 as specified below.

A.Resp-Appl	Treatment of User Data All User Data are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.

The application context specifies how the User Data shall be handled and protected. The evaluation of the Security IC according to this Security Target is conducted on generalized application context. The concrete requirements for the Security IC Embedded Software shall be defined in the Protection Profile respective Security Target for the Security IC Embedded Software. The Security IC cannot prevent any compromise or modification of User Data by malicious Security IC Embedded Software. The assumption A.Resp-Appl ensures that the Security IC Embedded Software follows the security rules of the application context.

The developer of the Smartcard Embedded Software must ensure the appropriate “Usage of Key-dependent Functions (A.Key-Function)” while developing this software in Phase 1 as specified below.

A.Key-Function	Usage of Key-dependent Functions Key-dependent functions (if any) shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced).

Note that here the routines which may compromise keys when being executed are part of the Smartcard Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines which are part of the TOE and (ii) the processing of User Data including cryptographic keys.

4 SECURITY OBJECTIVES

This chapter Security Objectives contains the following sections:

4.1 Security Objectives for the TOE

4.2 Security Objectives for the IC Embedded Software development Environment

4.3 Security Objectives for the operational Environment

4.4 Security Objectives Rationale

4.1 Security Objectives for the TOE

According to the Protection Profile[BSI-PP-0035] there are the following standard high-level security goals:

SG1 maintain the integrity of User Data and of the Security IC Embedded Software (when being executed/processed and when being stored in the TOE's memories) as well as

SG2 maintains the confidentiality of User Data and of the Security IC Embedded Software (when being processed and when being stored in the TOE's memories).

The Security IC may not distinguish between User Data which are public known or kept confidential. Therefore the security IC shall protect the confidentiality and integrity of the User Data, unless the Security IC Embedded Software chooses to disclose or modify it.

In particular integrity of the Security IC Embedded Software means that it is correctly being executed which includes the correct operation of the TOE's functionality. Though the Security IC Embedded Software (normally stored in the ROM) will in many cases not contain secret data or algorithms, it must be protected from being disclosed, since for instance knowledge of specific implementation details may assist an attacker.

SG3 maintains the correct operation of the security services provided by the TOE for the Security IC Embedded Software.

SG4 provide random numbers.

These standard high-level security goals are refined below by defining security objectives as required by the *Common Criteria* (refer to Figure 8). Note that the integrity of the TOE is a mean to reach these objectives.

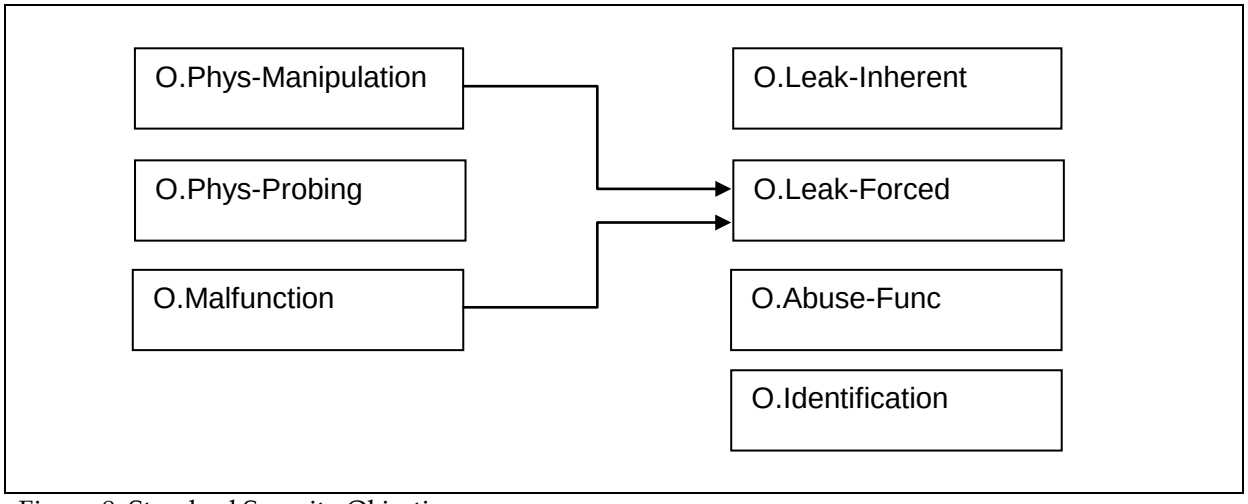


Figure 8: Standard Security Objectives

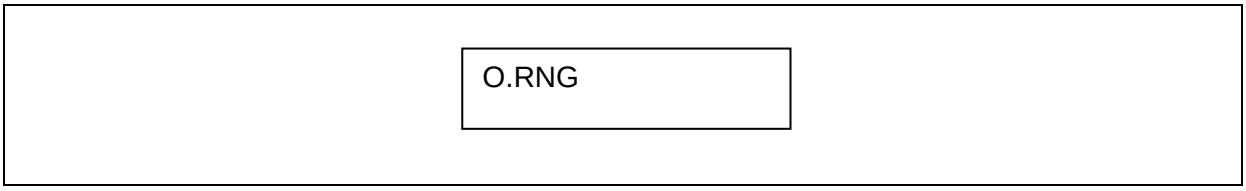


Figure 9: Security Objectives related to Specific Functionality

Standard Security Objectives

The TOE shall provide "Protection against Inherent Information Leakage (O.Leak-Inherent)" as specified below.

O.Leak-Inherent Protection against Inherent Information Leakage

The TOE must provide protection against disclosure of confidential data (User Data or TSF data) stored and/or processed in the Smartcard IC

- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and
- by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines).

This objective pertains to measurements with subsequent complex signal processing whereas O.Phys-Probing is about direct measurements on elements on the chip surface. Details correspond to an analysis of attack scenarios which is not given here.

The TOE shall provide "Protection against Physical Probing (O.Phys-Probing)" as specified below.

O.Phys-Probing Protection against Physical Probing

The TOE must provide protection against disclosure of User Data, against the disclosure/reconstruction of the Smartcard Embedded Software or against the disclosure of other critical operational information. This includes protection against

- measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or
- measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis)

with a prior

- reverse-engineering to understand the design and its properties and functions.

The TOE must be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

The TOE shall provide "Protection against Malfunctions (O.Malfunction)" as specified below.

O.Malfunction Protection against Malfunctions

The TOE must ensure its correct operation.

The TOE must prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent errors. The environmental conditions may include voltage, clock frequency, temperature, or external energy fields.

Remark: A malfunction of the TOE may also be caused using a direct interaction with elements on the chip surface. This is considered as being a manipulation (refer to the objective O.Phys-Manipulation) provided that detailed knowledge about the TOE´s internal construction is required and the attack is performed in a controlled manner.

The TOE shall provide "Protection against Physical Manipulation (O.Phys-Manipulation)" as specified below.

O.Phys-Manipulation Protection against Physical Manipulation

The TOE must provide protection against manipulation of the TOE (including its software and TSF data), the Smartcard Embedded Software and the User Data. This includes protection against

- reverse-engineering (understanding the design and its properties and functions),
- manipulation of the hardware and any data, as well as
- controlled manipulation of memory contents (User Data).

The TOE must be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

The TOE shall provide "Protection against Forced Information Leakage (O.Leak-Forced)" as specified below:

O.Leak-Forced Protection against Forced Information Leakage

The Security IC must be protected against disclosure of confidential data processed in the Security IC (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker

- by forcing a malfunction (refer to "Protection against Malfunction due to Environmental Stress (O.Malfunction)" and/or
- by a physical manipulation (refer to "Protection against Physical Manipulation (O.Phys-Manipulation)".

If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.

The TOE shall provide "Protection against Abuse of Functionality (O.Abuse-Func)" as specified below.

O.Abuse-Func Protection against Abuse of Functionality

The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order (i) to disclose critical User Data, (ii) to manipulate critical User Data of the Smartcard Embedded Software, (iii) to manipulate Soft-coded Smartcard Embedded Software or (iv) bypass, deactivate, change or explore security features or functions of the TOE. Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Test Software which are not specified here.

The TOE shall provide "TOE Identification (O.Identification)" as specified below:

O.Identification TOE Identification

The TOE must provide means to store Initialisation Data and Pre-personalisation Data in its non-volatile memory. The Initialisation Data (or parts of them) are used for TOE identification.

Security Objectives related to Specific Functionality (referring to SC4)

The TOE shall provide "Random Numbers (O.RND)" as specified below.

O.RND Random Numbers

The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have sufficient entropy.

The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

Security Objectives for Added Function

The TOE shall provide "Additional Specific Security Functionality (O.Add-Functions)" as specified below.

O.Add-Functions Additional Specific Security Functionality

The TOE must provide the following specific security functionality to the Smartcard Embedded Software:

- Triple Data Encryption Standard (3DES)
- Advanced Encryption Standard (AES)

The TOE shall provide "Area based Memory Access Control (O.Mem-Access)" as specified below.

O.Mem-Access	Area based Memory Access Control
	The TOE must provide the Smartcard Embedded Software with the capability to define restricted access memory areas. The TOE must then enforce the partitioning of such memory areas so that access of software to memory areas is controlled as required, for example, in a multi-application environment.

4.2 Security Objectives for the Security IC Embedded Software Development Environment

The development of the Security IC Embedded Software is outside the development and manufacturing of the TOE. The Security IC Embedded Software defines the operational use of the TOE. This section describes the security objectives for the operational environment enforced by the Security IC Embedded software.

Phase 1

The Security IC Embedded Software shall provide "Usage of Hardware Platform (OE.Plat-Appl)" as specified below.

