Public

Common Criteria
Information Technology
Security Evaluation

Security Target Lite

Samsung S3FV9VH 32-bit RISC Microcontroller for Smart Card with specific IC Dedicated Software

Version 1.1

31th August 2017

SAMSUNG

ELECTRONICS

REVISION HISTORY

UPDATES:

Version	Date	Modification
1.0	3rd July 2017	• Creation
1.1	31th August 2017	• The table 1and chapter 4.3 are updated • Chapter 7.3 is updated

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 31th August 2017.

The Security Target Lite is based on

[5] Eurosmart Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, BSI-CC-PP-0084-2014.

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

- Title: Security Target Lite of Samsung S3FV9VH 32-bit RISC Microcontroller for Smart Card with specific IC Dedicated Software
- Target of Evaluation: S3FV9VH 32-Bit RISC Microcontroller for Smart Cards, Revision 0 including specific IC Dedicated Software
- 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 S3FV9VH microcontroller featuring the a 32-bit SC300 ARM core, a TORNADO-H coprocessor for big integer arithmetic, AES/DES engines for symmetric cryptography and a true random number generator is a smartcard integrated circuit which is composed of a central processing unit (CPU), security components, contact based I/O ports, contactless based I/O ports, hardware circuit for testing purpose during the manufacturing process and volatile and non-volatile memories (hardware). The SC300 CPU architecture of the S3FV9VH microcontroller follows the Harvard style, that is, it has dedicated buses for program memory and data memory. Both instruction and data can be fetched simultaneously without causing a pipeline stall because there are separate paths for memory access. 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, e.g., PTG.2-compliant true random number generator as defined in [6, 7].

1.2.2 Physical scope

The S3FV9VH single-chip CMOS 32-bit microcontroller is designed and packaged specifically for "Smart Card" applications with both contact and contactless communication.

The main security features of the S3FV9VH device are:

- An ARM-based secure core SC300 which is a secure variation of ARM Cortex-M3 architecture. Some of SC300 features include Thumb-2 instruction set, a configurable Nested Vector Interrupt Controller (NVIC) for highly deterministic and low interrupt latency and integrated Memory Protection Unit (MPU) for memory access control.
- Security sensors, detectors or Filter
- Life Cycle detector
- Shields against physical intrusive attacks.
- Dedicated countermeasures for reverse-engineering including memory encryption and address scrambling.
- Dedicated tamper-resistant design based on synthesizable glue logic and secure topology
- Triple-DES hardware supporting both encryption and decryption in ECB mode
- AES hardware supporting both encryption and decryption in ECB mode
- Dedicated hardware countermeasures against side-channel analysis
- Accelerator called TORNADO-H with dedicated SRAM for its operations (referred to as Crypto RAM) for multi-precision integer arithmetic operations
- A true random number generator called DTRNG FRO M for secure random number generation compliant with PTG.2 requirements. Please refer to Section 5.1 and [6, 7] which defines PTG.2 RNG characteristics.

The main hardware blocks of the S3FV9VH Integrated Circuit are described in **Figure 1** below:

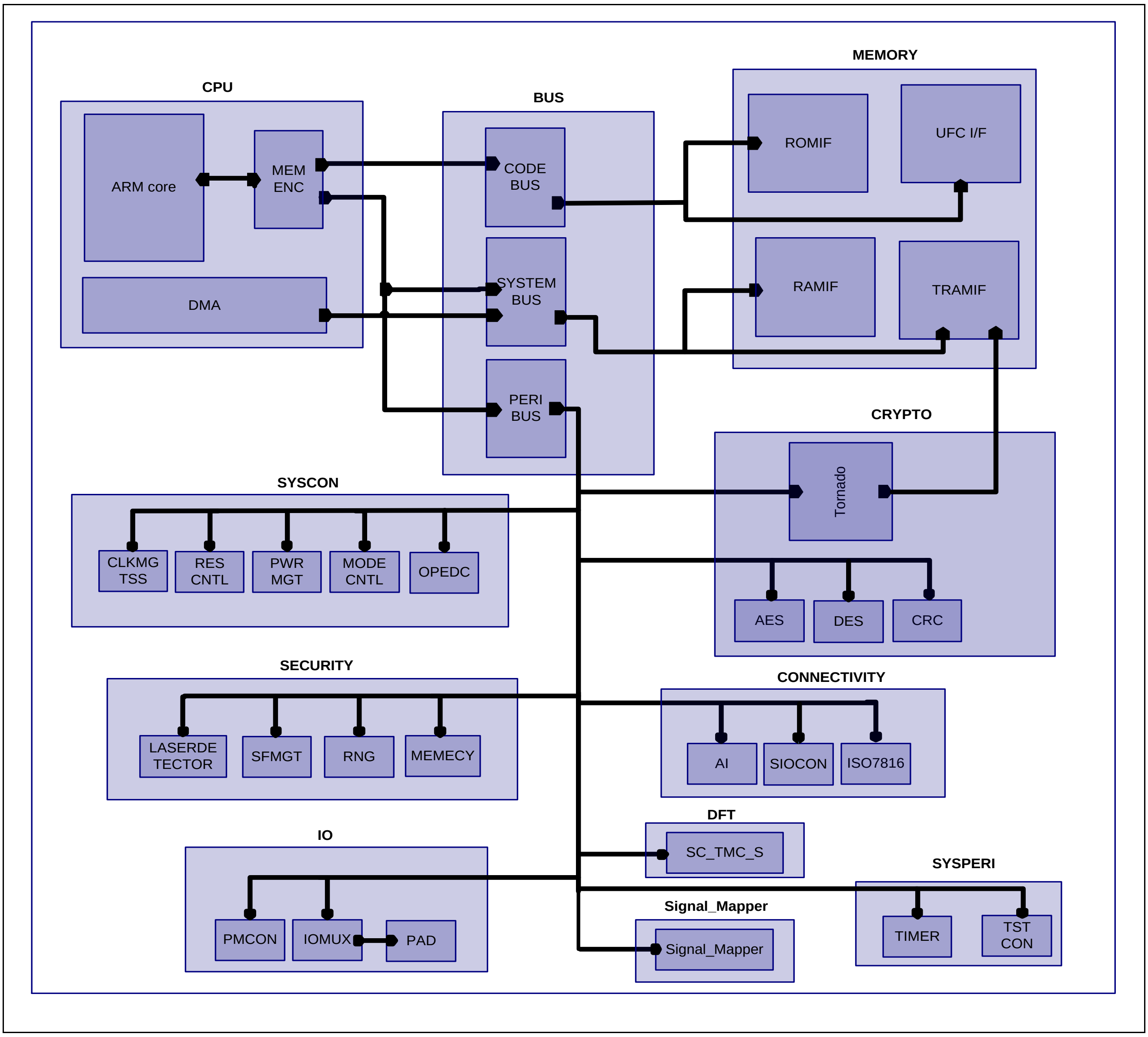


Figure 1: S3FV9VH Block Diagram

*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

CPU

- ARM SC300 32-bit core

Memory

- S3FV9VH 512Kbytes
- 56Kbytes User ROM for Bootloader
- 16Kbytes General-purpose RAM (SRAM)
- 7Kbytes Crypto RAM (Crypto RAM)
- 4Kbytes DMA RAM for Contactless operation (CLRAM)
- 2KB cache memory for code

FLASH Write Operations

TORNADO-H

- A coprocessor supporting Montgomery multiplication for public key cryptography
- Built-in countermeasure against timing and power analysis attacks

Triple DES

- Built-in hardware Triple DES accelerator supporting in ECB mode
- Circuit for resistance against SPA and DPA attacks

AES

- Built-in hardware AES accelerator supporting in ECB mode
- Circuit for resistance against SPA and DPA attacks

Abnormal Condition Detectors and Filters

Interrupts

- Nested Vector Interrupt Controller

Serial I/O Interface

- UART (ISO7816) for contact I/O interface
- Contactless UART (ISO 14443) for contactless I/O interface

Reset

- reset

Random Number Generator

- A Digital True random number generator

Memory Protection Unit

- Memory protection unit (MPU) up to 4 GB

Memory Encryption and Bus Encryption

Timers

- Two 16-Bit timers
- A 20-bit Watchdog Timer

Parity and CRC calculator

- CRC-32 calculator
- Parity calculator

Clock Sources

- Internal clock

Operating Voltage Range

- 1.62V to 5.5V

Operating Temperature

- - 25°C to 85°C

Package

- Wafer
- 8/6-pin COB

TOE Software

The TOE software comprises the following components that are in the scope of evaluation:

- Bootloader that is used for downloading software or data to FLASH memory. Bootloader's main functions are secure authentication and secure downloading. To download files into FLASH memory, it needs to be checked who is an authorized user. If the authentication is failed, cannot proceed any more.
- A DTRNG FRO M library built on top of DTRNG FRO M H/W that fulfills the requirements of PTG.2 class.
- Test ROM code is stored in Test ROM and locked before the TOE is delivered by disabling the Test Mode. This function is used once during for wafer testing the manufacturing process.

Please note that any other embedded software is not part of the TOE and therefore out of the scope of evaluation.

Item Type	Item	Version	Form of delivery
Hardware	S3FV9VH 32-Bit RISC Microcontroller for Smart Card	0	Wafer or Module
Software	Test ROM Code	1.0	Included in S3FV9VH Test ROM (part of physical cope but not logical scope of the TOE)
Software	Secure Bootloader code (S3FV9VH_Bootloader_v0.0.zip)	0.0	Included in S3FV9VH in ROM
Software	DTRNG FRO M Library (S3FV9VH_PTG2_DTRNG_library_v1.0.lib)	1.0	Software Library. This library is delivered as object file and is optionally integrated into user NVM code.
Document	DTRNG FRO M H/W and DTRNG FRO M library application note (S3FV9VH_DTRNG_FRO_M_AN_v1.2.pdf)	1.2	Softcopy
Document	Hardware User's manual (S3FV9VH_UM_REV0.4)	0.4	Softcopy
Document	Security Application Note (SAN_S3FV9VH_v0.5.pdf)	0.5	Softcopy
Document	Chip Delivery Specification (S3FV9VH_DV12.pdf)	1.2	Softcopy
Document	Boot Loader Specification (S3FV9VH_for_Bootloader_Specification_v0.2.pdf)	0.2	Softcopy
Document	SC300_Reference_Manual (SC300_Reference_Manual_v0.0.pdf)	0.0	Softcopy

Table 1. TOE Configuration

1.2.3 Logical scope

The TOE distinguishes two different modes of execution: TEST mode and NORMAL mode. TEST mode is only accessible for wafer tests during production. After switching to NORMAL mode, the TOE mode cannot be changed to TEST mode for the rest of its life. NORMAL mode further distinguishes PRIVILEGED mode and USER mode (Unprivileged) of execution which are both available to the user's IC Embedded Software that is not in the scope of the TOE.

Code running in USER mode has limited access to resources of the TOE while code running in PRIVILEGED mode has access to all resources. For instance, code running in USER mode is not allowed to use some instructions such as CPS to set FAULTMASK and PRIMASK.

The processor of the TOE supports two operation modes, Thread mode and Handler mode. Thread mode is entered on reset and normally on return from an exception. When in Thread mode, code can be executed as either Privileged or User. Handler mode will be entered as a result of an exception. Code in Handler mode is always executed as Privileged, therefore the core will automatically switch to Privileged mode when exceptions occur.

