Security Target Lite

MS6003 Security Target-Lite

TPG0235A

General Business Use

WISeKey

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Security Target Reference

Title: MS6003 Security Target-Lite

Version number: A

Sponsor: WISEKEY

Evaluation Scheme: France (ANSSI)

Evaluator: LETI

Version	Date	Changes	Author
A	10Mar20	First Release	PDE

1.2 Purpose

This document defines the Security Target of the MS6003 project, and is provided to satisfy the Assurance Class ASE Security Target Evaluation as defined in Part 3 [3] of the Common Criteria version 3.1, Revision 5.

1.3 Reference list

[TDS]	MS6003 Semi-Formal TOE Design	MS6003_TDS
[FSP]	MS6003 Semi-Formal Functional Specification	MS6003_FSP
[DESSPEC]	MS6003 Design Specifications	MS6003_DESSPEC
[ARC]	MS6003 Security Architecture Description	MS6003_ARC
[COF]	Customer Option Form	COF

Note: For the correct version of the above documents, the user of this document should refer to the TOE Deliverables list (EDL).

1.4 TOE Overview

1.4.1 TOE Reference

The Target of Evaluation is a Secure Microcontroller with Cryptographic Software library and Wear Levelling library. The TOE is identified as shown below:

		Identifier (FAU_SAS.1 where applicable)
Part Number	MS6003	PID = 0x45 [TD]
Product Identification Number	90T04	
Hardware Revision	C	SNB0 = 0x02 [TD_GEN]
Applicable WISeKey Toolbox(s)	06.04.01.05	0x06040105 [a]
Applicable WISeKey Wear Levelling	06.03.02.02	0x06030202 [b]
Secure Bootloader (Optional)	1.1	0x00010001[c]

Table 1 - TOE Reference

The TOE is a Secure Microcontroller (Security IC) that may be used in a variety of security applications, including, Banking, Identification, Pay TV and embedded systems.

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

- maintain the integrity of the content of the TOE memories and the confidentiality of the content of protected memory areas as required by the end application(s)
- maintain the correct execution of the software residing on the TOE

The TOE implements security features to protect the confidentiality of data in the protected memory areas and may protect data in other memory areas even if not required by SFR, e.g.in ROM. The TOE also maintains the integrity of its TSF and TSF data and their confidentiality when required. Protected information is in general secret or integrity sensitive data such as Personal Identification Numbers, Balance Value (Stored Value Cards), and Personal Data Files. Other protected information data representing the access rights; these include any cryptographic algorithms and keys needed for accessing and using the services provided by the system through use of the TOE.

The TOE can be used in a smartcard application, a USB token or other devices. The intended environment is very large; and generally once issued the TOE may be stored and used anywhere, generally there is no control applied to the TOE and its operational environment.

---

[a] The toolbox identification is output by the TOE when the self-test function of the toolbox is called

[b] The Wear Levelling identification is output by the TOE when the ROM function initialisation is called.

[c] The Secure bootloader identification is output by the TOE when the tag Bootloader version is called through the Get Data command.

1.4.2 TOE Definition

1.4.2.1 TOE Overview

General Features

- ARM® SecureCore® SC300™ 32-bit RISC core featuring:
  - Harvard architecture
  - Thumb2® High-code-density instruction set
  - 3-stage pipeline architecture
  - 8-bit, 16-bit, 32-bit data types
  - Nested Vector Interrupt Controller
  - Memory Protection Unit
- Programmable Internal Clock Up to 50 MHz
- Bond Pad Locations Conforming to ISO 7816-2
- ESD Protection to
  - USB and power pad ±8kV (HBM)
  - ISO Pad ± 6kV (HBM)
  - XIN/XOUT/XIN32/XOUT32 Pad ±2kV (HBM)
  - other pads ±4kV (HBM)
- Operating Ranges: from 2.7V to 5.5V
- Compliant with EMV 4.3 Specifications and CQM
- Available in Wafers, and Industry-standard Packages

Memory

- 64KB ROM Program Memory
  - WISEKEY's crypto library
  - Wear Levelling Mechanism
- 1MB of Flash memory
- 2KB of OTP
- 24KB of RAM
  - 20 KB of CPU RAM
  - 4KB (without PUF feature) of shared RAM between Ad-X3 and CPU
    - If the PUF feature is activated, the 1st KB of RAM becomes unaccessible and is dedicated to the PUF
  - 1KB dedicated to the PUF feature (if activated)
    - This KB is taken away from the 4KB of Shared RAM

Peripherals

- ISO 7816 Controller
  - Up to 625 kbps at 5 MHz
  - Compliant with T = 0 and T = 1 Protocols
- Two I/O, multiplexed with other interfaces
- Six GPIOs, multiplexed with other interfaces
- High-speed SPI interface up to 20Mbits/s
- I2C interface up to 1Mbits/s
- USB 2.0 Full speed with clock recovery