OE.Plat-Appl	Usage of Hardware Platform
	To ensure that the TOE is used in a secure manner the Security IC Embedded Software shall be designed so that the requirements from the following documents are met: (i) hardware data sheet for the TOE, (ii) data sheet of the IC Dedicated Software of the TOE, (iii) TOE application notes, other guidance documents, and (iv) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report.

The Security IC Embedded Software shall provide "Treatment of User Data (OE.Resp-Appl)" as specified below.

OE.Resp-Appl	Treatment of User Data
	Security relevant User Data (especially cryptographic keys) are treated by the Smartcard Embedded Software as required by the security needs of the specific application context. For example the Smartcard Embedded Software will not disclose security relevant user data to unauthorised users or processes when communicating with a terminal.

4.2.1 Clarification of "Usage of Hardware Platform (OE.Plat-Appl)"

Regarding the cryptographic services this objective of the environment has to be clarified. The TOE supports cipher schemes as additional specific security functionality. If required the Smartcard Embedded Software shall use these cryptographic services of the TOE and their interface as specified. When key-dependent functions implemented in the Smartcard Embedded Software are just being executed, the Smartcard Embedded Software must provide protection against disclosure of confidential data (User Data) stored and/or processed in the TOE by using the methods described under "Inherent Information Leakage (T.Leak-Inherent)" and "Forced Information Leakage (T.Leak-Forced)".

Regarding the area based access control this objective of the environment has to be clarified. For the separation of different applications the Smartcard Embedded Software (Operating System) may implement a memory management scheme based upon security mechanisms of the TOE.

For the separation of different applications the Smartcard Embedded Software may implement a memory management scheme based upon security mechanisms of the TOE as required by the security policy defined for the specific application context.

4.2.2 Clarification of "Treatment of User Data (OE.Resp-Appl)"

Regarding the cryptographic services this objective of the environment has to be clarified. By definition cipher or plain text data and cryptographic keys are User Data. The Smartcard Embedded Software shall treat these data appropriately, use only proper secret keys (chosen from a large key space) as input for the cryptographic function of the TOE and use keys and functions appropriately in order to ensure the strength of cryptographic operation.

This means that keys are treated as confidential as soon as they are generated. The keys must be unique with a very high probability, as well as cryptographically strong. If keys are imported into the TOE and/or derived from other keys, quality and confidentiality must be maintained. This implies that appropriate key management has to be realised in the environment.

Regarding the area based access control this objective of the environment has to be clarified. The treatment of User Data is also required when a multi-application operating system is implemented as part of the Smartcard Embedded Software on the TOE. In this case the multi-application operating system should not disclose security relevant user data of one application to another application when it is processed or stored on the TOE.

The treatment of User Data is still required when a multi-application operating system is implemented as part of the Smartcard Embedded Software on the TOE. In this case the multi-application operating system should not disclose security relevant user data of one application to another application when it is processed or stored on the TOE.

4.3 Security Objectives for the Operational Environment

TOE Delivery up to the End of Phase 6

Appropriate "Protection during Packaging, Finishing and Personalisation (OE.Process-Sec-IC)" must be ensured after TOE Delivery up to the end of Phases 6, as well as during the delivery to Phase 7 as specified below.

OE.Process-Sec-IC	Protection during composite product manufacturing Security procedures shall be used after TOE Delivery up to delivery to the "consumer" to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that Phases after TOE Delivery up to the end of Phase 6 must be protected appropriately.

4.3.1 Clarification of "Protection during Composite Product Manufacturing (OE.Process-Sec-IC)"

The protection during packaging, finishing and personalization includes also the personalization process and the personalization data during Phase 4, Phase 5 and Phase 6.

Since OE.Process-Sec-IC requires the Composite Product Manufacturer to implement those measures assumed in A.Process-Sec-IC, the assumption is covered by this objective.

4.4 Security Objectives Rationale

Table 4 below gives an overview, how the assumptions, threats, and organisational security policies are addressed by the objectives. The text following after the table justifies this in detail.

Assumption, Threat or Organisational Security Policy	Security Objective	Notes
A.Plat-Appl	OE.Plat-Appl	Phase 1
A.Resp-Appl	OE.Resp-Appl	Phase 1
P.Process-TOE	O.Identification	Phase 2 – 3 optional Phase 4
A.Process-Sec-IC	OE.Process-Sec-IC	Phase 5 – 6 optional Phase 4
T.Leak-Inherent	O.Leak-Inherent	
T.Phys-Probing	O.Phys-Probing	
T.Malfunction	O.Malfunction	
T.Phys-Manipulation	O.Phys-Manipulation	
T.Leak-Forced	O.Leak-Forced	
T.Abuse-Func	O.Abuse-Func	
T.RND	O.RND	
T.Mem-Access	O.Mem-Access	
P.Add-Functions	O.Add-Functions	
A.Key-Function	OE.Plat-Appl OE.Resp-Appl	

Table 4: Security Objectives versus Assumptions, Threats or Policies

The justification related to the assumption "Usage of Hardware Platform (A.Plat-Appl)" is as follows:

Since OE.Plat-Appl requires the Smartcard Embedded Software developer to implement those measures assumed in A.Plat-Appl, the assumption is covered by the objective.

The justification related to the assumption "Treatment of User Data (A.Resp-Appl)" is as follows:

Since OE.Resp-Appl requires the developer of the Smartcard Embedded Software to implement measures as assumed in A.Resp-Appl, the assumption is covered by the objective.

The justification related to the organisational security policy "Protection during TOE Development and Production (P.Process-TOE)" is as follows:

O.Identification requires that the TOE has to support the possibility of a unique identification. The unique identification can be stored on the TOE. Since the unique identification is generated by the production environment the production environment must support the integrity of the generated unique identification. The technical and organisational security measures that ensure the security of the development environment and production environment are evaluated based on the assurance measures that are part of the evaluation. For a list of material produced and processed by the TOE Manufacturer refer to paragraph 44. All listed items and the associated development and production environments are subject of the evaluation. Therefore, the organisational security policy P.Process-TOE is covered by this objective, as far as organisational measures are concerned.

The justification related to the assumption "Protection during Packaging, Finishing and Personalisation (A.Process-Sec-IC)" is as follows:

Since OE.Process-Sec-IC requires the Composite Product Manufacturer to implement those measures assumed in A.Process-Sec-IC, the assumption is covered by this objective.

The justification related to the threats "Inherent Information Leakage (T.Leak-Inherent)", "Physical Probing (T.Phys-Probing)", "Malfunction due to Environmental Stress (T.Malfunction)", "Physical Manipulation (T.Phys-Manipulation)", "Forced Information Leakage (T.Leak-Forced)", "Abuse of Functionality (T.Abuse-Func)" and "Deficiency of Random Numbers (T.RND)" is as follows:

For all threats the corresponding objectives are stated in a way, which directly corresponds to the description of the threat. It is clear from the description of each objective, that the corresponding threat is removed if the objective is valid. More specifically, in every case the ability to use the attack method successfully is countered, if the objective holds.

The justification related to the threat "Memory Access Violation (T.Mem-Access)" is as follows:

According to O.Mem-Access the TOE must enforce the partitioning of memory areas so that access of software to memory areas is controlled. Any restrictions are to be defined by the Smartcard Embedded Software. Thereby security violations caused by accidental or deliberate access to restricted data (which may include code) can be prevented (refer to T.Mem-Access). The threat T.Mem-Access is therefore removed if the objective is met.

The clarification of "Usage of Hardware Platform (OE.Plat-Appl)" makes clear that it is up to the Smartcard Embedded Software to implement the memory management scheme by appropriately administrating the TSF. This is also expressed both in T.Mem-Access and O.Mem-Access. The TOE shall provide access control functions as a means to be used by the Smartcard Embedded Software. This is further emphasised by the clarification of "Treatment of User Data (OE.Resp-Appl)" which reminds that the Smartcard Embedded Software must not undermine the restrictions it defines. Therefore, the clarifications contribute to the coverage of the threat T.Mem-Access.

The justification related to the security objective "Additional Specific Security Functionality (O.Add-Functions)" is as follows:

Since O.Add-Functions requires the TOE to implement exactly the same specific security functionality as required by P.Add-Functions, the organisational security policy is covered by the objective.

Nevertheless the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced define how to implement the specific security functionality required by P.Add-Functions. (Note that these objectives support that the specific security functionality is provided in a secure way as expected from P.Add-Functions.) Especially O.Leak-Inherent and O.Leak-Forced refer to the protection of confidential data (User Data or TSF data) in general. User Data are also processed by the specific security functionality required by P.Add-Functions.

Compared to Smartcard IC Platform Protection Profile a clarification has been made for the security objective "Usage of Hardware Platform (OE.Plat-Appl)": If required the Smartcard Embedded Software shall use these cryptographic services of the TOE and their interface as specified. In addition, the Smartcard Embedded Software must implement functions which perform operations on keys (if any) in such a manner that they do not disclose information about confidential data. The non disclosure due to leakage A.Key-Function attacks is included in this objective OE.Plat-Appl. This addition ensures that the assumption A.Plat-Appl is still covered by the objective OE.Plat-Appl although additional functions are being supported according to O.Add-Functions.

Compared to Smartcard IC Platform Protection Profile a clarification has been made for the security objective "Treatment of User Data (OE.Resp-Appl)": By definition cipher or plain text data and cryptographic keys are User Data. So, the Smartcard Embedded Software will protect such data if required and use keys and functions appropriately in order to ensure the strength of cryptographic operation. Quality and confidentiality must be maintained for keys that are imported and/or derived from other keys. This implies that appropriate key management has to be realised in the environment. That is expressed by the assumption A.Key–Function which is covered from OE.Resp–Appl. These measures make sure that the assumption A.Resp-Appl is still covered by the security objective OE.Resp-Appl although additional functions are being supported according to P.Add-Functions.