Code running in the PRIVILEGED mode can switch to USER mode via CONTROL register. When an exception takes place, the CPU always switches back to PRIVILEGED mode and returns to the previous state when exiting the exception handler. A user program running in USER mode cannot switch to PRIVILEGED mode by writing to CONTROL register. It has to go through an exception handler that programs CONTROL register to switch the processor to the privileged access level.

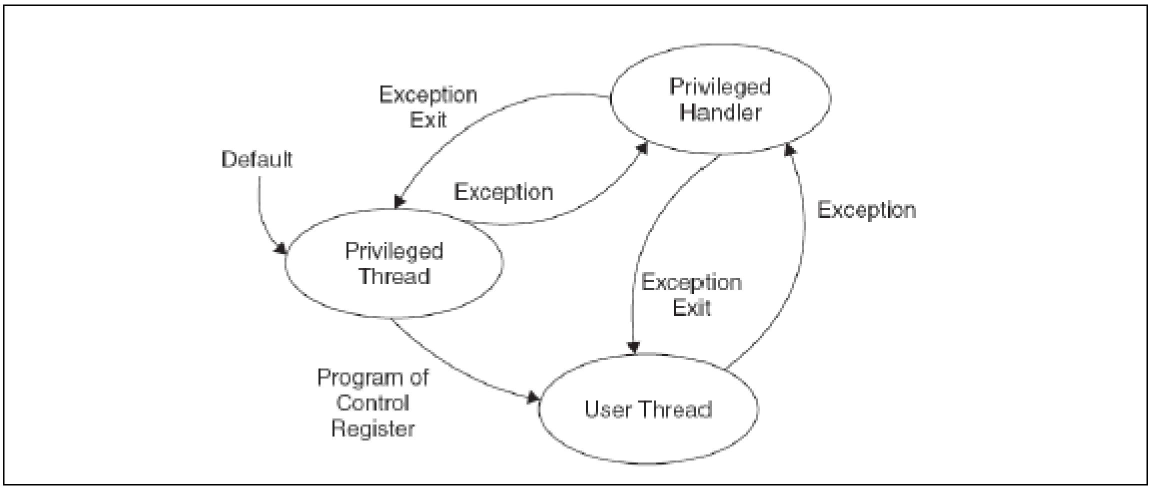


Table 2. Privilege and User Modes

The booting mode consists of ROM Booting mode and FLASH Booting mode.

- ROM Booting mode: This mode starts the code from ROM.
- FLASH Booting mode: This mode starts the code from FLASH.

The TOE features the following functionalities.

CPU:

- ARM SC300 CPU can operate in one of two operating states
- The built-in memory protection unit (MPU) can be used to control accesses to critical memory areas.

Memory:

- The TOE supports 56Kbytes for User ROM which is occupied by Bootloader and 12Kbytes for Test ROM which is occupied by code for Test Mode. Users cannot use these memory areas.
- The TOE supports 512Kbytes NOR flash which can be used for both users' code and data.
- The TOE has total 27Kbytes of SRAM, of which, 16Kbytes are for general-purpose RAM, 7Kbytes (Crypto RAM) are for TORNADO-H coprocessor and 4Kbytes (DMA RAM) are for contactless operation.

SYSCON:

- When the TOE is in reset state, all control registers are set to their respective hardware reset values.
- Internal clock supports variable clock frequency.

Cryptography:

- Triple-DES encryption and decryption in ECB mode.
- AES encryption and decryption in ECB mode.
- A digital true random number generator (DTRNG FRO M) which is capable of generating PTG.2-compliant random numbers is supported.

Security:

- The secure state is maintained by the TOE's security detectors. These detectors monitor if the TOE is under external attacks or its security policies are being violated.

Connectivity, IO and PAD:

- The high frequency filter is used to cut off extremely high frequencies of clock signal.

The Security IC dedicated Support Software:

- Boot loader code is in ROM. The Bootloader that is used for downloading software or data to FLASH memory.

Software Library:

- A Digital True Random Number Generator (DTRNG FRO M) library that fulfills the requirements of *PTG.2 class*.

Please note that Test ROM code is not part of the TOE logical scope as it is locked after being manufactured and tested by Samsung.

Security IC Embedded Software like operating system and applications of Security IC is developed by the customers. It is stored in Flash. It is not part of TOE.

In order to securely use the TOE, a set of guidance documents are provided, including:

- The "DTRNG FRO M H/W and DTRNG FRO M library application note" gives some information on how to use DTRNG FRO M library in compliance with the AIS31 PTG.2 requirements.
- The "Hardware User's manual" provides the functional description of the TOE.
- The "Security Application Note" describes the security features implemented in the TOE and to give user guide how to use the TOE in a secure manner.
- The "Chip Delivery Specification" describes physical identifications of the TOE and the secure delivery process.
- The "Boot Loader Specification" describes features and functionalities of Bootloader in the TOE.
- The "SC300_Reference_Manual" describes features and instruction set of the CPU of the TOE.

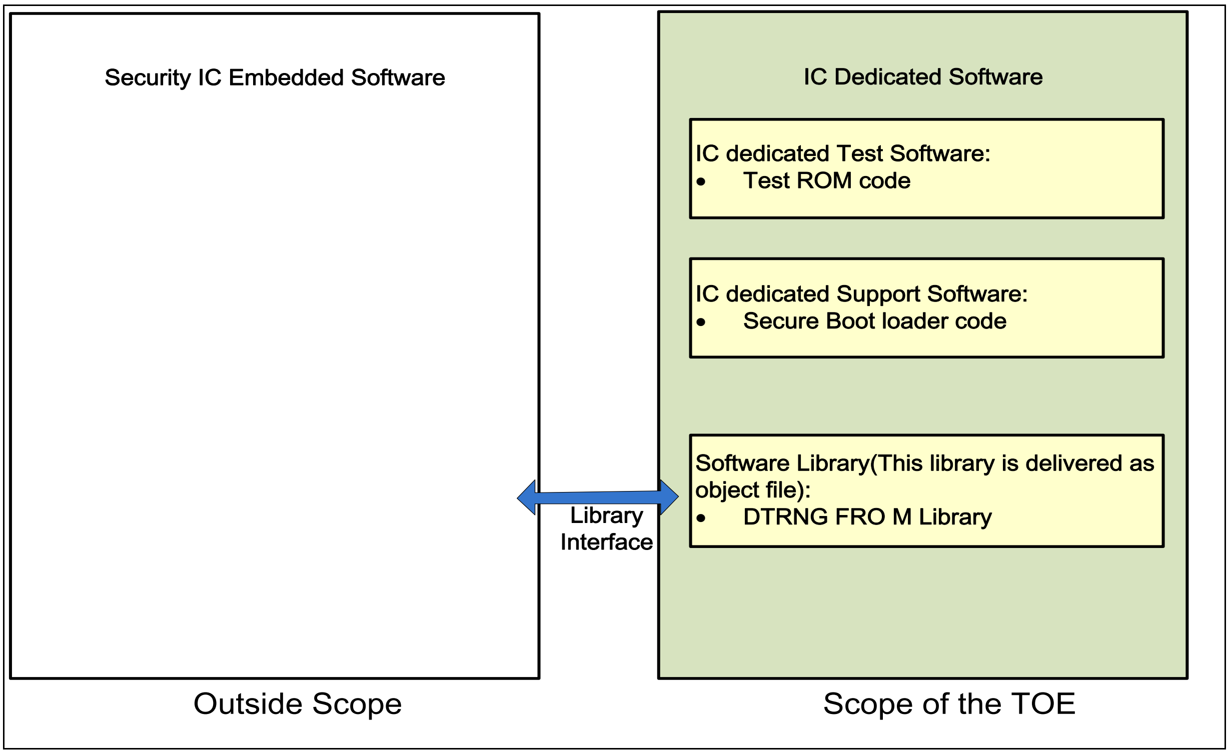

Security IC Embedded Software
IC Dedicated Software
IC dedicated Test Software:
• Test ROM code
IC dedicated Support Software:
• Secure Boot loader code
Software Library(This library is delivered as object file):
• DTRNG FRO M Library
Library Interface
Outside Scope
Scope of the TOE

Figure 3: Scope of the TOE S/W

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:

Site / Building	Phase
Hwasung Plant	Phase 2+3
Giheung Plant	Phase 3
Onyang Plant	Phase 3+4
PKL Plant	Phase 3
HANAMICRON Plant	Phase 3+4
Inesa Plant	Phase 3+4
Eternal Plant	Phase 3+4
TESNA Plant	Phase 3
ASE Korea	Phase 3+4

Table 2. TOE Sites & Phases

- 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-personalization 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-personalization 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 personalization line for the Composite Product (Composite Product Integration Phase 5),
- the Composite Product personalization and testing stage where the User Data is loaded into the Security IC's memory (Personalization 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.

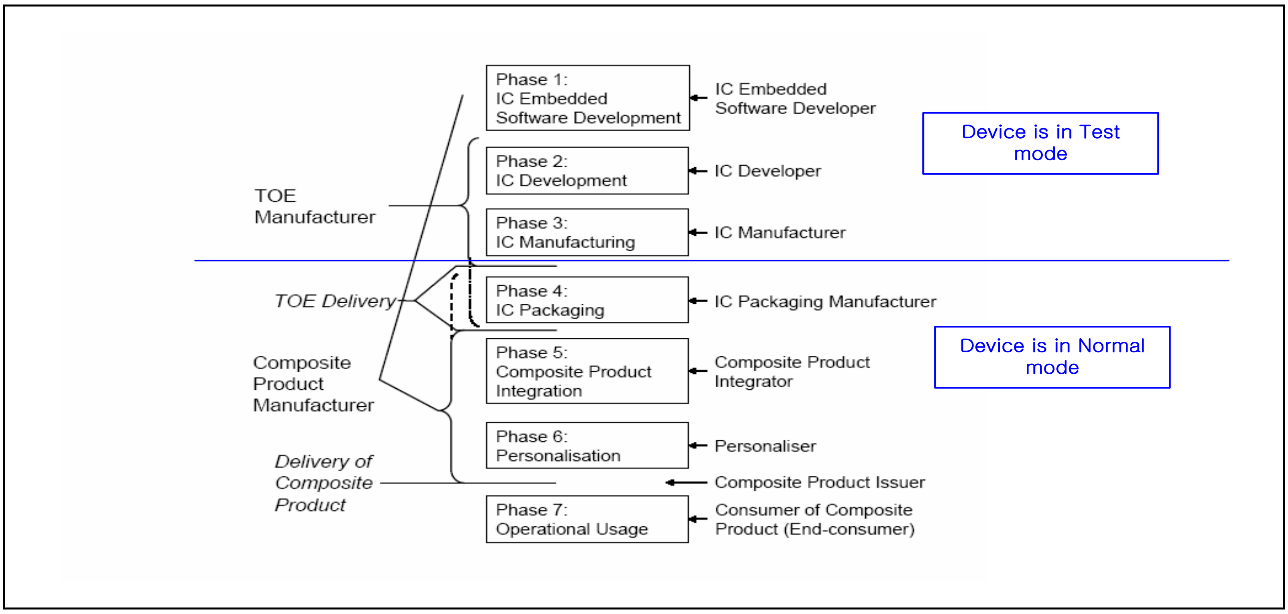


Figure 4: Definition of "TOE Delivery" and responsible Parties

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. In short, there are several options to add IC Embedded Software the TOE as follows:

- Option 1: The IC Embedded Software is stored in User ROM and has to be finished in Phase 2 or Phase 3. In this case, Bootloader is not available.
- Option 2: The IC Embedded Software is stored in FLASH. This can be done in either Phase 3 or Phase 4 at the Samsung's site. After downloading the software, the loader shall be deactivated.
- Option 3: The IC Embedded Software is stored in FLASH but the programming procedure is done in either Phase 5 or Phase 6 at the IC Embedded Software developer's certified sites. In this case, a successful mutual authentication between the loader and programming device is required. The authentication key is selected by IC Embedded Software developer and can be injected into the TOE by Samsung or modified by IC Embedded Software developer themselves via Loader interface. After downloading the software, the IC Embedded Software developer is responsible to lock the loader.