The justification of the additional policy and the additional assumption show that they do not contradict to the rationale already given in the Protection Profile for the assumptions, policy and threats defined there.

5 EXTENDED COMPONENTS DEFINITION

This chapter 5 Extended Components Definition contains the following sections:

5.1 Definition of the family FCS_RNG

5.2 Definition of the Family FMT_LIM

5.3 Definition of the Family FAU_SAS

5.1 Definition of the Family FCS_RNG

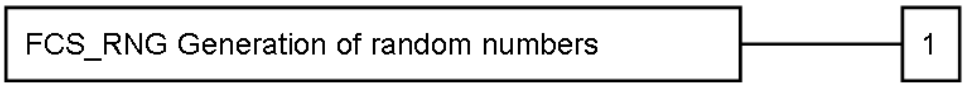
To define the IT security functional requirements of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

FCS_RNG 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_RNG.1	Generation of random numbers requires that random numbers meet a defined quality metric.
Management:	FCS_RNG.1 There are no management activities foreseen.
Audit:	FCS_RNG.1 There are no actions defined to be auditable.
FCS_RNG.1	Random number generation
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	The TSF shall provide a [selection: physical, non-physical true, deterministic, hybrid] random number generator that implements: [assignment: list of security capabilities].
FCS_RNG.1.2	The TSF shall provide random numbers that meet [assignment: a defined quality metric].

5.2 Definition of the Family FMT_LIM

To define the IT security functional requirements of the TOE an additional family (FMT_LIM) of the Class FMT (Security Management) is defined here. This family describes the functional requirements for the Test Features of the TOE. The new functional requirements were defined in the class FMT because this class addresses the management of functions of the TSF. The examples of the technical mechanism used in the TOE appropriate to address the specific issues of preventing the abuse of functions by limiting the capabilities of the functions and by limiting their availability.

The family "Limited capabilities and availability (FMT_LIM)" is specified as follows.

FMT_LIM Limited capabilities and availability

Family behaviour

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

Component levelling:

FMT_LIM.1	Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.
FMT_LIM.2	Limited availability requires that the TSF restrict the use of functions (refer to Limited capabilities (FMT_LIM.1)). This can be achieved, for instance, by removing or by disabling functions in a specific phase of the TOE's life-cycle.
Management:	FMT_LIM.1, FMT_LIM.2 There are no management activities foreseen.
Audit:	FMT_LIM.1, FMT_LIM.2 There are no actions defined to be auditable.

The TOE Functional Requirement "Limited capabilities (FMT_LIM.1)" is specified as follows.

FMT_LIM.1	Limited capabilities
Hierarchical to:	No other components.
FMT_LIM.1.1	The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced [assignment: Limited capability and availability policy].

Dependencies: FMT_LIM.2 Limited availability.

The TOE Functional Requirement "Limited availability (FMT_LIM.2)" is specified as follows.

FMT_LIM.2 Limited availability

Hierarchical to: No other components.

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

Dependencies: FMT_LIM.1 Limited capabilities.

Application note: The functional requirements FMT_LIM.1 and FMT_LIM.2 assume that there are two types of mechanisms (limited capabilities and limited availability) which together shall provide protection in order to enforce the policy. This also allows that

(i) the TSF is provided without restrictions in the product in its user environment but its capabilities are so limited that the policy is enforced

or conversely

(ii) the TSF is designed with high functionality but is removed or disabled in the product in its user environment.

The combination of both requirements shall enforce the policy.

5.3 Definition of the Family FAU_SAS

To define the security functional requirements of the TOE an additional family (FAU_SAS) of the Class FAU (Security Audit) is defined here. This family describes the functional requirements for the storage of audit data. It has a more general approach than FAU_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

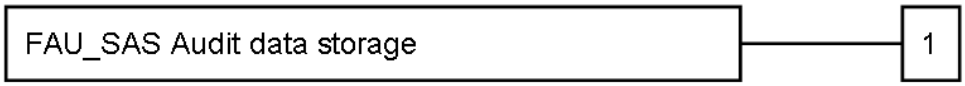
The family "Audit data storage (FAU_SAS)" is specified as follows.

FAU_SAS Audit data storage

Family behaviour

This family defines functional requirements for the storage of audit data.

Component levelling



FAU_SAS.1 Requires the TOE to provide the possibility to store audit data.

Management: FAU_SAS.1

	There are no management activities foreseen.
Audit:	FAU_SAS.1
	There are no actions defined to be auditable.
FAU_SAS.1	Audit storage
Hierarchical to:	No other components.
FAU_SAS.1.1	The TSF shall provide [assignment: list of subjects] with the capability to store [assignment: list of audit information] in the [assignment: type of persistent memory].
Dependencies:	No dependencies.

6 IT SECURITY REQUIREMENTS

This chapter 6 IT Security Requirements contains the following sections:

6.1 Security Functional Requirements for the TOE

6.2 Security Assurance Requirements for the TOE

6.3 Security Requirements Rationale

6.1 Security Functional Requirements for the TOE

In order to define the Security Functional Requirements the Part 2 of the Common Criteria was used. However, some Security Functional Requirements have been refined. The refinements are described below the associated SFR. The operations completed in the ST are marked in italic font.

Malfunctions

The TOE shall meet the requirement "Limited fault tolerance (FRU_FLT.2)" as specified below.

FRU_FLT.2	Limited fault tolerance
Hierarchical to:	FRU_FLT.1
FRU_FLT.2.1	The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: *exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1) .*
Dependencies:	FPT_FLS.1 Failure with preservation of secure state
Refinement:	The term "failure" above means "circumstances". The TOE prevents failures for the "circumstances" defined above.

The TOE shall meet the requirement "Failure with preservation of secure state (FPT_FLS.1)" as specified below.

FPT_FLS.1	Failure with preservation of secure state
Hierarchical to:	No other components.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: *exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur.*
Dependencies:	No dependencies
Refinement:	The term "failure" above also covers "circumstances". The TOE prevents failures for the "circumstances" defined above.
Application note:	The secure state is maintained by TOE's detectors. The TOE's detectors are monitoring the failure occurs. The failures are abnormal frequency, abnormal voltage, abnormal temperature, and power glitch detectors that detect out of the specified range (refer to table 9). If the failures are happen, the TOE goes into RESET state. This satisfies the FPT_FLS.1 "Failure with preservation of secure state."

Abuse of Functionality

The TOE shall meet the requirement "Limited capabilities (FMT_LIM.1)" as specified below (Common Criteria Part 2 extended).

FMT_LIM.1	Limited capabilities
Hierarchical to:	No other components.
FMT_LIM.1.1	The TSF shall be designed in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: *Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.*
Dependencies:	FMT_LIM.2 Limited availability.

The TOE shall meet the requirement "Limited availability (FMT_LIM.2)" as specified below (Common Criteria Part 2 extended).

FMT_LIM.2	Limited availability
Hierarchical to:	No other components.
FMT_LIM.2.1	The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: *Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.*
Dependencies:	FMT_LIM.1 Limited capabilities.

The TOE shall meet the requirement "Audit storage (FAU_SAS.1)" as specified below (Common Criteria Part 2 extended).

FAU_SAS.1	Audit storage
Hierarchical to:	No other components.
FAU_SAS.1.1	The TSF shall provide the test process before TOE Delivery with the capability to store the Initialisation Data and/or Prepersonalisation Data and/or supplements of the Smartcard Embedded Software in *a Test ROM area*.
Dependencies:	No dependencies.
Application Note:	The integrity and uniqueness of the unique identification of the TOE must be supported by the development, production and test environment.

Physical Manipulation and Probing

The TOE shall meet the requirement "Resistance to physical attack (FPT_PHP.3)" as specified below.

FPT_PHP.3	Resistance to physical attack
Hierarchical to:	No other components.

FPT_PHP.3.1	The TSF shall resist *physical manipulation and physical probing* to the *TSF* by responding automatically such that the SFRs are always enforced.
Dependencies:	No dependencies.
Refinement:	The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, "automatic response" means here (i) assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.
Application Note:	This requirement is achieved by security feature as the Active shield must be removed and bypassed in order to perform physical intrusive attacks. The TOE makes a reset or FIQ occurs to stops operation if a physical manipulation or physical probing attack is detected. And also Static Address/Data scrambling for bus and memory & Synthesizable processor core make the reverse-engineering of the TOE layout unpractical. So these functionalities meet the security functional requirement of FPT_PHP.3: Resistance to physical attack.

Leakage

The TOE shall meet the requirement "Basic internal transfer protection (FDP_ITT.1)" as specified below.

FDP_ITT.1	Basic internal transfer protection
Hierarchical to:	No other components.
FDP_ITT.1.1	The TSF shall enforce the *Data Processing Policy* to prevent the *disclosure* of user data when it is transmitted between physically-separated parts of the TOE.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
Refinement:	The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as physically-separated parts of the TOE.

The TOE shall meet the requirement "Basic internal TSF data transfer protection (FPT_ITT.1)" as specified below.

FPT_ITT.1	Basic internal TSF data transfer protection
Hierarchical to:	No other components.
FPT_ITT.1.1	The TSF shall protect TSF data from *disclosure* when it is transmitted between separate parts of the TOE.
Dependencies:	No dependencies.
Refinement:	The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE.

This requirement is equivalent to FDP_ITT.1 above but refers to TSF data instead of User Data. Therefore, it should be understood as to refer to the same *Data Processing Policy* defined under FDP_IFC.1 below.

The TOE shall meet the requirement " Subset information flow control (FDP_IFC.1)"as specified below:

FDP_IFC.1	Subset information flow control
Hierarchical to:	No other components.
FDP_IFC.1.1	The TSF shall enforce the *Data Processing Policy* on *all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software.*
Dependencies:	FDP_IFF.1 Simple security attributes

The following Security Function Policy (SFP) **Data Processing Policy** is defined for the requirement " Subset information flow control (FDP_IFC.1)":

User Data and TSF data shall not be accessible from the TOE except when the Security IC Embedded Software decides to communicate the User Data via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the Security IC Embedded Software.

Random Numbers (DTRNG)

The TOE shall meet the requirement "Quality metric for random numbers (FCS_RNG.1)" as specified below (Common Criteria Part 2 extended).

FCS_RNG.1	Random number generation
Hierarchical to:	No other components.
FCS_RNG.1.1	The TSF shall provide *a physical* random number generator that implements *total failure test of the random source*.
FCS_RNG.1.2	The TSF shall provide random numbers *together with a post processing described in DTRNG application note and DTRNG library that meet the "standard" level of ANSSI requirements (French metric).* In additions, The TSF shall provide random numbers that meet *AIS 31 version 1 Functional Classes and Evaluation Methodology for Physical Random Number Generators, 25 September 2001, Class P2.*
Dependencies:	No dependencies.
Application Note:	The DTRNG FRO library comprises some functions that performs statistical test on the DTRNG output in order to ensure that the DTRNG is working properly. If test is fails the function return an error value and the DTRNG is shuttled down. Those functions are described in DTRNG FRO Application note.

Memory Access Control

Usage of multiple applications in one Smartcard often requires separating code and data in order to prevent that one application can access code and/or data of another application. To support the TOE provides Area based Memory Access Control.

The security service being provided is described in the Security Function Policy (SFP) **Memory Access Control Policy**. The security functional requirement **"Subset access control (FDP_ACC.1)"** requires

that this policy is in place and defines the scope were it applies. The security functional requirement **"Security attribute based access control (FDP_ACF.1)"** defines addresses security attribute usage and characteristics of policies. It describes the rules for the function that implements the Security Function Policy (SFP) as identified in FDP_ACC.1. The decision whether an access is permitted or not is taken based upon attributes allocated to the software. The user software defines the attributes and memory areas. The corresponding permission control information is evaluated "on-the-fly" by the hardware so that access is granted/effective or denied/inoperable.

The security functional requirement **"Static attribute initialization (FMT_MSA.3)"** ensures that the default values of security attributes are appropriately either permissive or restrictive in nature. Alternative values can be specified by any subject provided that the **Memory Access Control Policy** allows that. This is described by the security functional requirement **"Management of security attributes (FMT_MSA.1)"**. The attributes are determined during TOE manufacturing (FMT_MSA.3) or set at run-time (FMT_MSA.1).

From TOE´s point of view the different roles in the user software can be distinguished according to the memory based access control. However the definition of the roles belongs to the user software.

The following Security Function Policy (SFP) **Memory Access Control Policy** is defined for the requirement "Security attribute based access control (FDP_ACF.1)":

Memory Access Control Policy

The TOE shall control *read, write, delete, and execute accesses of software running at between two different modes (privilege and user mode)* on *data including code stored in memory areas.*

The TOE shall restrict the ability to define, to change or at least to finally accept the applied rules (as mentioned in FDP_ACF.1) to *software with privilege mode).*

The TOE shall meet the requirement "Subset access control (FDP_ACC.1)" as specified below.

FDP_ACC.1	Subset access control
Hierarchical to:	No other components.
FDP_ACC.1.1	The TSF shall enforce the *Memory Access Control Policy* on *all subjects (software with privilege mode and user mode), all objects (data including code stored in memories) and all the operations defined in the Memory Access Control Policy.* Subjects are software codes in Privilege and User mode. Objects are data stored in ROM, RAM and FLASH memories.
Dependencies:	FDP_ACF.1 Security attribute based access control

The TOE shall meet the requirement "Security attribute based access control (FDP_ACF.1)" as specified below.

FDP_ACF.1	Security attribute based access control The attributes are all the operations related to the data stored in memories, which are the *read, write, delete* and *execute* operations.
Hierarchical to:	No other components.
FDP_ACF.1.1	The TSF shall enforce the *Memory Access Control Policy* to objects based on the *memory area where the software is executed from and/or the memory area where* the *access is performed to and/or the operation to be performed.*

FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *evaluate the corresponding permission control information before the access so that accesses to be denied cannot be utilised by the subject attempting to perform the operation*.
FDP_ACF.1.3	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *none*.
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: none.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation

The TOE shall meet the requirement "Static attribute initialisation (FMT_MSA.3)" as specified below.

FMT_MSA.3	Static attribute initialisation
Hierarchical to:	No other components.
FMT_MSA.3.1	The TSF shall enforce the *Memory Access Control Policy* to provide *well defined* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow *any subject (provided that the Memory Access Control Policy is enforced and the necessary access is therefore allowed)* 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

The TOE shall meet the requirement "Management of security attributes (FMT_MSA.1)" as specified below:

FMT_MSA.1	Management of security attributes
Hierarchical to:	No other components.
FMT_MSA.1.1	The TSF shall enforce the *Memory Access Control Policy* to restrict the ability to *change default, modify or delete* the security attributes *permission control information* to *running at privilege mode*.
Dependencies:	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

The TOE shall meet the requirement "Specification of management functions (FMT_SMF.1)" as specified below:

FMT_SMF.1	Specification of management functions
Hierarchical to:	No other components
FMT_SMF.1.1	The TSF shall be capable of performing the following security management functions*: access the control registers of the MPU*.
Dependencies:	No dependencies

Cryptographic Support

FCS_COP.1 Cryptographic operation requires, a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard.

The following additional specific security functionality is implemented in the TOE:

- Triple Data Encryption Standard (3DES) with 112bit or 168bit key size
- Advanced Encryption Standard (AES) with 128 bit, 192bit and 256bit key size

Triple-DES Operation

The Triple DES (3DES) operation of the TOE shall meet the requirement "Cryptographic operation (FCS_COP.1)" as specified below.

FCS_COP.1/3DES	Cryptographic operation
Hierarchical to:	No other components.
FCS_COP.1.1/3DES	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Triple Data Encryption Standard (3DES) - ECB mode* and cryptographic key sizes *112 bit or 168 bit key size* that meet the following standards: *[FIPS SP800-67], chapter 2 and 3. TOE implements 3DES with key option 1 and 2 with ECB mode.*
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

AES Operation

The AES operation of the TOE shall meet the requirement "Cryptographic operation (FCS_COP.1)" as specified below.

FCS_COP.1/AES	Cryptographic operation
Hierarchical to:	No other components.
FCS_COP.1.1/AES	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Advanced Encryption Standard (AES) - ECB mode* and cryptographic key sizes *128bit, 192bit or 256bit key size* that meet the following standard: *[FIPS197], chapter 5.*
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

Summary of Security Functional Requirements

Security Functional Requirements
Limited fault tolerance (FRU_FLT.2)
Failure with preservation of secure state (FPT_FLS.1)

Audit storage (FAU_SAS.1)
Limited capabilities(FMT_LIM.1)
Limited availability (FMT_LIM.2)
Resistance to physical attack (FPT_PHP.3)
Basic internal transfer protection (FDP_ITT.1)
Basic internal TSF data transfer protection (FPT_ITT.1)
Subset information flow control (FDP_IFC.1)
Quality metric for random numbers (FCS_RNG.1)

Table 5. Security Functional Requirements defined in Smart Card IC Protection Profile

Security Functional Requirements
Subset access control (FDP_ACC.1)
Security attribute based access control (FDP_ACF.1)
Static attribute initialization (FMT_MSA.3)
Management of security attributes (FMT_MSA.1)
Specification of management functions (FMT_SMF.1)
Cryptographic operation (FCS_COP.1/3DES)
Cryptographic operation (FCS_COP.1/AES)

Table 6. Augmented Security Functional Requirements

6.2 TOE Assurance Requirements

The Security Target will be evaluated according to

Security Target evaluation (Class ASE)

The TOE Assurance Requirements for the evaluation of the TOE and its development and operating environment are those taken from the

Evaluation Assurance Level 5 (EAL5)

and augmented by the following components

ALC_DVS.2 and AVA_VAN.5

corresponding to level "***EAL5+***".

All refinements from *Protection Profile BSI-PP-0035 version 1.0* for the assurance requirements (ALC_DEL, ALC_DVS, ALC_CMS, ALC_CMC, ADV_ARV, ADV_FSP, ADV_IMP, ATE_COV, AGD_OPE, AGD_PRE and ADV_VAN) have to be taken into consideration. *In particular the document [11] is used in the context of vulnerability analysis*

Class ADV: Development

Architectural design	(ADV_ARC.1)
Functional Specification	(ADV_FSP.5)
Implementation Representation	(ADV_IMP.1)
TSF Internals	(ADV_INT.2)
TOE Design	(ADV_TDS.4)

Class AGD: Guidance documents activities

Operational User Guidance	(AGD_OPE.1)
Preparative procedures	(AGD_PRE.1)

Class ALC: Life-cycle support

CM Capabilities	(ALC_CMC.4)
CM Scope	(ALC_CMS.5)
Delivery	(ALC_DEL.1)
Development Security	***(ALC_DVS.2)***
Life Cycle Definition	(ALC_LCD.1)
Tools and Techniques	(ALC_TAT.2)

Class ASE: Security Target evaluation

Conformance claims	(ASE_CCL.1)
Extended components definition	(ASE_ECD.1)
ST introduction	(ASE_INT.1)
Security objectives	(ASE_OBJ.2)
Derived security requirements	(ASE_REQ.2)
Security problem definition	(ASE_SPD.1)
TOE summary specification	(ASE_TSS.1)

Class ATE: Tests

Coverage	(ATE_COV.2)
Depth	(ATE_DPT.3)
Functional Tests	(ATE_FUN.1)
Independent Testing	(ATE_IND.2)

Class AVA: Vulnerability assessment

Vulnerability Analysis	***(AVA_VAN.5)***

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

Table 7 below gives an overview, how the security functional requirements are combined to meet the security objectives. The detailed justification follows after the table.