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 FRO M 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

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

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 claims strict conformance 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-CC-PP-0084, Version 1.0, dated 01.2014.

This ST does not claim conformance to any other PP.

2.3 Package Claim

The assurance level for this Security Target is EAL6 augmented with ASE_TSS.2.

This ST claims compliance to the following packages described in [5]:

- Package "Authentication of the Security IC": authentication of the TOE operating under secured environments at Samsung certified sites and IC Embedded S/W developer sites up to Phase 6 (**package conformant)** as described in Section 7.2 of [5].
- Loader Package 1+: Loader dedicated for usage in secured environments (MSSR audited) up to Phase 6 (**package augmented**) as described in Section 7.3.1 of [5] and Section 3 of [17].
- Package "TDES": Cryptographic service Triple-DES (**package conformant**) as described in Section 7.4.1 of [5].
- Package "AES": Cryptographic service AES (**package conformant**) as described in Section 7.4.2 of [5]

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 (with respect to vulnerability analysis in [12]). The Assurance Requirements of the TOE obtain the Evaluation Assurance Level 6 augmented with the assurance components ASE_TSS.2 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 2.2, 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, organizational 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.

The TOE provides additional security functionalities including cryptographic services TDES and AES. In order to maintain strict conformance to the PP [5] by the security target, the two Packages “TDES” and “AES” described in Section 7.4.1 and 7.4.2 of the PP [5] are used with the following SFRs are implemented:

- FCS_COP.1/TDES “Cryptographic operation - TDES”
- FCS_CKM.4/TDES “Cryptographic key destruction – TDES”
- FCS_COP.1/AES “Cryptographic operation - AES”
- FCS_CKM.4/AES “Cryptographic key destruction – AES”

The augmented package for Loader, i.e., Loader package 1+, has been added according to the interpretation of the PP [5] by ANSSI in [17]. The Loader package 1+ is a superset of Loader package 1 in a sense that the former requires some additional SFRs to be implemented. More specifically:

- FMT_LIM.1/Loader “Limited capabilities” : this SFR is required by both Loader package 1 and package 1+
- FMT_LIM.2/Loader “Limited availability– Loader”: this SFR is required by both Loader package 1 and package 1+
- FDP_ACC.1/Loader “Subset access control – Loader”: this SFR is required by Loader package 1+
- FDP_ACF.1/Loader “Security attribute based access control - Loader”: this SFR is required by Loader package 1+

The additional two SFRs, FDP_ACC.1/Loader and FDP_ACF.1/Loader, are justified due to the requirement of ANSSI interpretation [17] and the functionality available in the TOE which also covers the Masquerade_TOE threat mentioned in the security target. For this same reason, the security target makes use of the Package “Authentication of the Security OC” described in the PP [5]. The SFR FIA_API.1 is therefore also justified.

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 of the Composite TOE,
- the Security IC Embedded Software, stored and in operation,,
- 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 of the Composite TOE,

SC2 confidentiality of user data of the Composite TOE being stored in the TOE's protected memory areas,

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

Note the Security IC Embedded Software is user data and shall be protected while being executed/processed and while being stored in the TOE's protected 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 of the Composite TOE in integrity and in confidentiality if stored in protected memory areas, 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. Parts of the Security IC Embedded Software which do not contain secret data or security critical source code, may not require protection from being disclosed. Other parts of the Security IC Embedded Software may need to be kept confidential since 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 an 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 includes user data and code of the Composite TOE, 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.

- Disclosure of data (which may include user data and code of the Composite TOE, stored in or processed by the Security IC) means that an attacker is realistically[1] able to determine a meaningful block of data. This should be considered for the threats T.Leak-Inherent, T.Phys-Probing, T.Leak-Forced and T.Abuse-Func.

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

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

The cloning of the functional behavior 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 behavior requires to (i) develop a functional equivalent of the Security IC Embedded Software, (ii) disclose, interpret and employ the secret User Data of the Composite TOE 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 of the Composite TOE are stored and processed in a secure way (refer to below). The Security IC Embedded Software must also ensure that critical User Data of the Composite TOE are treated as required in the application context. In addition, the personalization 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 behavior 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 personalization 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 5). 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.

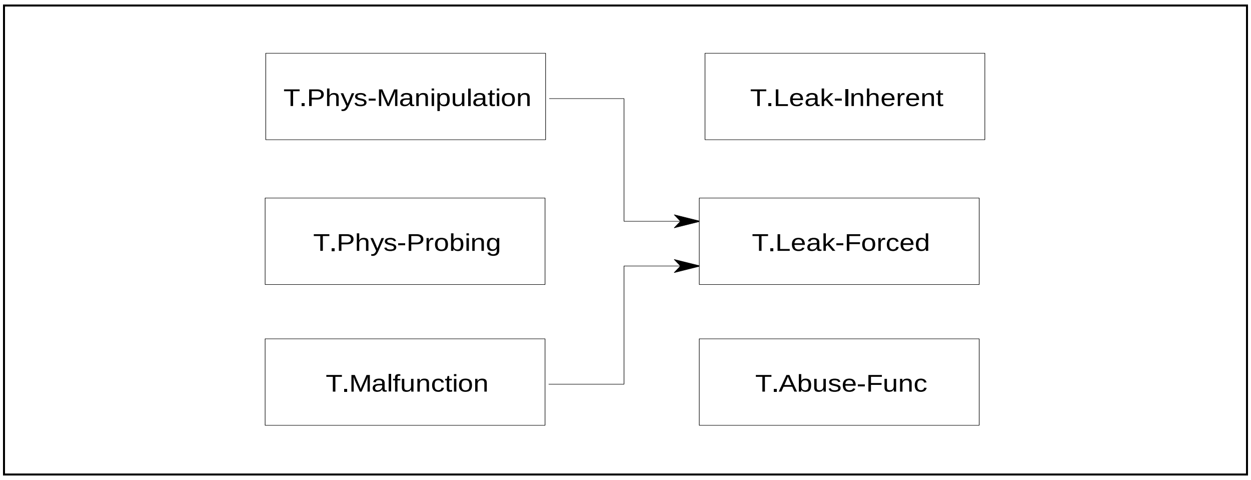


Figure 5: Standard Threats

The following explanations help to understand the focus of the threats and objectives related to security services, which also addresses Application Note 4 mentioned in [5].

- The security of cryptographic operations including key generation, authentication, etc relies on the assumption of generated random numbers are indistinguishable from ones generated by an ideal random number generator. Therefore, if the TOE's random number generator generates predictable or low-entropy output in any way or form, the application security of the TOE can be compromised. This should be considered as a threat, namely T.RND.

- IC Embedded Software may intentionally or unintentionally attempt access to restricted memory areas (containing either sensitive code or data). This can lead to corruption or exposure of sensitive assets. This should be considered as a threat, namely T.Mem-Access.

- The TOE is responsible for safeguarding user data of the Composite TOE and doing sensitive operations with the data. Therefore, if the TOE is misrepresented by a malicious device, users are at risk of submitting sensitive data to the malicious device. This should be considered as a threat, namely T.Masquerade_TOE.

The high-level security concern related to security service is refined below by defining threats as required by the Common Criteria (refer to Figure 6).

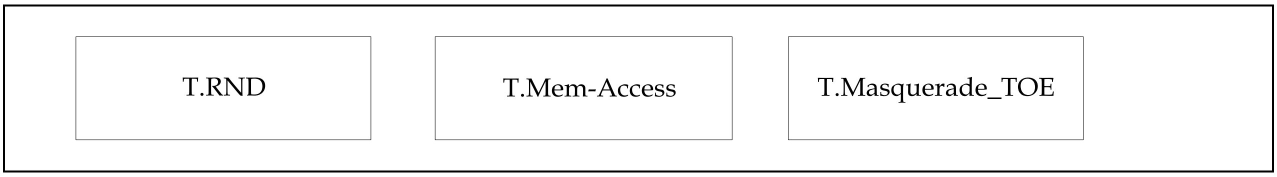


Figure 6: Threats related to security service

The Security IC Embedded Software maybe required 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 7. Due to the intended usage of the TOE all interactions are considered as possible.

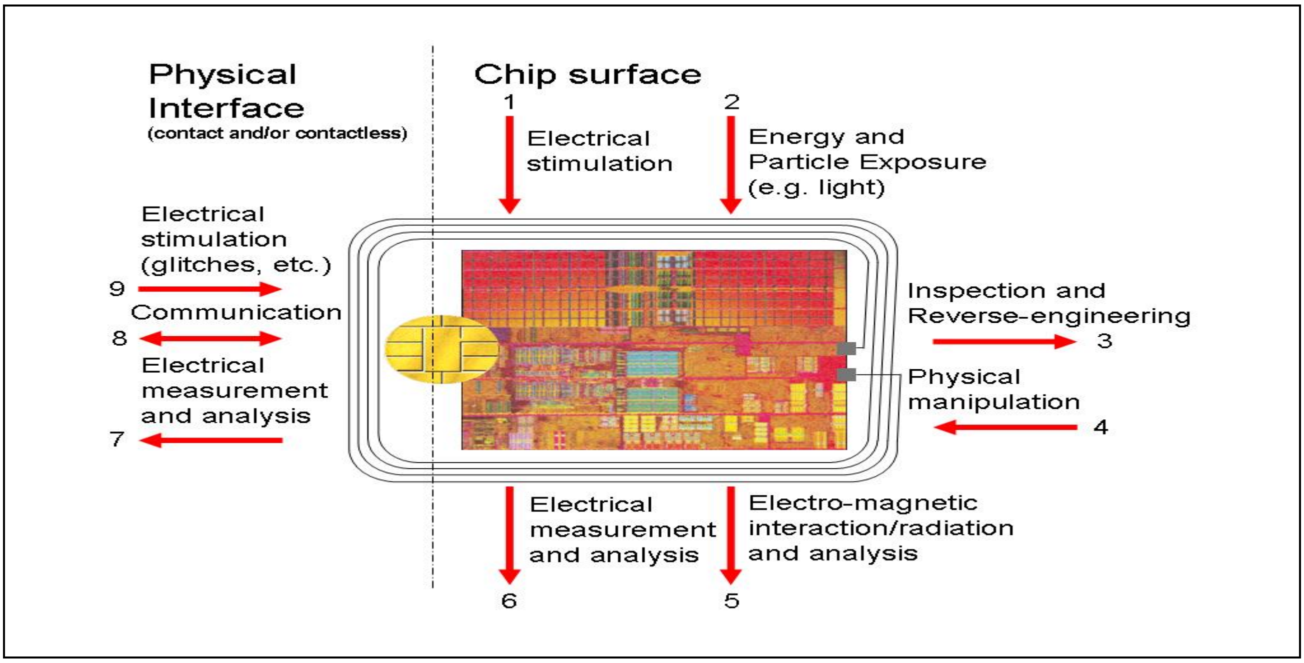

Physical Interface
(contact and/or contactless)
Chip surface
1
Electrical stimulation
2
Energy and Particle Exposure (e.g. light)
Electrical stimulation (glitches, etc.)
9
Communication
8
Electrical measurement and analysis
7
Inspection and Reverse-engineering
3
Physical manipulation
4
Electrical measurement and analysis
6
Electro-magnetic interaction/radiation and analysis
5

Figure 7: 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 7) which are realized using contacts interface. Influences or interactions with the TOE also occur through the chip surface (Number 1 – 6 in Figure 7). 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 behavior 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 (via Section 3.2 of [5]).

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 7) or measurement of emanations (Number 5 in Figure 7) 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 (via Section 3.2 of [5]).

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 7). 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 7). 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 (via Section 3.2 of [5]).

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 of the Composite TOE 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 7).

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 this, 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 (via Section 3.2 of [5]).

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 of the Composite TOE 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 7) and IC reverse engineering efforts (Number 3 in Figure 7). 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 is required to gather significant knowledge about the TOE's internal construction here (Number 3 in Figure 7).

The TOE shall avert the threat "Forced Information Leakage (T.Leak-Forced)" as specified below (via Section 3.2 of [5]).

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 of the Composite TOE 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 7) which normally do not contain significant information about secrets.

The TOE shall avert the threat "Abuse of Functionality (T.Abuse-Func)" as specified below (via Section 3.2 of [5]).

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 of the Composite TOE, (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 of the Composite TOE 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 (via Section 3.2 of [5]).

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 random numbers produced by the TOE security service. Because unpredictability is the main property of random numbers this may be a problem in case they are used to generate cryptographic keys. The entropy provided by the random numbers must be appropriate for the strength of the cryptographic algorithm, the key or the cryptographic variable is used for. Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the TOE. 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 user data of the Composite TOE. 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.

3.2.4 Threats related to Authentication of the Security IC

The TOE shall avert the threat "Masquerade the TOE (T. Masquerade_TOE)" as specified below (via Section 7.2.1 of [5]).

T.Masquerade_TOE Masquerade the TOE

An attacker may threaten the property being a genuine TOE by producing a chip which is not a genuine TOE but wrongly identifying itself as genuine TOE sample.

Note: This element is related to TOE authentication which are intended to be used in secured environments (MSSR audited Samsung sites and IC Embedded S/W developer sites) up to Phase 6. However, this ST claim conformance to the Package "Authentication of the Security IC" up to Phase 6 only as no authentication mechanism can be provided after the bootloader is locked. After locking of the bootloader, the related threats and objectives for the operational environment and SFRs related to the TOE authentication are regarded as not applicable (out of scope of the intended use-case), as the authentication functionality is no longer available.

Composite products can implement authentication functionality on their own. In this case all elements related to the TOE authentication are still applicable. However, this is outside of the scope of the evaluation of this TOE.

3.3 Organizational Security Policies

The following Figure 8 shows the policies applied in this Security Target.

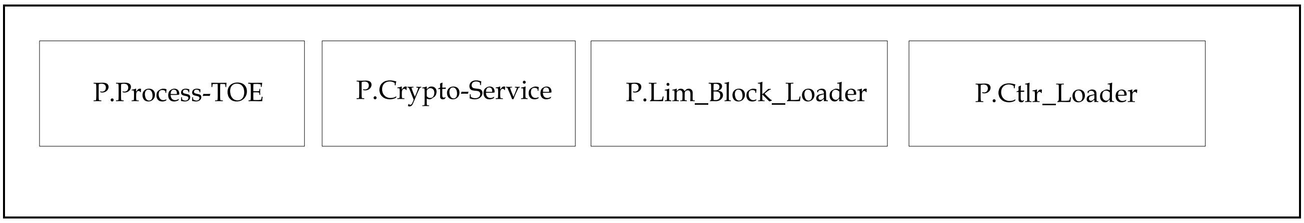


Figure 8: Policies

The following explanations help to understand the focus of additional organization security policies, which also addresses Application Note 5 mentioned in [5].

- The organizational security policy P.Crypto-Services applies to "Cryptographic Support" provide by the TOE for Security IC Embedded Software.
- The organisational security policy "Limiting and Blocking the Loader Functionality (P.Lim_Block_Loader)" applies to Loader dedicated for usage in secured environment.

The IC Developer / Manufacturer must apply the policy "Protection during TOE Development and Production (P.Process-TOE)" as specified below (via Section 3.3 of [5]).

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 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 "Cryptographic Service (P.Crypto-Service)" as specified below (via Section 7.4 of [5]).

P.Crypto-Service Cryptographic Services provided by the TOE

The TOE shall provide the following cryptographic services to the IC Embedded Software:

- Triple Data Encryption Standard (TDES)
- Advanced Encryption Standard (AES)

The IC Developer / Manufacturer must apply the organisational security policy "Limiting and Blocking the Loader Functionality (P.Lim_Block_Loader)" applies to Loader dedicated for usage in secured environment specified below (via Section 7.3.1 of [5]).

P.Lim_Block_Loader Limiting and Blocking the Loader Functionality

The composite manufacturer uses the Loader for loading of Security IC Embedded Software, user data of the Composite Product or IC Dedicated Support Software in charge of the IC Manufacturer. He limits the capability and blocks the availability of the Loader in order to protect stored data from disclosure and manipulation.

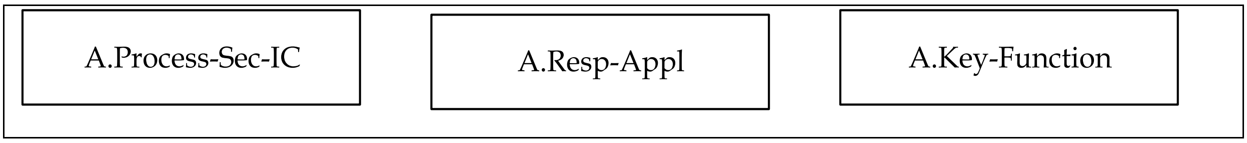
The organizational security policy "Controlled usage to Loader Functionality (P.Ctlr_Loader)" applies to Loader dedicated for usage by authorized users only.

P.Ctlr_Loader Controlled usage to Loader Functionality

Authorized user controls the usage of the Loader functionality in order to protect stored and loaded user data from disclosure and manipulation.

3.4 Assumptions

The following Figure 7 shows the assumptions applied in this Security Target.



A.Process-Sec-IC

A.Resp-Appl

A.Key-Function

Figure 9: 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 (via Section 3.4 of [5]).

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 "Treatment of User Data (A.Resp-Appl)" while developing this software in Phase 1 as specified below (via Section 3.4 of [5]).

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 can not 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 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 of the Composite TOE 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 [5] there are the following standard high-level security goals:

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

SG2 maintain the confidentiality of User Data (when being processed and when being stored in the TOE's memories).

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

Note, the Security IC may not distinguish between user data which are public known or kept confidential. Therefore the security IC shall protect the user data in integrity and in confidentiality if stored in protected memory areas, unless the Security IC Embedded Software chooses to disclose or modify it. Parts of the Security IC Embedded Software which do not contain secret data or security critical source code, may not require protection from being disclosed. Other parts of the Security IC Embedded Software may need kept confidential since specific implementation details may assist an attacker.

These standard high-level security goals in the context of the security problem definition build the starting point for the definition of security objectives as required by the Common Criteria (refer to Figure 10). Note that the integrity of the TOE is a means to reach these objectives.

According to the protection profile there is the following high-level security goal related to specific functionality:

SG4 provide true random numbers.

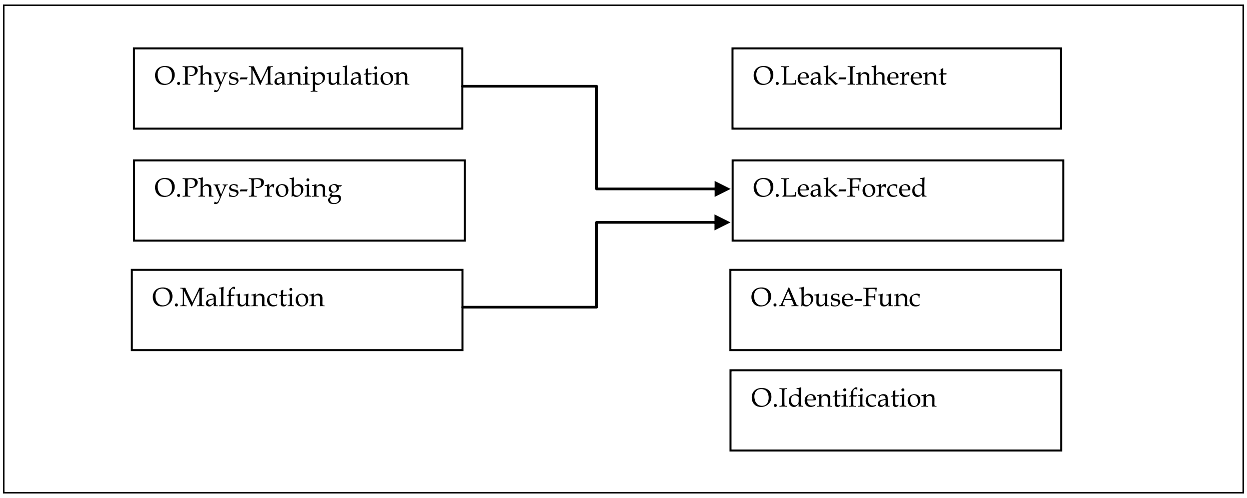


Figure 10: Standard Security Objectives

The additional high-level security considerations are refined below by defining security objectives as required by the Common Criteria (refer to Figure 11).