Objective	TOE Security Functional and Assurance Requirements
O.Leak-Inherent	- FDP_ITT.1 "Basic internal transfer protection" - FPT_ITT.1 "Basic internal TSF data transfer protection" - FDP_IFC.1 "Subset information flow control" - AVA_VAN.5 "Advanced methodical vulnerability analysis"
O.Phys-Probing	- FPT_PHP.3 "Resistance to physical attack"
O.Malfunction	- FRU_FLT.2 "Limited fault tolerance - FPT_FLS.1 "Failure with preservation of secure state" - ADV_ARC.1 "Architectural Design with domain separation and non-bypassability"
O.Phys-Manipulation	- FPT_PHP.3 "Resistance to physical attack"

Objective	TOE Security Functional and Assurance Requirements
O.Leak-Forced	All requirements listed for O.Leak-Inherent - FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, AVA_VAN.5 plus those listed for O.Malfunction and O.Phys-Manipulation - FRU_FLT.2, FPT_FLS.1, FPT_PHP.3, ADV_ARC.1
O.Abuse-Func	- FMT_LIM.1 "Limited capabilities" - FMT_LIM.2 "Limited availability" plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced - FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1, ADV_ARC.1
O.Identification	- FAU_SAS.1 "Audit storage"
O.RND	- FCS_RNG.1 "Quality metric for random numbers" plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced - FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1, AVA_VAN.5, ADV_ARC.1
OE.Plat-Appl	not applicable
OE.Resp-Appl	not applicable
OE.Process-Sec-IC	not applicable
O.Mem-Access	- FDP_ACC.1 "Subset access control" - FDP_ACF.1 "Security attribute based access control" - FMT_MSA.3 "Static attribute initialisation" - FMT_MSA.1 "Management of security attributes" - FMT_SMF.1 "Specification of Management Functions"

Table 7: Security Requirements versus Security Objectives

The justification related to the security objective "Protection against Inherent Information Leakage (O.Leak-Inherent)" is as follows:

The refinements of the security functional requirements FPT_ITT.1 and FDP_ITT.1 together with the policy statement in FDP_IFC.1 explicitly require the prevention of disclosure of secret data (TSF data as well as User Data) when transmitted between separate parts of the TOE or while being processed. This includes that attackers cannot reveal such data by measurements of emanations, power consumption or other behavior of the TOE while data are transmitted between or processed by TOE parts.

Of course this has also to be supported by the Security IC Embedded Software. For example timing attacks were possible if the processing time of algorithms implemented in the software would depend on the content of secret variables.

The justification related to the security objective "Protection against Physical Probing (O.Phys-Probing)" is as follows:

The scenario of physical probing as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective.

It is possible that the TOE needs additional support by the Security IC Embedded Software (e. g. to send data over certain buses only with appropriate precautions). In this case the combination of the Security IC Embedded Software together with FPT_PHP.3 is suitable to meet the objective.

The justification related to the security objective "Protection against Malfunctions (O.Malfunction)" is as follows:

The definition of this objective shows that it covers a situation, where malfunction of the TOE might be caused by the operating conditions of the TOE (while direct manipulation of the TOE is covered O.Phys-Manipulation). There are two possibilities in this situation: Either the operating conditions are inside the tolerated range or at least one of them is outside of this range. The second case is covered by FPT_FLS.1, because it states that a secure state is preserved in this case. The first case is covered by FRU_FLT.2 because it states that the TOE operates correctly under normal (tolerated) conditions. To support this, the functions implementing FRU_FLT.2 and FPT_FLS.1 must work independently so that their operation cannot affect by the Security IC Embedded Software (refer to the refinement). Therefore, there is no possible instance of conditions under O.Malfunction, which is not covered. The suitability of the implementation is subject of the evaluation of the assurance component ADV_ARC.1

The justification related to the security objective "Protection against Physical Manipulation (O.Phys-Manipulation)" is as follows:

The scenario of physical manipulation as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective.

It is possible that the TOE needs additional support by the Embedded Software (for instance by implementing FDP_SDI.1 to check data integrity with the help of appropriate checksums). This support must be addressed in the Guidance Documentation. Together with this FPT_PHP.3 is suitable to meet the objective.

The justification related to the security objective "Protection against Forced Information Leakage (O.Leak-Forced)" is as follows:

This objective is directed against attacks, where an attacker wants to force an information leakage, which would not occur under normal conditions. In order to achieve this the attacker has to combine a first attack step, which modifies the behaviour of the TOE (either by exposing it to extreme operating conditions or by directly manipulating it) with a second attack step measuring and analysing some output produced by the TOE. The first step is prevented by the same measures which support O.Malfunction and O.Phys-Manipulation, respectively. The requirements covering O.Leak-Inherent also support O.Leak-Forced because they prevent the attacker from being successful if he tries the second step directly.

The justification related to the security objective "Protection against Abuse of Functionality (O.Abuse-Func)" is as follows:

This objective states that abuse of functions (especially provided by the IC Dedicated Test Software, for instance in order to read secret data) must not be possible in Phase 7 of the life-cycle. There are two possibilities to achieve this: (i) They cannot be used by an attacker (i. e. its availability is limited) or (ii) using them would not be of relevant use for an attacker (i. e. its capabilities are limited) since the functions are designed in a specific way. The first possibility is specified by FMT_LIM.2 and the second one by FMT_LIM.1. Since these requirements are combined to support the policy, which is suitable to fulfil O.Abuse-Func, both security functional requirements together are suitable to meet the objective.

Other security functional requirements which prevent attackers from circumventing the functions implementing these two security functional requirements (for instance by manipulating the hardware) also support the objective. The relevant objectives are also listed in Table 7.

It was chosen to define FMT_LIM.1 and FMT_LIM.2 explicitly (not using Part 2 of the Common Criteria) for the following reason: Though taking components from the Common Criteria catalogue makes it easier to recognise functions, any selection from Part 2 of the Common Criteria would have made it harder for the reader to understand the special situation meant here. As a consequence, the statement of explicit security functional requirements was chosen to provide more clarity.

The justification related to the security objective "TOE Identification (O.Identification)" is as follows:

Obviously the operations for FAU_SAS.1 are chosen in a way that they require the TOE to provide the functionality needed for O.Identification. The Initialisation Data (or parts of them) are used for TOE identification. The technical capability of the TOE to store Initialisation Data and/or Pre-personalisation Data is provided according to FAU_SAS.1.

It was chosen to define FAU_SAS.1 explicitly (not using a given security functional requirement from Part 2 of the Common Criteria) for the following reason: The security functional requirement FAU_GEN.1 in Part 2 of the CC requires the TOE to generate the audit data and gives details on the content of the audit records (for instance data and time). The possibility to use the functions in order to store security relevant data which are generated outside of the TOE, is not covered by the family FAU_GEN or by other families in Part 2. Moreover, the TOE cannot add time information to the records, because it has no real time clock. Therefore, the new family FAU_SAS was defined for this situation.

The objective must be supported by organisational and other measures, which the TOE Manufacturer has to implement. These measures are a subset of those measures, which are examined during the evaluation of the assurance requirements of the classes AGD, ALC and ADO.

The justification related to the security objective "Random Numbers (O.RND)" is as follows:

Other security functional requirements, which prevent physical manipulation and malfunction of the TOE (see the corresponding objectives listed in the table), support this objective because they prevent attackers from manipulating or otherwise affecting the random number generator.

Random numbers are often used by the Security IC Embedded Software to generate cryptographic keys for internal use. Therefore, the TOE must prevent the unauthorised disclosure of random numbers. Other security functional requirements which prevent inherent leakage attacks, probing and forced leakage attacks ensure the confidentiality of the random numbers provided by the TOE.

Depending on the functionality of specific TOEs the Security IC Embedded Software will have to support the objective by providing runtime-tests of the random number generator. Together, these requirements allow the TOE to provide cryptographically good random numbers and to ensure that no information about the produced random numbers is available to an attacker.

It was chosen to define FCS_RNG.1 explicitly, because Part 2 of the Common Criteria does not contain generic security functional requirements for Random Number generation. (Note, that there are security functional requirements in Part 2 of the Common Criteria, which refer to random numbers. However, they define requirements only for the authentication context, which is only one of the possible applications of random numbers.)

The justification related to the security objective "Area based Memory Access Control (O.Mem-Access)" is as follows:

The security functional requirement "Subset access control (FDP_ACC.1)" with the related Security Function Policy (SFP) "Memory Access Control Policy" exactly require the implementation of an area based memory access control, which is a requirement from O.Mem-Access. Therefore, FDP_ACC.1 with its SFP is suitable to meet the security objective.

The security functional requirement "Static attribute initialisation (FMT_MSA.3)" requires that the TOE provides default values for the security attributes. Since the TOE is a hardware platform these default values are generated by the reset procedure. Therefore FMT_MSA.3 is suitable to meet the security objective O.Mem-Access.

The security functional requirement "Management of security attributes (FMT_MSA.1)" requires that the ability to change the security attributes is restricted to privileged subject(s). It ensures that the

access control required by O.Mem-Access can be realised using the functions provided by the TOE. Therefore FMT_MSA.1 is suitable to meet the security objective O.Mem_Access.

Finally, the security functional requirement "Specification of Management Functions (FMT_SMF.1)" is used for the specification of the management functions to be provided by the TOE as required by O.MEM_ACCESS. Therefore, FMT_SMF.1 is suitable to meet the security objective O.Mem_Access.