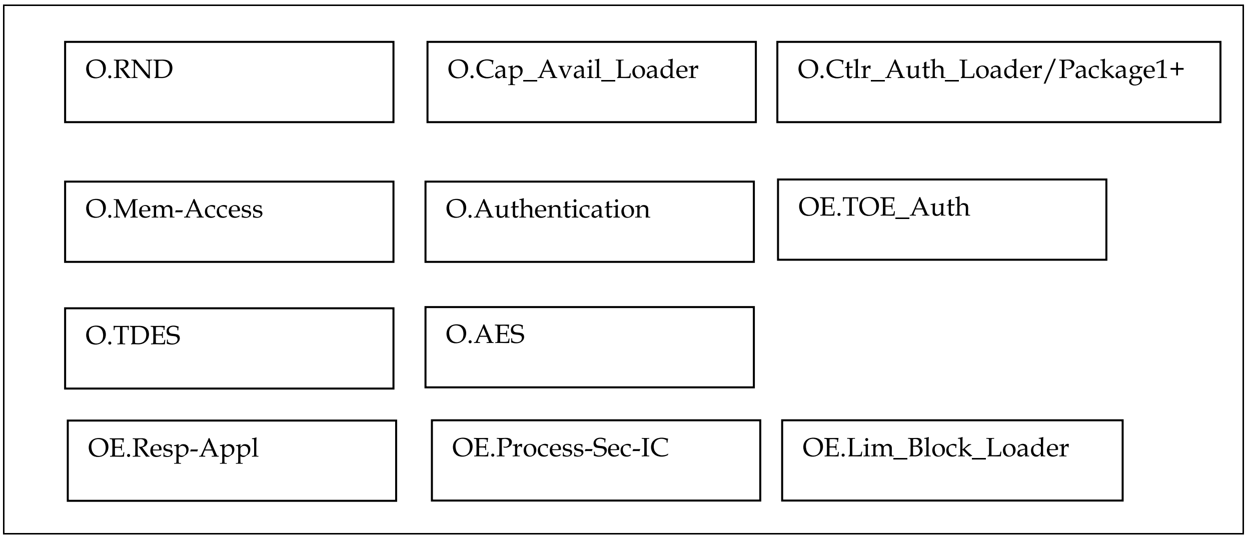


Figure 11: Security Objectives related to Specific Functionality

Standard Security Objectives

The TOE shall provide "Protection against Inherent Information Leakage (O.Leak-Inherent)" as specified below (via Section 4.1 of [5]).

O.Leak-Inherent Protection against Inherent Information Leakage

The TOE must provide protection against disclosure of confidential data stored and/or processed in the Security 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 (via Section 4.1 of [5]).

O.Phys-Probing Protection against Physical Probing

The TOE must provide protection against the disclosure/reconstruction of user data while stored in protected memory areas and processed or against the disclosure of other critical information about the operation of the TOE. 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 (via Section 4.1 of [5]).

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 malfunctions. 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 (via Section 4.1 of [5]).

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
- undetected manipulation of memory contents.

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 (via Section 4.1 of [5]).

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 againstPhysical 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 (via Section 4.1 of [5]).

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 of the Composite TOE, (ii) to manipulate critical User Data of the Composite TOE, (iii) to manipulate Security IC 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 (via Section 4.1 of [5]).

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 SG4)

The TOE shall provide "Random Numbers (O.RND)" as specified below (via Section 4.1 of [5]).

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 O.Leak-Inherent 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 "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.

Security Objectives for Loader

The TOE shall provide " Capability and availability of the Loader (O.Cap_Avail_Loader)" as specified below (via Section 7.3.1 of [5]).

O. Cap_Avail_Loader	Capability and availability of the Loader The TSF provides limited capability of the Loader functionality and irreversible termination of the Loader in order to protect stored user data from disclosure and manipulation.

Security Objectives for Cryptographic Services

The TOE shall provide " Cryptographic service Triple-DES (O.TDES)" as specified below (via Section 7.4.1 of [5]).

O.TDES	Cryptographic service Triple-DES The TOE provides secure hardware based cryptographic services implementing the Triple-DES for encryption and decryption.

The TOE shall provide "Cryptographic service AES (O.AES)" as specified below (via Section 7.4.2 of [5]).

O.AES	Cryptographic service AES The TOE provides secure hardware based cryptographic services for the AES for encryption and decryption.

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.

The Security IC Embedded Software shall provide "Treatment of user data of the Composite TOE (OE.Resp-Appl)" as specified below (via Section 4.2 of [5]).

OE.Resp-Appl	Treatment of user data of the Composite TOE Security relevant user data of the Composite TOE (especially cryptographic keys) are treated by the Security IC Embedded Software as required by the security needs of the specific application context.

For example the Security IC Embedded Software will not disclose security relevant user data of the Composite TOE to unauthorised users or processes when communicating with a terminal.

The Security IC Embedded Software shall provide "Authentication to external entities (O.Authentication)" as specified below (via Section 7.2.1 of [5]).

O. Authentication Authentication to external entities

The TOE shall be able to authenticate itself to external entities. The Initialisation Data (or parts of them) are used for TOE authentication verification data.

Note: This element is related to TOE authentication which are intended to be used in secured environments (MSSR audited Samsung sites and IC Embedded S/W developer sites) up to Phase 6. However, this ST claim conformance to the Package "Authentication of the Security IC" up to Phase 6 only as no authentication mechanism can be provided after the bootloader is locked. After locking of the bootloader, the related threats and objectives for the operational environment and SFRs related to the TOE authentication are regarded as not applicable (out of scope of the intended use-case), as the authentication functionality is no longer available.

Composite products can implement authentication functionality on their own. In this case all elements related to the TOE authentication are still applicable. However, this is outside of the scope of the evaluation of this TOE.

4.2.1 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. For example, it must be ensured that it is beyond practicality to derive the private key from a public key if asymmetric algorithms are used. 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 (via Section 4.3 of [5]).

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.

The operational environment of the TOE shall provide "Limitation of capability and blocking the Loader (OE.Lim_Block_Loader)" as specified below (via Section 7.3.1 of [5]).

OE.Lim_Block_Loader Limitation of capability and blocking the Loader

Authorized user will limit the capability of the Loader before the TOE is delivered to unauthorized user and terminate irreversibly the Loader after intended usage.

The TOE shall provide "Access control and authenticity for the Loader (O.Ctrl_Auth_Loader/Package1+)" as specified below (via Section 3 of [17]).

O.Ctrl_Auth_Loader/Package1+ Access control and authenticity for the Loader

The TSF provides communication channel with authorized user and access control for usage of the Loader functionality.

The operational environment of the TOE shall provide "Secure usage of the Loader (OE.Loader_Usage/Package1+)" as specified below (via Section 3 of [17]).

OE.Loader_Usage/Package1+ Secure usage of the Loader

The authorized user must fulfill the access conditions required by the Loader

The operational environment shall provide "External entities authenticating of the TOE (OE.TOE_Auth)" as specified below (via Section 7.2.1 of [5]).

OE.TOE_Auth External entities authenticating of the TOE

Note: This element is related to TOE authentication which are intended to be used in secured environments (MSSR audited Samsung sites and IC Embedded S/W developer sites) up to Phase 6. However, this ST claim conformance to the Package "Authentication of the Security IC" up to Phase 6 only as no authentication mechanism can be provided after the bootloader is locked. After locking of the bootloader, the related threats and objectives for the operational environment and SFRs related to the TOE authentication are regarded as not applicable (out of scope of the intended use-case), as the authentication functionality is no longer available.

Composite products can implement authentication functionality on their own. In this case all elements related to the TOE authentication are still applicable. However, this is outside of the scope of the evaluation of this TOE.

The operational environment shall support the authentication verification mechanism and know authentication reference data of the TOE.

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 2 below gives an overview, how the assumptions, threats, and organizational 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.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.Crypto-Service	O.TDES O.AES	
A.Key-Function	OE.Resp-Appl	
P.Lim_Block_Loader	O.Cap_Avail_Loader OE.Lim_Block_Loader	Phase 5
T.Masquerade_TOE	O.Authentication OE.TOE_Auth	
P.Ctlr_Loader	O.Ctlr_Auth_Loader/Package1+ OE.Loader_Usage/Package1+	Phase 4 - 6

Table 3: Security Objectives versus Assumptions, Threats or Policies

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 of the composite TOE can be prevented (refer to T.Mem-Access). The threat T.Mem-Access is therefore removed if the objective is met.

The clarification of O.Mem-Access makes clear that it is up to the Smartcard Embedded Software to implement the memory management scheme by appropriately administrating the TSF. 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. .

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.

The organisational security policy Limitation of capability and blocking the Loader (P.Lim_Block_Loader) is directly implemented by the security objective for the TOE "Capability and availability of the Loader (O.Cap_Avail_Loader)" and the security objective for the TOE environment "Limitation of capability and blocking the Loader (OE.Lim_Block_Loader)". The TOE security objective "Capability and availability of the Loader" (O.Cap_Avail_Loader)" mitigates also the threat "Abuse of Functionality " (T.Abuse-Func) if attacker tries to misuse the Loader functionality in order to manipulate security services of the TOE provided or depending on IC Dedicated Support Software or user data of the TOE as IC Embedded Software, TSF data or user data of the smartcard product.

The threat "Masquerade the TOE (T.Masquerade_TOE)" is directly covered by the TOE security objective "Authentication to external entities (O.Authentication)" describing the proving part of the authentication and the security objective for the operational environment of the TOE "External entities authenticating of the TOE (OE.TOE_Auth)" the verifying part of the authentication.

The organisational security policy "Controlled usage to Loader Functionality (P.Ctlr_Loader) is directly implemented by the security objective for the TOE "Access control and authenticity for the Loader (O.Ctrl_Auth_Loader/Package1+)" and the security objective for the TOE environment "Secure of the Loader (OE.Loader_Usage)/Package1+".

5 EXTENDED COMPONENTS DEFINITION

This Security Target does not define extended components. Note that the PP “Security IC Protection Profile” defines extended security functional requirements, which are included in this Security Target.

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

5.4 Definition of the Family FDP_SDC

5.5 Definition of the Family FIA_API

5.1 Definition of the Family FCS_RNG

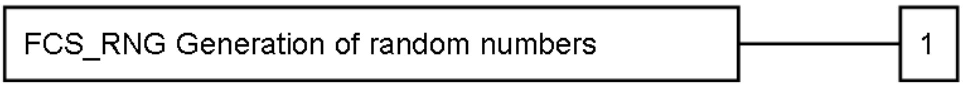
To define the IT security functional requirements of the TOE, an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

FCS_RNG Generation of Random Numbers

Family behaviour

This family defines quality requirements for the generation of random numbers which are intended to be used for cryptographic purposes.

Component levelling:



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

FCS_RNG.1/PTG.2 Random number generation (AIS31 PTG.2 class as defined in [5, 6])

Hierarchical to: No other components.

FCS_RNG.1.1/PTG.2 The TSF shall provide a physical random number generator that implements:

(PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.

(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG [selection: *prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates DRG.2 as long as its internal state entropy guarantees the claimed output entropy*].

(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.

(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.

(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered [selection: *externally, at regular intervals, continuously, applied upon specified internal events*]. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time

FCS_RNG.1.2/PTG.2 The TSF shall provide [selection: *bits, octets of bits, numbers* [assignment: *format of the numbers*]] that meet:

(PTG.2.6) Test procedure A [assignment: *additional standard test suites*] does not distinguish the internal random numbers from output sequences of an ideal RNG.

(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997

Dependencies: No dependencies

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 behavior

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 Limited capabilities and availability
1
2

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 same policy or two mutual supportive policies related to the same functionality. This allows e.g. 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.

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.

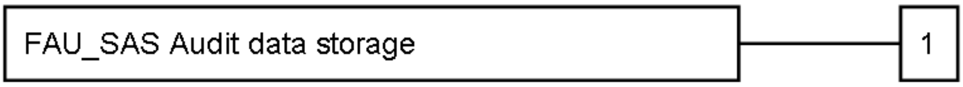
The family "Audit data storage (FAU_SAS)" is specified as follows.

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.

5.4 Definition of the Family FDP_SDC

To define the security functional requirements of the TOE an additional family (FDP_SDC.1) of the Class FDP (User data protection) is defined here.

The family "Stored data confidentiality (FDP_SDC)" is specified as follows.

FDP_SDC Stored data confidentiality

Family behavior

This family provides requirements that address protection of user data confidentiality while these data are stored within memory areas protected by the TSF. The TSF provides access to the data in the memory through the specified interfaces only and prevents compromise of their information bypassing these interfaces. It complements the family "Stored data integrity (FDP_SDI)" which protects the user data from integrity errors while being stored in the memory.

Component leveling

FDP_SDC.1 Requires the TOE to protect the confidentiality of information of the user data in specified memory areas.

Management: FDP_SDC.1.

There are no management activities foreseen.

Audit: FDP_SDC.1

There are no actions defined to be auditable.

FDP_SDC.1 Stored data confidentiality

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDC.1.1 The TSF shall ensure the confidentiality of the information of the user data while it is stored in the [*assignment: memory area*]

5.5 Definition of the Family FIA_API

To describe the IT security functional requirements of the TOE a functional family FIA_API (Authentication Proof of Identity) of the Class FIA (Identification and authentication) is defined here. This family describes the functional requirements for the proof of the claimed identity by the TOE and enables the authentication verification by an external entity. The other families of the class FIA address the verification of the identity of an external entity by the TOE.

The other families of the Class FIA describe only the authentication verification of users' identity performed by the TOE and do not describe the functionality of the user to prove their identity. The following paragraph defines the family FIA_API in the style of the Common Criteria part 2 (cf. [3], chapter "Extended components definition (APE_ECD)") from a TOE point of view.

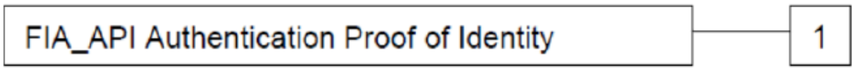
The family "Authentication Proof of Identity (FIA_API)" is specified as follows.

FIA_API.1 Authentication Proof of Identity

Family behavior

This family defines functions provided by the TOE to prove its identity and to be verified by an external entity in the TOE IT environment.

Component leveling



FIA_API.1 Authentication Proof of Identity, provides proof of the identity of the TOE, an object or an authorized user or role to an external entity.

Management: FIA_API.1

The following actions could be considered for the management functions in FMT: Management of authentication information used to prove the claimed identity.

Audit: FIA_API.1

There are no actions defined to be auditable.

FIA_API.1 **Authentication Proof of Identity**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1 The TSF shall provide a [*assignment: authentication mechanism*] to prove the identity of the [*selection: TOE*, [*assignment: object, authorized user or role*]] to an external entity.

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 detectors that detect out of the specified range. If the failures are happen, the TOE goes into secure 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 and implemented 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 of the Composite TOE 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 and implemented 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 of the Composite TOE 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 Prepersonalisation Data and supplements of the Smartcard Embedded Software in *the 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 "Stored data confidentiality (FDP_SDC.1)" as specified below.

FDP_SDC.1 Stored data confidentiality

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDC.1.1 The TSF shall ensure the confidentiality of the information of the user data while it is stored in the *FLASH, RAM and ROM*.

The TOE shall meet the requirement "Stored data integrity monitoring and action (FDP_SDI.2)" as specified below.

FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for *error* on all objects, based on the following attributes: *FLASH, RAM and ROM read operation*.

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall enforce *either a device RESET or an interrupt which is configurable by Security IC Embedded Software*.

Application Note: This requirement is achieved by security features such internal encryption and scrambling mechanisms.

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 shield must be removed and bypassed in order to perform physical intrusive attacks. The TOE makes appropriate secure reaction to stop operation if a physical manipulation or physical probing attack is detected. And also internal scrambling & encryption for memories and logic area 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/Memory Subset access 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 FRO M)

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

FCS_RNG.1/PTG.2	Random number generation
Hierarchical to:	No other components.
FCS_RNG.1.1/PTG.2	The TSF shall provide a *physical true* random number generator that implements:
FCS_RNG.1.2/PTG.2	The TSF shall provide *numbers, in sets of 16 bits* that meet:

Application Note: The DTRNG FRO M library comprises some functions that performs statistical test on the DTRNG output. If either test fails, the function returns an error value and the DTRNG should be turned off. Those functions are described in DTRNG FRO M Application note in detail and are available to embedded software.

Additional standard test suites in PTG.2.6 are left empty on purpose.

Dependencies: No dependencies.

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 this, 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/Memory)"** 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/Memory)"** 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)":

Memory Access Control Policy

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

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

FDP_ACC.1/Memory Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1/Memory The TSF shall enforce the *Memory Access Control Policy* on *all subjects (software with privileged mode and user mode), all objects (data including code stored in memories) and all the operations defined in the Memory Access Control Policy*.

Dependencies: FDP_ACF.1 Security attribute based access control

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

FDP_ACF.1/Memory Security attribute based access control

The attributes are all the operations related to the data stored in memories, which are the *read, write* and *execute* operations.

Hierarchical to: No other components.

FDP_ACF.1.1/Memory The TSF shall enforce the *Memory Access Control Policy* to objects, which include *registers* where the values are *read from or written to*, by subjects, which include *software running in privileged mode and user mode*.

FDP_ACF.1.2/Memory The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed

- Automatically evaluate the corresponding permission control information as soon as the access is attempted so that the action to be denied cannot be realized by the subject attempting to perform the operation.

FDP_ACF.1.3/Memory The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *none*.

FDP_ACF.1.4/Memory The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*.

Dependencies: FDP_ACC.1/Memory 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 *software running in privileged mode*.

Dependencies: FDP_ACC.1/Memory Subset access 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 (TDES) with in ECB mode,
- Advanced Encryption Standard (AES) with in ECB mode.
- Cryptographic key destruction.

Triple-DES Operation

The Triple DES (TDES) operation of the TOE shall meet the requirement "Cryptographic operation (FCS_COP.1/TDES)" compliant with complete TDES package defined in [5] as specified below.

FCS_COP.1/TDES Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1/TDES The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *TDES in 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/TDES Cryptographic key destruction

AES Operation

The AES operation of the TOE shall meet the requirement "Cryptographic operation (FCS_COP.1/AES)" compliant with complete AES package defined in [5] 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 *AES in ECB mode* and cryptographic key sizes *128 bit, 192 bit, 256 bit* that meet the following *FIPS-197 [11], chapter 5 and [SP800-38A].*

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/AES Cryptographic key destruction

Cryptographic Key Destruction

The cryptographic key destruction of the TOE shall meet the requirement "Cryptographic Key Destruction (FCS_CKM.4/TDES)" as specified below.

FCS_CKM.4/TDES Cryptographic key destruction (TDES)

Hierarchical to: No other components.

FCS_CKM.4.1/TDES The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *zeroization* that meets the following: *none.*

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]

Application Note: The cryptographic key destruction may be provided by overwriting the internal stored key when a new key value is provided through the key interface or a key zeroization initiated by special signal.

The cryptographic key destruction of the TOE shall meet the requirement "Cryptographic Key Destruction (FCS_CKM.4/AES)" as specified below.

FCS_CKM.4/AES Cryptographic key destruction (AES)

Hierarchical to: No other components.

FCS_CKM.4.1/AES The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *zeroization* that meets the following: *none.*

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]

Application Note: The cryptographic key destruction may be provided by overwriting the internal stored key when a new key value is provided through the key interface or a key zeroization initiated by special signal..

Bootloader Access Control

Before being deployed in the field, a Smartcard device is required to be programmed appropriately via the TOE's Bootloader software. This shall be done in a controlled manner in order to preserve the authenticity and confidentiality of user data in the Composite TOE. To support this, the TOE provides so-called **Loader Access Control**.

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

Loader Access Control Policy

The TOE shall control accesses of external parties to restricted Bootloader functions via a secret-based mutual authentication procedure in a secure environment (that is, certified sites). In other words, the TOE and an external party involve in a session of exchanging information and at the end of the session, either both parties are convinced that the other indeed holds a shared secret key, in which case authentication is deemed successful, or not at all, in which case authentication is deemed failed. The TOE shall grant the access to restricted Bootloader functions if and only if mutual authentication is successful. Otherwise, access to restricted Bootloader functions is denied.

Subjects are *authorized users for using Loader*. Objects are *user data of the Composite TOE stored in FLASH and ROM memories*. Restricted Bootloader functions (controlled operations) include:

- *FLASH programming,*
- *Changing booting mode from ROM to FLASH.*

After flash programming is completed and booting mode is switched to FLASH-booting, any access to Bootloader functionalities shall be locked.

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

FMT_LIM.1/Loader Limited capabilities

Hierarchical to: No other components.

FMT_LIM.1.1/Loader 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: Deploying Loader functionality after *authentication being done successfully* does not allow stored user data to be disclosed or manipulated by unauthorized user.

Dependencies: FMT_LIM.2 Limited availability.

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

FMT_LIM.2/Loader Limited availability - Loader

Hierarchical to: No other components.

FMT_LIM.2.1/Loader The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: The TSF prevents deploying the Loader functionality after *locking the TOE to Flash booting mode in hardware.*

Dependencies: FMT_LIM.1 Limited capabilities.

Note: Bootloader is optionally implemented. In case IC Embedded Software is stored User ROM, replacing Bootloader code, Bootloader is not available. Note that, in this case the SFR is also met because Bootloader is intrinsically not available.

The TOE Functional Requirement "Subset access control - Loader (FDP_ACC.1/Loader)" is specified as follows.

FDP_ACC.1/ Loader Subset access control - Loader

Hierarchical to: No other components.

FDP_ACC.1.1/ Loader The TSF shall enforce the Loader SFP on

(1) the subjects Authentication Sequence and Flash Attribute control APDU command,

(2) the objects user data in FLASH or ROM

(3) the operation deployment of Loader

Dependencies: FDP_ACF.1/Loader Security attribute based access control.

Application Note 38: The TOE enforces the Loader SFP by FDP_ACF.1/Loader to describe additional access control rules. FDP_ACF.1/Loader is part of Loader Package 1+ which is defined in [17] as an addition to the package definition of the PP [5].

The TOE Functional Requirement "Security attribute based access control - Loader (FDP_ACF.1/Loader)" is specified as follows.

FDP_ACF.1/ Loader Security attribute based access control - Loader

Hierarchical to: No other components.

FDP_ACF.1.1/Loader FDP_ACF.1.1 The TSF shall enforce the Loader SFP to objects based on the following:

(1) the subjects Bootloader with security attributes FLASH write.

(2) the objects user data in FLASH with security attributes FLASH write.

FDP_ACF.1.2/ Loader FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: Bootloader can do write operation in FLASH after succession of Authentication.

FDP_ACF.1.3/ Loader FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: FLASH can be controlled based on security attributes, which can be limited by Bootloader APDU command.

FDP_ACF.1.4/ Loader The TSF shall explicitly deny access of subjects to objects based on the following additional rules: Bootloader can't access the FLASH without explicit success of Authentication.

Dependencies: FMT_MSA.3 Static attribute initialization.

Authentication Proof of Identity

The TOE shall meet the requirement "Authentication Proof of Identity (FIA_API.1)" as specified below.

FIA_API.1 Authentication Proof of Identity

Hierarchical to: No other components

Dependencies: No dependencies.

FIA_API.1.1 The TSF shall provide *a challenge-response-based authentication mechanism with AES block cipher and a shared secret key* (*Section B-2.2 of [18]*) to prove the identity of *the TOE* to an external entity.