The justification related to the security objective "Additional Specific Security Functionality (O.Add-Functions)" is as follows:

The justification related to the security objective "Protection during Packaging, Finishing and Personalisation (OE.Process-Sec-IC)" is as follows:

The Composite Product Manufacturer has to use adequate measures to fulfil OE.Process-Sec-IC. Depending on the security needs of the application, the Security IC Embedded Software may have to support this for instance by using appropriate authentication mechanisms for personalisation functions.

6.3.2 Dependencies of Security Functional Requirements

Table 8 below lists the security functional requirements defined in this Protection Profile, their dependencies and whether they are satisfied by other security requirements defined in this Protection Profile. The text following the table discusses the remaining cases.

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FRU_FLT.2	FPT_FLS.1	Yes
FPT_FLS.1	None	No dependency
FMT_LIM.1	FMT_LIM.2	Yes
FMT_LIM.2	FMT_LIM.1	Yes
FAU_SAS.1	None	No dependency
FPT_PHP.3	None	No dependency
FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1	Yes
FDP_IFC.1	FDP_IFF.1	See discussion below
FPT_ITT.1	None	No dependency
FCS_RNG.1	None	No dependency

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FCS_COP.1 /3DES	FCS_CKM.4	Yes (by environment, see discussion below)
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) or FCS_CKM.1	Yes (by environment, see discussion below)
FCS_COP.1 /AES	FCS_CKM.4	Yes (by environment, see discussion below)
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) or FCS_CKM.1	Yes (by environment, see discussion below)
FDP_ACC.1	FDP_ACF.1	Yes
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Yes Yes
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Yes See discussion below
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	Yes See discussion below Yes
FMT_SMF.1	None	No dependency

Table 8: Dependencies of the Security Functional Requirements

Part 2 of the Common Criteria defines the dependency of FDP_IFC.1 (information flow control policy statement) on FDP_IFF.1 (Simple security attributes). The specification of FDP_IFF.1 would not capture the nature of the security functional requirement nor add any detail. As stated in the *Data Processing Policy* referred to in FDP_IFC.1 there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1 and its *Data Processing Policy* (FDP_IFC.1). Therefore the dependency is considered satisfied.

In particular the security functional requirements providing resistance of the hardware against manipulations (e. g. FPT_PHP.3) support all other more specific security functional requirements (e. g. FCS_RNG.1) because they prevent an attacker from disabling or circumventing the latter. Together with the discussion of the dependencies above this shows that the security functional requirements build a mutually supportive whole.

The functional requirements FCS_CKM.1 and FCS_CKM.4 which are dependent to FCS_COP.1/DES and FCS_COP.1/AES are not included in this Security Target since the TOE only provides an engine for encryption and decryption. But the Security IC Embedded Software may fulfill these requirements related to the needs of the implemented application. The dependent requirements of FCS_COP.1/DES and FCS_COP.1/AES concerning these functions shall be fulfilled by the environment (Security IC Embedded Software).

The dependency FMT_SMR.1 introduced by the two components FMT_MSA.1 and FMT_MSA.3 is considered to be satisfied because the access control specified for the intended TOE is not role-based but enforced for each subject. Therefore, there is no need to identify roles in form of a security functional requirement FMT_SMR.1.

6.3.3 Rationale for the Assurance Requirements

The assurance level EAL5 and the augmentation with the requirements ALC_DVS.2, and AVA_VAN.5 were chosen in order to meet assurance expectations explained in the following paragraphs.

An assurance level of EAL5 is required for this type of TOE since it is intended to defend against sophisticated attacks. This evaluation assurance level was selected since it is designed to permit a

developer to gain maximum assurance from positive security engineering based on good commercial practices. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defence against such attacks, the evaluators should have access to the low level design and source code.

ALC_DVS.2 Sufficiency of Security Measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE.

In the particular case of a Security IC the TOE is developed and produced within a complex and distributed industrial process which must especially be protected. Details about the implementation, (e.g. from design, test and development tools as well as Initialization Data) may make such attacks easier. Therefore, in the case of a Security IC, maintaining the confidentiality of the design is very important.

This assurance component is a higher hierarchical component to EAL5 (which only requires ALC_DVS.1). ALC_DVS.2 has no dependencies.

AVA_VAN.5 Advanced Methodical Vulnerability Analysis

Due to the intended use of the TOE, it must be shown to be highly resistant to penetration attacks. This assurance requirement is achieved by the AVA_VAN.5 component.

Independent vulnerability analysis is based on highly detailed technical information. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing high attack potential.

AVA_VAN.5 has dependencies to ADV_ARC.1 “Security Architectural Design”, ADV_FSP.4 “Complete functional specification”, ADV_TDS.3 “Basic modular design”, ADV_IMP.1 “Implementation representation of the TSF”, AGD_OPE.1 “Operational user guidance”, AGD_PRE.1 “Preparative procedures”, and ATE_DPT.1 “Testing: Basic design”.

All these dependencies are satisfied by EAL5.

It has to be assumed that attackers with high attack potential try to attack Security ICs like smart cards used for digital signature applications or payment systems. Therefore, specifically AVA_VAN.5 was chosen in order to assure that even these attackers cannot successfully attack the TOE.

6.3.4 Security Requirements are Internally Consistent

The discussion of security functional requirements 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 security functional requirements and assurance requirements support each other and that there are no inconsistencies between these groups.

The security functional requirement FPT_PHP.3 makes it harder to manipulate data. This protects the primary assets and other security features or functionality which use these data.

Though a manipulation of the TOE (refer to FPT_PHP.3) is not of great value for an attacker in itself, it can be an important step in order to threaten the primary assets. Therefore, the security functional requirement FPT_PHP.3 is not only required to meet the security objective O.Phys-Manipulation. Instead it protects other security features or functions of both the TOE and the Security IC Embedded Software from being bypassed, deactivated or changed. In particular this may pertain to the security features or functions being specified using FDP_ITT.1, FPT_ITT.1, FPT_FLS.1, FMT_LIM.2, FCS_RNG.1, and those implemented in the Security IC Embedded Software.

A malfunction of TSF (refer to FRU_FLT.2 and FPT_FLS.1) can be an important step in order to threaten the primary assets. Therefore, the security functional requirements FRU_FLT.2 and FPT_FLS.1 are not only required to meet the security objective O.Malfunction. Instead they protect other security features or functions of both the TOE and the Security IC Embedded Software from

being bypassed, deactivated or changed. In particular this pertains to the security features or functions being specified using FDP_ITT.1, FPT_ITT.1, FMT_LIM.1, FMT_LIM.2, FCS_RNG.1, and those implemented in the Security IC Embedded Software.

In a forced leakage attack the methods described in "Malfunction due to Environmental Stress" (refer to T.Malfunction) and/or "Physical Manipulation" (refer to T.Phys-Manipulation) are used to cause leakage from signals which normally do not contain significant information about secrets. Therefore, in order to avert the disclosure of primary assets it is important that the security functional requirements averting leakage (FDP_ITT.1, FPT_ITT.1) and those against malfunction (FRU_FLT.2 and FPT_FLS.1) and physical manipulation (FPT_PHP.3) are effective and bind well. The security features and functions against malfunction ensure correct operation of other security functions (refer to above) and help to avert forced leakage themselves in other attack scenarios. The security features and functions against physical manipulation make it harder to manipulate the other security functions (refer to above).

Physical probing (refer to FPT_PHP.3) shall directly avert the disclosure of primary assets. In addition, physical probing can be an important step in other attack scenarios if the corresponding security features or functions use secret data. For instance the security functional requirement FMT_LIM.2 may use passwords. Therefore, the security functional requirement FPT_PHP.3 (against probing) help to protect other security features or functions including those being implemented in the Security IC Embedded Software. Details depend on the implementation.

Leakage (refer to FDP_ITT.1, FPT_ITT.1) shall directly avert the disclosure of primary assets. In addition, inherent leakage and forced leakage (refer to above) can be an important step in other attack scenarios if the corresponding security features or functions use secret data. For instance the security functional requirement FMT_LIM.2 may use passwords. Therefore, the security functional requirements FDP_ITT.1 and FPT_ITT.1 help to protect other security features or functions implemented in the Security IC Embedded Software (FDP_ITT.1) or provided by the TOE (FPT_ITT.1). Details depend on the implementation.

According to the assumption Usage of Hardware Platform (A.Plat-Appl) the Security IC Embedded Software will correctly use the functions provided by the TOE. Hereby the User Data are treated as required to meet the requirements defined for the specific application context (refer to Treatment of User Data (A.Resp-Appl)). However, the TOE may implement additional functions. This can be a risk if their interface cannot completely be controlled by the Security IC Embedded Software. Therefore, the security functional requirements FMT_LIM.1 and FMT_LIM.2 are very important. They ensure that appropriate control is applied to the interface of these functions (limited availability) and that these functions, if being usable, provide limited capabilities only.

The combination of the security functional requirements FMT_LIM.1 and FMT_LIM.2 ensures that (especially after TOE Delivery) these additional functions cannot be abused by an attacker to (i) disclose or manipulate User Data, (ii) to manipulate (explore, bypass, deactivate or change) security features or functions of the TOE or of the Security IC Embedded Software or (iii) to enable an attack. Hereby the binding between these two security functional requirements is very important:

The security functional requirement Limited Capabilities (FMT_LIM.1) must close gaps which could be left by the control being applied to the function's interface (Limited Availability (FMT_LIM.2)). Note that the security feature or function which limits the availability can be bypassed, deactivated or changed by physical manipulation or a malfunction caused by an attacker. Therefore, if Limited Availability (FMT_LIM.2) is vulnerable, it is important to limit the capabilities of the functions in order to limit the possible benefit for an attacker.