Note: This element is related to TOE authentication which are intended to be used in secured environments (MSSR audited Samsung sites and IC Embedded

S/W developer sites) up to Phase 6. However, this ST claim conformance to the Package "Authentication of the Security IC" up to Phase 6 only as no authentication mechanism can be provided after the bootloader is locked. After locking of the bootloader, the related threats and objectives for the operational environment and SFRs related to the TOE authentication are regarded as not applicable (out of scope of the intended use-case), as the authentication functionality is no longer available.

Composite products can implement authentication functionality on their own. In this case all elements related to the TOE authentication are still applicable. However, this is outside of the scope of the evaluation of this TOE.

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)
Stored data confidentiality (FDP_SDC.1)
Stored data integrity monitoring and action (FDP_SDI.2)
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)
Authentication Proof of Identity (FIA_API.1)
Quality metric for random numbers (FCS_RNG.1/PTG.2)
Limited capabilities(FMT_LIM.1/Loader)
Limited availability - Loader(FMT_LIM.2/Loader)

Table 4. Security Functional Requirements defined in Smart Card IC Protection Profile

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

Cryptographic key destruction – Triple-DES (FCS_CKM.4/TDES)
Cryptographic key destruction – AES (FCS_CKM.4/AES)

Table 5. 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 6 (EAL6)

and augmented by the following components

ASE_TSS.2

corresponding to level "***EAL6+***".

All refinements from Protection Profile BSI-PP-0084 version 1.0 for the assurance requirements (ALC_DEL, ALC_DVS, ALC_CMS, ALC_CMC, ADV_ARC, ADV_FSP, ADV_IMP, ATE_COV, AGD_OPE, AGD_PRE and AVA_VAN) have to be taken into consideration. In particular the document [12] as recommended by SOG-IS is used in the context of vulnerability analysis (AVA_VAN.5).

Class ADV: Development

Architectural design	(ADV_ARC.1)	(EAL6 and PP)
Security Policy Model	(ADV_SPM.1)	(EAL6)
Functional Specification	(ADV_FSP.5)	(EAL6)
Implementation Representation	(ADV_IMP.2)	(EAL6)
TSF Internals	(ADV_INT.3)	(EAL6)
TOE Design	(ADV_TDS.5)	(EAL6)

Class AGD: Guidance documents activities

Operational User Guidance	(AGD_OPE.1)	(EAL6 and PP)
Preparative procedures	(AGD_PRE.1)	(EAL6 and PP)

Class ALC: Life-cycle support

CM Capabilities	(ALC_CMC.5)	(EAL6)
CM Scope	(ALC_CMS.5)	(EAL6)
Delivery	(ALC_DEL.1)	(EAL6 and PP)
Development Security	(ALC_DVS.2)	(EAL6 and PP)
Life Cycle Definition	(ALC_LCD.1)	(EAL6 and PP)
Tools and Techniques	(ALC_TAT.3)	(EAL6)

Class ASE: Security Target evaluation

Conformance claims	(ASE_CCL.1)	(EAL6 and PP)
Extended components definition	(ASE_ECD.1)	(EAL6 and PP)
ST introduction	(ASE_INT.1)	(EAL6 and PP)
Security objectives	(ASE_OBJ.2)	(EAL6 and PP)
Derived security requirements	(ASE_REQ.2)	(EAL6 and PP)
Security problem definition	(ASE_SPD.1)	(EAL6 and PP)
TOE summary specification	(ASE_TSS.2)	(ST)

Class ATE: Tests			
	Coverage	(ATE_COV.3)	(EAL6)
	Depth	(ATE_DPT.3)	(EAL6)
	Functional Tests	(ATE_FUN.1)	(EAL6 and PP)
	Independent Testing	(ATE_IND.2)	(EAL6 and PP)
Class AVA: Vulnerability assessment			
	Vulnerability Analysis	(AVA_VAN.5)	(EAL6 and PP)

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	- FDP_SDC.1 "Stored data confidentiality" - 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	- FDP_SDI.2 "Stored data integrity monitoring and action" - FPT_PHP.3 "Resistance to physical attack"
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,FDP_SDI.2
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, AVA_VAN.5, FDP_SDI.2, FDP_SDC.1
O.Identification	- FAU_SAS.1 "Audit storage"

Objective	TOE Security Functional and Assurance Requirements
O.RND	- FCS_RNG.1/PTG.2 "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, FDP_SDI.2 ,FDP_SDC.1
OE.Resp-Appl	not applicable
OE.Process-Sec-IC	not applicable
O.Mem-Access	- FDP_ACC.1/Memory "Subset access control" - FDP_ACF.1/Memory "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"
O.TDES	- FCS_COP.1/TDES - FCS_CKM.4/TDES
O.AES	- FCS_COP.1/AES - FCS_CKM.4/AES
O.Authentication	- FIA_API.1 " Authentication Proof of Identity"
OE.TOE_Auth	not applicable
O.Cap_Avail_Loader	- FMT_LIM.1/Loader "Limited capabilities - Loader" - FMT_LIM.2/Loader "Limited availability – Loader"
OE.Lim_Block_Loader	not applicable
O.Ctrl_Auth_Loader/Package1+	- FDP_ACC.1/Loader "Subset access control - Loader" - FDP_ACF.1/Loader "Security attribute based access control - Loader"
OE.Loader_Usage/Package1+	not applicable

Table 6: 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.

It is possible that the TOE needs additional support by the Security IC Embedded Software (e.g. timing attacks are possible if the processing time of algorithms implemented in the software depends on the content of secret). This support must be addressed in the Guidance Documentation. Together with this FPT_ITT.1, FDP_ITT.1 and FDP_IFC.1 are suitable to meet the objective.

The justification related to the security objective "Protection against Physical Probing (O.Phys-Probing)" is as follows: The SFR FDP_SDC.1 requires the TSF to protect the confidentiality of the information of the user data stored in specified memory areas and prevent its compromise by physical attacks bypassing the specified interfaces for memory access. 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. The functions implementing FRU_FLT.2 and FPT_FLS.1 must work independently so that their operation cannot be affected 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 justification related to the security objective "Protection against Physical Manipulation (O.Phys-Manipulation)" is as follows: The SFR FDP_SDI.2 requires the TSF to detect the integrity errors of the stored user data and react in case of detected errors. 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, refer to Section 6.1). 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 behavior 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 fulfill 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 recognize 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 Initialization Data (or parts of them) are used for TOE identification. The technical capability of the TOE to store Initialization Data and/or Pre-personalization 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 organizational 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 and ALC.

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 unauthorized 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 security objective "Capability and availability of the Loader (O.Cap_Avail_Loader) is directly covered by the SFR FMT_LIM.1/Loader and FMT_LIM.2/Loader.

The security objective Access control and authenticity for the Loader (O.Ctrl_Auth_Loader/Package1+) is only partly covered by SFRs. The objective requires (1) authenticity of the user data and (2) access control for the usage of the loader functionality. The second item is addressed by FDP_ACC.1/Loader and FDP_ACF.1/Loader which defines the Loader Policy. The first item, i.e., authentication of user data is addressed as follows: Samsung and his customer agree on a shared secret for a secure usage of the Boot Loader (see option 3 in Section 1.2.4). If the customer entrusts a third party to use the Loader, he will have to share this secret with this third party as well. To maintain the authenticity of the user data to be loaded to the TOE, this third party will have to be audited and certified (MSSR) including the following items: (1) handling of security-relevant assets such as the shared secret agreed between Samsung and his customer, and (2) to ensure that the user data to be loaded to the TOE are not altered by this third party.

The FCS_COP.1/TDES meets the security objective "Cryptographic service Triple-DES (O.TDES)".

The FCS_COP.1/AES meets the security objective "Cryptographic service AES (O.AES)".

The security objective "Authentication to external entities (O.Authentication) is directly covered by the SFR FIA_API.1.

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/Memory)" 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/Memory with its SFP is suitable to meet the security objective.

The security functional requirement "Static attribute initialization (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 realized 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 "Protection during Packaging, Finishing and Personalisation (OE.Process-Sec-IC)" is as follows: The Composite Product Manufacturer has to use adequate measures to fulfill 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 personalization functions.

6.3.2 Dependencies of Security Functional Requirements

Table 8 below lists the security functional requirements defined in the protection profile, their dependencies and whether they are satisfied by other security requirements defined in the 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
FDP_SDC.1	None	No dependency
FDP_SDI.2	None	No dependency
FPT_PHP.3	None	No dependency
FDP_ITT.1	FDP_ACC.1/Memory	Yes
FDP_IFC.1	FDP_IFF.1	See discussion below
FPT_ITT.1	None	No dependency
FCS_RNG.1/PTG.2	None	No dependency
FCS_COP.1/TDES	FCS_CKM.4/TDES	Yes (by environment, see discussion below)
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1/TDES) or FCS_CKM.1/TDES	Yes (by environment, see discussion below)
FCS_COP.1/AES	FCS_CKM.4/AES	Yes (by environment, see discussion below)
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1/AES) or FCS_CKM.1/AES	Yes (by environment, see discussion below)

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FCS_CKM.4/TDES	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1/TDES) or FCS_CKM.1/TDES	Yes (by Security IC Emedded Software with the hardware TOE providing the interface)
FCS_CKM.4/AES	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1/AES) or FCS_CKM.1/AES	Yes (by Security IC Emedded Software with the hardware TOE providing the interface)
FDP_ACC.1/Memory	FDP_ACF.1/Memory	Yes
FDP_ACF.1/Memory	FDP_ACC.1/Memory FMT_MSA.3	Yes Yes
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Yes See discussion below
FMT_MSA.1	FDP_ACC.1/Memory FMT_SMR.1 FMT_SMF.1	Yes See discussion below Yes
FMT_SMF.1	None	No dependency
FMT_LIM.1/Loader	FMT_LIM.2	Yes
FMT_LIM.2/Loader	FMT_LIM.1	Yes
FDP_ACC.1/ Loader	FDP_ACF.1/Loader	Yes
FDP_ACF.1/ Loader	FMT_MSA.3	Yes
FIA_API.1	None	No dependency

Table 7: 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/PTG.2) 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 requirement FCS_CKM.1 which is dependent to FCS_COP.1/TDES and FCS_COP.1/AES is not included in this Security Target since the TOE only provides an engine for encryption and decryption. The functional requirement and FCS_CKM.4 which is also dependent of FCS_COP.1/TDES and FCS_COP.1/AES is included in this Security Target but it is not directly implemented by the TOE. The TOE does provide the necessary interface for IC Embedded Software to perform key destruction operation including FLASH/SRAM erase and write operations. In conclusion, the dependent requirements of FCS_COP.1/TDES and FCS_COP.1/AES concerning these functions (FCS_CKM.1 and FCS_CKM.4) 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 EAL6 and the augmentation with the requirement ASE_TSS.2 were chosen to demonstrate that the TOE fulfills the high-level Common Criteria requirements. An assurance level of EAL6 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 defense against such attacks, the evaluators should have access to the low level design and all the source code.

In addition, the TOE security policy is formally described and its security objective i.e. the complete memory access control is formally proved. The ASE_TSS.2 was chosen to demonstrate further assurance extensions provided by the TOE.