The security functional requirement Limited Availability (FMT_LIM.2) must close gaps which could result from the fact that the function's kernel in principle would allow to perform attacks. The TOE must limit the availability of functions which potentially provide the capability to disclose or manipulate User Data, to manipulate security features or functions of the TOE or of the Security IC Embedded Software or to enable an attack. Therefore, if an attacker could benefit from using such functions, it is important to limit their availability so that an attacker is not able to use them.

No perfect solution to limit the capabilities (FMT_LIM.1) is required if the limited availability (FMT_LIM.2) alone can prevent the abuse of functions. No perfect solution to limit the availability

(FMT_LIM.2) is required if the limited capabilities (FMT_LIM.1) alone can prevent the abuse of functions. Therefore, it is correct that both requirements are defined in a way that they together provide sufficient security.

It is important to avert malfunctions of TSF and of security functions implemented in the Security IC Embedded Software (refer to above). There are two security functional requirements which ensure that malfunctions cannot be caused by exposing the TOE to environmental stress. First it must be ensured that the TOE operates correctly within some limits (Limited fault tolerance (FRU_FLT.2)). Second the TOE must prevent its operation outside these limits (Failure with preservation of secure state (FPT_FLS.1)). Both security functional requirements together prevent malfunctions. The two functional requirements must define the "limits". Otherwise there could be some range of operating conditions which is not covered so that malfunctions may occur. Consequently, the security functional requirements Limited fault tolerance (FRU_FLT.2) and Failure with preservation of secure state (FPT_FLS.1) are defined in a way that they together provide sufficient security.

The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced protect the cryptographic algorithms (FCS_COP.1) and the cryptographic key generations (FCS_CKM.1). Therefore these security functional requirements support the secure implementation and operation of FCS_COP.1 and FCS_CKM.1.

Parts of the Smartcard IC Embedded Software may cause security violations by accidentally or deliberately accessing restricted data (which may include code). In order to avert the memory access violation it is important to the security functional requirement defining the scope where the Memory Access Policy is applied (FDP_ACC.1) and the security functional requirement defining the Memory Access Policy(FDP_ACF.1), and the security functional requirement ensuring the default value of security attribute(FMT_MSA.3) and the security functional requirement managing security attribute (FMT_MSA.1) and the security functional requirement performing security management function(FMT_SMF.1) are effective and bind well.

Two refinements from the PP [5] have to be discussed here in the ST as the assurance level is increased. The refinement for ALC_CMS from the PP [5] can even be applied at the assurance level EAL 5 augmented with ALC_CMS.5. The assurance component ALC_CMS.4 is augmented to ALC_CMS.5 with aspects regarding the configuration control system for the TOE. The refinement is not touched. The refinement for ADV_FSP from the PP [5] can even be applied at the assurance level EAL 5 augmented with ADV_FSP.5. The assurance component ADV_FSP.4 is extended to ADV_FSP.5 with aspects regarding the description level. The level is increased from informal to semi-formal with informal description. The refinement is not touched by this measure.

7 TOE SUMMARY SPECIFICATION

This chapter 7 TOE Summary Specification contains the following sections:

7.1 List of Security Functional Requirements

7.1 List of Security Functional Requirements

SFR1: FPT_FLS.1: Failure with preservation of secure state

The detection thresholds of **TOE's detectors** are inside the operating range of the TOE. Therefore abnormal events/failures are detected before the secure state is compromised. This allows to take User's defined appropriate actions by software or to immediately RESET the TOE.

The secure state is maintained by TOE's detectors. The TOE's detectors are monitoring the failure occurs. The failures are abnormal frequency, abnormal voltage, abnormal temperature, and power glitch detectors that detect out of the specified range (refer to table 9). If the failures are happen, the TOE goes into RESET state. This satisfies the FPT_FLS.1 "Failure with preservation of secure state."

TOE's Detectors

These functions records in register the events notified by the detectors (refer to list below). The software configures the reaction in case of detection:

- The TOE is immediately reset when an event is detected.
- Or, a special function register bit is set.

List of detectors:

- Abnormal frequency Detector
- Abnormal voltage Detector
- Abnormal temperature Detector
- Light Detector
- Active shield removal Detector
- Glitch Detector (External/Internal power and Ground glitch)
- Life Cycle Detector

SFR2: FRU_FLT.2: Limited fault tolerance

All operating signals (Clock, RESET and supply voltage) are filtered/regulated in order to prevent malfunction.

TOE's Filters

These filters are used for preventing noise, glitches and extremely high frequency in the external reset or clock pad from causing undefined or unpredictable behavior of the chip.

- High Frequency Filter
- Reset Noise Filter

TOE's filters and detectors are implemented by the hardware. The filtering and detection cannot be affected or bypassed by Smartcard Embedded Software. The reaction to the detection can be configured by the software. The influence on security and the way how to configure it is described in details in the S3FT9MD/S3FT9MC *User's Manual*. Therefore, FRU_FLT.2 is implemented by TOE.

Detectors & Filters		Specification (typical)					
		Low			High		
		Min.	Typ.	Max.	Min.	Typ.	Max.
Voltage		1.3	1.5	1.62	5.5	6.2	7.0
Frequency		0.1	0.5	1.0	10	13	16
Temperature		-60	-45	-25	+85	+125	+150
Glitch Detector	External Power Glitch	Rising & falling time of Power is lower than 400ns					
	Internal Power Glitch	Rising time of Power is lower than 400ns					
	Ground Glitch	Rising time of Power is lower than 400ns					
High Frequency Filter		13Mhz and above					
Reset Noise Filter		Reset width 1000ns and below					

Table 9. Detectors & Filters Specification

Security domains are maintained since accesses to the access-prohibited area are trapped by this access control function. (Invalid memory access, MASCON register and MPU)

SFR3: FPT_PHP.3: Resistance to physical attacks

This requirement is achieved by security feature as the Active shield must be removed and bypassed in order to perform physical intrusive attacks. The TOE makes a reset or FIQ occurs to stops operation if a physical manipulation or physical probing attack is detected. And also Static Address/Data scrambling for bus and memory & Synthesizable processor core make the reverse-engineering of the TOE layout unpractical. So these functionalities meet the security functional requirement of FPT_PHP.3: Resistance to physical attack.

Static Address/Data scrambling for bus and memory protects memory and address/data bus from probing attacks.

Synthesizable processor core: The Central Processing Unit (CPU) of the TOE is synthesizable with glue logic, which makes reverse engineering and signal identification more difficult. Most sensitive hardware components such as buses are also hidden and implemented in deepest layers.

SFR4: FDP_ACC.1: Subset access control

This requirement is achieved by security register access control, invalid address access and access right for the code executed in FLASH.

1) Security registers access control: This security function manages access to the security control registers through access control security attributes. The USER mode has another function, which is write-enable bit for security related registers. If user does not enable this bit in 128cycles after the reset, user cannot write security control registers any more.

2) Invalid address access: This function detects invalid address access occurrence. In case of an invalid address access is detected, an FIQ is evoked. The memory access rights are defined and configured trough the control register MASCON and the Memory Protection Unit (MPU). The MPU provide the Embedded Software the ability to define different access rights for different data and program memory areas. In case of an illegal memory access, a non-maskable interrupt (FIQ) is generated, allowing to take dedicated and appropriate actions.

3) Access rights for the code executed in FLASH: This security function manages the code execution in FLASH, through access control security attributes in MPU. If an invalid access is detected, then a FIQ occurs.

4) Access control for Operating state: This security function select booting memory area. User can select ROM-BOOT or FLASH-BOOT. Also it can hide ROM memory. If OPRMON.1 is set "0', user cannot change ROM-HIDE value permanently.

5) Flash protection about Write operation: This function provides protection about flash write operation. When user write flash write enable bits, user can write flash memory. Each MAT(MAT1, MAT2) have protection bit, also have locking bit.

SFR5: FDP_ACF.1: Security attributes based access control.

This is covered by the Privilege and User modes of the TOE. The more information on chapter 1.2 Table 3. Privilege and User Modes basic description

SFR6: FMT_MSA.3: Static attribute initialization.

All Special Function Registers including MPU have DEFAULT values after Power on Reset.

SFR7: FMT_MSA.1: Management of security attributes.

This is achieved with the MPU feature. The Memory Protection Unit (MPU) enables user to partition memory and set individual protection attributes for each partition. This allows the operating system to control the memory regions accessible by a User mode application process. The protection unit enables user to divide memory into 8 regions, each with their own access permission attributes. If access against the set condition is performed, chip automatically generates FIQ, and sets a specific bit of FIQIMON register.

SFR8: FMT_SMF.1: Specification of management functions.

This is achieved via access to Special Function Registers of Memory Protection Unit(MPU). MPU provides Special Function Registers which defines the base address and the limit address for a partition. The Registers exist for Flash, and RAM. Additional Registers exist for defining the protection attribute for each partition.

SFR9: FAU_SAS.1: Audit Storage

This is fulfilled by the traceability/identification data written once and for all during the TEST mode of the manufacturing process.

1) Non-reversibility of TEST mode and NORMAL mode: This function disables the TEST mode and enables the NORMAL mode of the TOE. This function ensures the non-reversibility of the NORMAL mode. This function is used once during the manufacturing process.

2) **TEST mode communication protocol and data commands:** This function is the proprietary protocol used to operate the chip in TEST mode. This function enforces the identification and authentication of the TEST administrator during the test phase of the manufacturing process.

3) Functional Tests: During the manufacturing process, the operation of the TOE and the embedded software checksum are verified. This security function ensures the correct operation of the TOE security functions and the integrity of the embedded software.

4) Identification: During the TEST mode of manufacturing process, traceability data are written in the non-volatile memory of the TOE. Once the TOE is switched from TEST to NORMAL mode, those traceability data are READ ONLY and cannot be modified anymore. This enables to identify and track the TOE during the rest of its life.

SFR10: FMT_LIM.1: Limited capabilities

TEST mode can be accessed only by the TEST administrator by supplying an authentication password through a proprietary protocol. Once the TOE is changed to NORMAL mode, TEST mode functions are no more available for NORMAL mode.

SFR11: FMT_LIM.2: Limited availabilities

TEST mode can be accessed only by the TEST administrator by supplying an authentication password through a proprietary protocol. Once the TOE is changed to NORMAL mode, TEST mode commands are no more available for NORMAL mode. Functional test during manufacturing process is only available for TEST mode only.

SFR12: FDP_IFC.1: Subset information flow control

Memory Encryption: This is achieved by the function protects the memory contents of the TOE from data analysis on the stored data as well as on internally transmitted data. The algorithms used for encryption are proprietary. The ROM encryption is static key while the RAM and the FLASH encryption is dynamic key. RAM encryption is performed automatically while FLASH encryption is defined and managed by the embedded software. The key size for FLASH encryption is 16-byte.

SFR13: FDP_ITT.1: Basic internal transfer protection

This requirement is achieved by the combination of the TOE security features TOE features 1) to 5) as it is unpractical to get access to internal signals and interpret them.

1) Static Address/Data scrambling for bus and memory: This function protects memory and address/data bus from probing attacks.

2) Dynamic Data encryption for bus: This function protects data bus from probing attacks.

3) Memory encryption: This security function protects the memory contents of the TOE from data analysis on the stored data as well as on internally transmitted data. The algorithms used for encryption are proprietary. The ROM encryption is static key while the RAM and the FLASH encryption is dynamic key. RAM encryption is performed automatically while FLASH encryption is defined and managed by the embedded software. The key size for FLASH encryption is 16-byte.

4) Synthesizable processor core: The Central Processing Unit (CPU) of the TOE is synthesizable with glue logic, which makes reverse engineering and signal identification more difficult. Most sensitive hardware components such as buses are also hidden and implemented in deepest layers.

5) De-synchronization and signal-to-noise ratio reduction mechanisms: The TOE operations can be made asynchronous by using the Internal Variable Clock, Random Current Generator and the Random Wait Generator security features. They make a full range of intrusive (e.g. probing attacks) and non intrusive attacks (e.g. side-channel attacks) more complex and difficult.

SFR14: FPT_ITT.1: Basic internal TSF data transfer protection

This requirement is achieved by the combination of the TOE security features TOE features 1) to 5) as it is unpractical to get access to internal signals and interpret them.

1) Static Address/Data scrambling for bus and memory: This function protects memory and address/data bus from probing attacks.

2) Dynamic Data encryption for bus: This function protects data bus from probing attacks.

3) Memory encryption: This security function protects the memory contents of the TOE from data analysis on the stored data as well as on internally transmitted data. The algorithms used for encryption are proprietary. The ROM encryption is static key while the RAM and the FLASH encryption is dynamic key. RAM encryption is performed automatically while FLASH encryption is defined and managed by the embedded software. The key size for FLASH encryption is 16-byte.

4) Synthesizable processor core: The Central Processing Unit (CPU) of the TOE is synthesizable with

glue logic, which makes reverse engineering and signal identification more difficult. Most sensitive hardware components such as buses are also hidden and implemented in deepest layers.

5) De-synchronization and signal-to-noise ratio reduction mechanisms: The TOE operations can be made asynchronous by using the Internal Variable Clock, Random Current Generator and the Random Wait Generator security features. They make a full range of intrusive (e.g. probing attacks) and non intrusive attacks (e.g. side-channel attacks) more complex and difficult.

SFR15: FCS_RNG.1: Random number generation

This requirement is ensured by the design of the random number generation algorithm that follows Digital True Random Number Generator (DTRNG) that follows the requirement of the "*standard*" level of ANSSI requirements as well as *BSI-AIS31 Class P2* requirements (German metric) for Random Number Generation (FCS_RNG.1).

SFR16: FCS_COP.1: Cryptographic operation

This requirement is covered by the TOE.

Triple Data Encryption Standard Engine

This function is used for encrypting and decrypting data using the Triple DES symmetric algorithm with 112bit or 168bit key size. (FCS_COP.1/3DES)

AES (Advanced Encryption Standard)

This function supports the AES operation with 128 bit, 192bit and 256bit key size. (FCS_COP.1/AES)

8 ANNEX

8.1 Glossary

Application Data

All data managed by the Security IC Embedded Software in the application context. Application data comprise all data in the final Security IC.

Composite Product Integrator

Role installing or finalising the IC Embedded Software and the applications on platform transforming the TOE into the unpersonalised Composite Product after TOE delivery. The TOE Manufacturer may implement IC Embedded Software delivered by the Security IC Embedded Software Developer before TOE delivery (e.g. if the IC Embedded Software is implemented in ROM or is stored in the non-volatile memory as service provided by the IC Manufacturer or IC Packaging Manufacturer)

Composite Product Manufacturer

The Composite Product Manufacturer has the following roles (i) the Security IC Embedded Software Developer (Phase 1), (ii) the Composite Product Integrator (Phase 5) and (iii) the Personaliser (Phase 6). If the TOE is delivered after Phase 3 in form of wafers or sawn wafers (dice) he has the role of the IC Packaging Manufacturer (Phase 4) in addition.

End-consumer

User of the Composite Product in Phase 7.

IC Dedicated Software

IC proprietary software embedded in a Security IC (also known as IC firmware) and developed by the IC Developer. Such software is required for testing purpose (IC Dedicated Test Software) but may provide additional services to facilitate usage of the hardware and/or to provide additional services (IC Dedicated Support Software)..

IC Dedicated Test Software

That part of the IC Dedicated Software (refer to above) which is used to test the TOE before TOE Delivery but which does not provide any functionality thereafter.

IC Dedicated Support Software

That part of the IC Dedicated Software (refer to above) which provides functions after TOE Delivery. The usage of parts of the IC Dedicated Software might be restricted to certain phases.

Initialisation Data

Initialisation Data defined by the TOE Manufacturer to identify the TOE and to keep track of the Security IC's production and further life-cycle phases are considered as belonging to the TSF data. These data are for instance used for traceability and for TOE identification (identification data).

Integrated Circuit (IC)

Electronic component(s) designed to perform processing and/or memory functions.

Pre-personalisation Data

Any data supplied by the Card Manufacturer that is injected into the non-volatile memory by the Integrated Circuits manufacturer (Phase 3). These data are for instance used for traceability and/or to secure shipment between phases.

Security IC

Composition of the TOE, the Security IC Embedded Software, User Data and the package (the Security IC carrier).

Security IC Embedded Software

Software embedded in a Security IC and normally not being developed by the IC Designer. The Security IC Embedded Software is designed in Phase 1 and embedded into the Security IC in Phase 3 or in later phases of the Security IC product life-cycle. Some part of that software may actually implement a Security IC application others may provide standard services. Nevertheless, this distinction doesn't matter here so that the Security IC Embedded Software can be considered as being application dependent whereas the IC Dedicated Software is definitely not.

Security IC Product

Composite product which includes the Security Integrated Circuit (i.e. the TOE) and the Embedded Software and is evaluated as composite target of evaluation in the sense of the Supporting Document

TOE Delivery

The period when the TOE is delivered which is either (i) after Phase 3 (or before Phase 4) if the TOE is delivered in form of wafers or sawn wafers (dice) or (ii) after Phase 4 (or before Phase 5) if the TOE is delivered in form of packaged products.

TOE Manufacturer

The TOE Manufacturer must ensure that all requirements for the TOE and its development and production environment are fulfilled. The TOE Manufacturer has the following roles: (i) IC Developer (Phase 2) and (ii) IC Manufacturer (Phase 3). If the TOE is delivered after Phase 4 in form of packaged products, he has the role of the (iii) IC Packaging Manufacturer (Phase 4) in addition.

TSF data

Data created by and for the TOE, that might affect the operation of the TOE. This includes information about the TOE's configuration, if any is coded in non-volatile non-programmable memories (ROM), in specific circuitry, in non-volatile programmable memories (for instance E2PROM) or a combination thereof.

User data

All data managed by the Smartcard Embedded Software in the application context. User data comprise all data in the final Smartcard IC except the TSF data.

8.2 Abbreviations

CC

Common Criteria

EAL

Evaluation Assurance Level

IT

Information Technology

PP

Protection Profile

ST

Security Target

TOE

Target of Evaluation

TSC

TSF Scope of Control

TSF

TOE Security Functionality

TSFI

TSF Interface

TSP

TOE Security Policy

8.3 Literature

[1] Common Criteria, Part 1: Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-001

[2] Common Criteria, Part 2: Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-002

[3] Common Criteria, Part 3: Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 4, Sept. 2012, CCMB-2012-09-004

[5] Eurosmart Security IC Platform Protection Profile, Version 1.0, June 2007, BSI-PP-0035.

[6] AIS31: Functionality classes and evaluation methodology for true (physical) random number generators, Version 1, 25.09.2001, Bundesamt für Sicherheit in der Informationstechnik

[7] AIS20: Functionality classes and evaluation methodology for deterministic random number generators, Version 1, 2.12.1999, Bundesamt für Sicherheit in der Informationstechnik

[8] ALGO: Federal Gazette No 19, Notification in accordance with the Electronic Signatures Act and the Electronic Signatures Ordinance (overview of suitable algorithms), Federal Network Agency for Electricity, Gas, Telecommunications, Post and Railway, 2008-11-17

[9] [FIPS SP800-67] Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, version 1.1

[10] [FIPS 197] Advanced Encryption Standard (AES), 2001-11-26

[11] CC Supporting Document, Mandatory Technical Document, **"Application of Attack Potential to Smartcards": version 2.9 (January 2013) as recommended by SOG-IS.**