ADV_SPM.1 Formal TOE Security Policy Model

ADV_SPM.1 Formal TOE Security Policy Model as defined in Section 12.5 of [3]

Dependencies: ADV_FSP.4 Complete functional specification

Developer action elements:

ADV_SPM.1.1D The developer shall provide a formal security policy model for the *following policies*:

- *Memory access control policy*
- *Loader access control policy*
- *Security detectors policy*
- *Non-reversibility of TEST mode*

ADV_SPM.1.2D For each policy covered by the formal security policy model, the model shall identify the relevant portions of the statement of SFRs that make up that policy.

ADV_SPM.1.3D The developer shall provide a formal proof of correspondence between the model and any formal specification.

ADV_SPM.1.4D The developer shall provide a demonstration of correspondence between the model and the functional specification.

Refinement

The following security policies and relevant security functional requirements (SFRs) are modeled by ADV_SPM.1:

- The limited availability and capabilities of Bootloader functions are correctly enforced. The relevant SFRs include FMT_LIM.1/Loader and FMT_LIM.2/Loader.
- The reaction to security detectors' events is correctly enforced. The relevant SFRs include FDP_SDI.2 and FDP_SDC.1.
- The access to the TOE's Test Mode shall be locked. The relevant SFRs include FMT_LIM.1 and FMT_LIM.2.
- The access control to the security registers, the special Flash, ROM regions and Booting memory area are correctly enforced. The relevant SFRs include FDP_ACC.1/Memory, FDP_ACF.1/Memory, FMT_MSA.3, FMT_MSA.1 and FMT_SMF.1.
- In addition, the access control with respect to the MPU settings is correctly enforced according to Chapter 9 of [22]. The relevant SFRs include FDP_ACC.1/Memory and FDP_ACF.1/Memory.
- The consistency of security attributes is correctly enforced i.e., memory regions and access rights are correctly initialized at the IC reset. The relevant SFRs include FMT_MSA.3, FMT_MSA.1 and FMT_SMF.1.

ASE_TSS.2 TOE Summary Specification with Architectural Design Summary

The augmentation ASE_TSS.2 is required in order to provide the potential users (e.g. the embedded software developers) with a succinct but comprehensive explanation on the TOE security functions that protect it against interference, logical tampering and bypass. This description is also necessary to establish the component ASE_TSS.2 for any composed TOE.

This assurance component is a higher hierarchical component to EAL6. ASE_TSS.2 has two dependencies (ASE_INT.1 and ASE_REQ.1) that both are satisfied by this 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 requirements FDP_SDC.1 and FDP_SDI.2 address the protection of user data in the specified memory areas against compromise and manipulation. The security functional requirement FPT_PHP.3 makes it harder to manipulate data. This protects the primary assets identified in Section 3.1 and other security features or functionality which use these data.

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.

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 vulnerable1, 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 functions1F, 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/Memory) and the security functional requirement defining the Memory Access Policy (FDP_ACF.1/Memory), 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 Life Cycle Detector is used to protect each detector. 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 events are recorded in the register which are notified by the detectors (refer to list below). The software can configure the reactions in case of detection:

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

List of detectors:

- Detectors

SFR2: FRU_FLT.2: Limited fault tolerance

All operating signals are filtered/regulated in order to prevent malfunction.

TOE's Filters

These filters are used for preventing abnormal environment conditions from causing undefined or unpredictable behaviors of the chip.

- Filters

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 S3FV9VH *User's Manual*. Therefore, FRU_FLT.2 is implemented by TOE.

Security domains are maintained since accesses to the access-prohibited area are trapped by this access control function.

SFR3: FPT_PHP.3: Resistance to physical attacks

This requirement is achieved by security feature as the shield must be removed and bypassed in order to perform physical intrusive attacks. The TOE shall generate a device reset or an interrupt to stop the device operation if a physical manipulation or physical probing attack is detected. And also scrambling and encryption mechanisms make reverse engineering of the TOE layout unpractical and protect from probing attack and signal identification of the TOE layout unpractical. So these functionalities meet the security functional requirement of FPT_PHP.3: Resistance to physical attack.

SFR4: FDP_ACC.1/Memory: Subset access control

This requirement is achieved by security register access control, invalid address access, access right for the code executed in FLASH (FDP_ACC.1/Memory).

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

This is covered by the Privileged and User modes of the TOE. (FDP_ACF.1/Memory)

The more information is on 1.2 TOE Overview and TOE 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.

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.

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.

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.

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.

SFR13: FDP_ITT.1: Basic internal transfer protection

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

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

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

SFR15: FCS_RNG.1/PTG.2: Random number generation

This requirement is ensured by a Digital True Random Number Generator (DTRNG FRO M) that follows the requirements of the *PTG.2* class as defined in [5, 6] for true random number generator.

SFR16A: FCS_COP.1/TDES: Cryptographic operation - TDES

This requirement is covered by the TOE.

> **1) Triple Data Encryption Standard Engine:** This function is used for encrypting and decrypting data using the Triple DES symmetric algorithm with 112-bit or 168-bit key sizes in ECB mode. (FCS_COP.1/TDES).

SFR16B: FCS_COP.1/AES: Cryptographic operation - AES

This requirement is covered by the TOE.

> **1) AES (Advanced Encryption Standard):** This function supports the AES operation with 128-bit, 192-bit and 256-bit key size in ECB mode (FCS_COP.1/AES).

SFR17: Reserved

SFR18: FMT_LIM.1/Loader : Limited capabilities

This requirement is achieved by Bootloader mutual authentication procedure. Only when authentication is completed successfully, the restricted Bootloader functions can be used by authorized users.

SFR19: FMT_LIM.2/Loader : Limited availability – Loader

This requirement is achieved by the changing the Operating Mode Selection from ROM Booting mode to Flash Booting mode and then locking.

SFR20: Reserved

SFR21: Reserved

SFR22: Reserved

SFR23: FDP_ACC.1/ Loader : Subset access control - Loader

> This requirement is achieved by following functions.
>
> ROM hiding function: The attribute of ROM is changed from Accessible ROM to Inaccessible ROM.
>
> Flash memory attribute as Read only.

SFR24: FDP_ACF.1/Loader: Security attribute based access control - Loader

> This is covered by the ROM Booting (ROM Reset) and Flash Booting (Flash Reset) mode of the TOE. TOE can be set to ROM Booting (ROM Reset) and FLASH Booting (FLASH Reset) mode domains exclusively. All Bootloader APDU commands are accessible only in Rom Booting mode. The Flash Booting mode cannot access all Bootloader APDU commands.

SFR25: FDP_SDC.1: Stored data confidentiality

This requirement is achieved by the combination of the TOE security features TOE features 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) 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 reverse-engineering and relevant analysis on the stored data as well as on internally transmitted data

4) Invalid address access: This function detects invalid address access occurrence

5) shield: This requirement is achieved by security feature as the shield must be removed and bypassed in order to perform physical intrusive attacks

6) Life cycle detector: Life cycle detector modifications.

7) Filters

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

9) Control of Booting mode: This requirement is achieved by changing the Operating Mode Selection.

SFR26: FDP_SDI.2: Stored data integrity monitoring and action

This requirement is achieved by following functions.

1) Flash/RAM: Error manages features.

SFR27: FIA_API.1: Authentication Proof of Identity

This requirement is achieved by processing the Authentication sequence.

SFR28A: FCS_CKM.4/TDES: Cryptographic Key Destruction - TDES

This requirement shall be covered by the Security IC Embedded Software by using the facility provided by the hardware TOE including the interface to erase / write to FLASH and SRAM.

1) Cryptographic key destruction for Triple Data Encryption Standard: This function is used for destructing cryptographic keys for Triple-DES (FCS_CKM.4/TDES)**.**

SFR28B: FCS_CKM.4/AES : Cryptographic Key Destruction - AES

This requirement shall be covered by the Security IC Embedded Software by using the facility provided by the hardware TOE including the interface to erase / write to FLASH and SRAM.

1) Cryptographic key destruction for AES (Advanced Encryption Standard): This function is used for destructing cryptographic keys for AES (FCS_CKM.4/AES).

7.2 Architectural Design Summary

The TOE claims the assurance requirement ASE_TSS.2, the security architectural information on a very high level is included in the TSS to inform the embedded software developers on how the TOE protects itself against interference, logical tampering and Insuperability**.**

Interference

Interference is manifested as interfering with the TSFs in order to get access to user data without authorization.

Logical tampering

Logical tampering consists in get access to the assets by a logical means (in contrast with physical tampering). For this TOE, logical tampering may be used on

- the access control including initialization and management of security attributes.
- the information flow control

The information flow control is enforced by the following security function "Memory Encryption".

Insuperability

Insuperability is a property that the security functionality of the TSF is always invoked. For this TOE, bypassing a security function may be caused by the following factors.

A physical perturbation on the IC: protection against this bypass is ensured by the security functions.

Leakage: leakage may lead to disclosure of sensitive information. This is countered by data processing policy.

Switching back from Normal mode to Test mode in order to get more privilege: protection against this bypass is ensured by the security functions.

Masking the security errors: protection against bypass is ensured by the security function.

Access to restricted memory: Intentional and unintentional access to restricted memory can be prevented my using memory access control.

Abusing Loader restricted functions: Access to restricted Bootloader functions is controlled by authentication.

7.3 List of the cryptographic mechanisms

The following table show the list of cryptographic mechanisms implemented.

No.	Purpose	Cryptographic Mechanism	Standard of Implementation	Key Size in Bits	Security Level above 100 Bits
1	Cryptographic Primitive	Triple DES in ECB mode	[NIST_SP800-67], [NIST_SP800-38A]	112 and 168	No
2		AES in ECB mode	[FIPS197], [NIST_SP800-38A]	128, 192 and 256	No
3		PTG.2 Random number generator	[AIS31]	-	-
4	Bootloader Authentication	Authentication Protocol based on CBC-MAC using AES	[ISO9797-1], [FIPS197], [MRTD_3, Appendix A5.1 and A5.2 using AES instead of TDES]	128	No

Table 8. List of Cryptographic Mechanisms

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

hat 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 References

[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 with Augmentation Packages, Version 1.0, BSI-CC-PP-0084-2014.

[6] A proposal for: Functionality classes for random number generators, Version 2.0, 2011, Bundesamt für Sicherheit in der Informationstechnik

[7] Developer Evidence, Version 0.8, 2013, 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

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

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

[12] JIL, "Application of Attack Potential to Smartcards": version 2.9 (January 2013).

[13] NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation.

[14] ISO/IEC 11568-2:2012, Financial services - Key management (retail) - Part 2: Symmetric ciphers, their key management and life cycle.

[15] ISO/IEC 9797-1:2011, Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher.

[16] NIST SP800-38A, Recommendation for Block Cipher Modes of Operation.

[17] ANSSI, PP0084: Interpretations, 2016-06-01.

[18] S3FV9VH: Bootloader Specification version 0.0, April 2016.

[19] ICAO, Machine Readable Travel Documents, Part 3: Machine Readable Official Travel Documents, Volume 2: Specifications for Electronically Enabled MRtds with Biometric Identification Capability, 3rd Edition, 2008.

[20] ISO/IEC 9797-1 Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher.

[21] NIST Special Publication 800-57, Recommendation for Key Management.

[22] ARM, Cortex-M3 Rev. r1p1 Technical Reference Manual, Issue E, 13th June 2007.