- Two Timers
  - One 32-bit timer that can be clocked by ISO clock
  - One 16-bit timer with watchdog capability
- SysTick 24-bit Timer part of the SC300
- Random Number Generator (RNG), compliant AIS31 PTG.2
- Hardware Simple DES and Triple DES 2 keys and 3 keys, DPA/DEMA Resistant
- Hardware AES (128, 192, 256 bits key system), DPA/DEMA Resistant
- CRC16 and CRC32 Engine (Compliant with ISO/IEC 3309)
- 32-Bit Cryptographic Accelerator (Ad-X3 for Asymmetric Cryptographic Operations)
- High performance Hardware Java Card Accelerator
- Real time Counter

Security

- Dedicated Hardware for Protection Against SPA/DPA/SEMA/DEMA Attacks
- Advanced Protection Against Physical Attack, Including Active Shield, EPO, CStack Checker, Slope Detector, and Parity Error Detector.
- Environmental Protection Systems
  - Voltage Monitor
  - Frequency Monitor
  - Temperature Monitor
  - Light Protection
  - Glitch Detector
- Secure Memory Management/Access Protection (Privileged / Unprivileged)
- Memory Protection Unit (part of the SC300)
- Bus Polarity, Uniform Data Dependency Timing, Uniform Branch Timing, Trash Register Write, Clock Gating Randomisation, Secure Bridge and CPU Lockup Protection
- Physicaly Unclonable Function (PUF)

Software

- Crypto Software Toolbox
  - – AIS31 Online Test, RSA, RSA with CRT, PrimeGen (Miller Rabin), Lucas Test, ECC Multiply over GF(P), ECC Multiply over GF(2n), ECDSA generation and verification over GF(2n), ECDSA generation and verification over GF(P), Self-Test, SHA
- Wear Levelling
- Secure Bootloader
  - Optional software used in case the application code is not to be loader by Wisekey on the product.
  - Responsible for composite product manufacturer authentication and secure code loading

1.4.2.2 Security IC Embedded Software Developer Guidance Documents

REF	Title	Identifier	Version	Note
[TD_GEN]	MS6xxx Technical Datasheet	TPR0702	D	Hardware Datasheet details the FSP
[TD]	MS6003 Technical Datasheet	TPR0704	C	Hardware Datasheet details the FSP
[APP_SEC]	Security Recommendations for 90nm Products	TPR0706	D	General Security recommendations for the TOE
[APP_DES]	Secure Hardware DES/TDES for 90nm products	TPR0707	E	Hardware TDES recommendations
[APP_AES]	Secure Hardware AES for 90nm products	TPR0708	D	Hardware AES recommendations
[APP_AD-X3]	Ad-X3 Datasheet	TPR0701	C	Ad-X3 Hardware Datasheet
[APP_RNG]	Generating Random numbers to known standards for 90nm products	TPR0709	E	Details how to write an AIS31 driver using the hardware and the AIS31 test routines from the WISeKey toolbox
[APP_TBX]	Toolbox 06.04.01.xx	TPR0711	H	Toolbox 06.04.01.xx Datasheet details the FSP for the Toolbox functions
[APP_TBX_ERR]	Tbx 06.04.01.xx Erratasheet	TPR0727	D	Tbx 06.04.01.xx erratasheet
[APP_TBX_SEC]	Secure use of 06.04.01.xx	TPR0712	L	Toolbox 06.04.01.xx family Security recommendations
[APP_WEAR]	Wear Levelling library and low level Flash drivers	TPR0710	B	Wear Levelling user guide
[APP_TSB]	Secure Bootloader Transport for MS6xxx DataSheet	TPR0771	A	Secure bootloader transport user guide
[APP_CRYPT]	Efficient use of AD-X3	TPR0726	D	Ad-X3 User Guide
[APP_SEC_ACC]	MS6xxx Secure Acceptance Guidance	TPR0754	C	Secure Acceptance of MS6xxx products
[APP_SC300]	SC300 Guide DDI 0447A	0447A	A	SC300 Guide [ARM Datasheet]
[ACT]	SmartACT User's Manual	TPR0134	F	Security IC developer Code entry user manual
[COF]	Customer Option Form	MS600X_COF_V1.applu_RV.pdf	1.1	Customer Option Form

Table 2 - Reference Documents

1.4.2.3 TOE Life Cycle Addresses

Function	Company	Location
○ IC Design ○ Dataprep ○ Cryptographic Support Software Development ○ Secure bootloader development	WISeKey (MEY)	WISeKey Arteparc Bachasson, Bat A Rue de la carriere de Bachasson, CS70025 13590 MEYREUIL – FRANCE
○ Wafer Manufacturing Site	TSMC	Fab 14 1-1, Nan-Ke North Rd., Tainan Science Park, Tainan, Taiwan 741-44, R.O.C
○ Mask Manufacturing Site	TSMC	Fab 2/5: 121, Park Ave. 3, Hsinchu Science Park, Hsinchu 300-77, Taiwan, R.O.C.,

Table 3 - TOE Life Cycle Addresses

Ressource	Delivery Method
Hardware	Secure carrier
Software	Secure carrier or Secures website download
Document	Locklizard or PGP

Table 4 - TOE Delivery Method

1.4.3 TOE Description

The following figure gives an overview of the MS6003 device

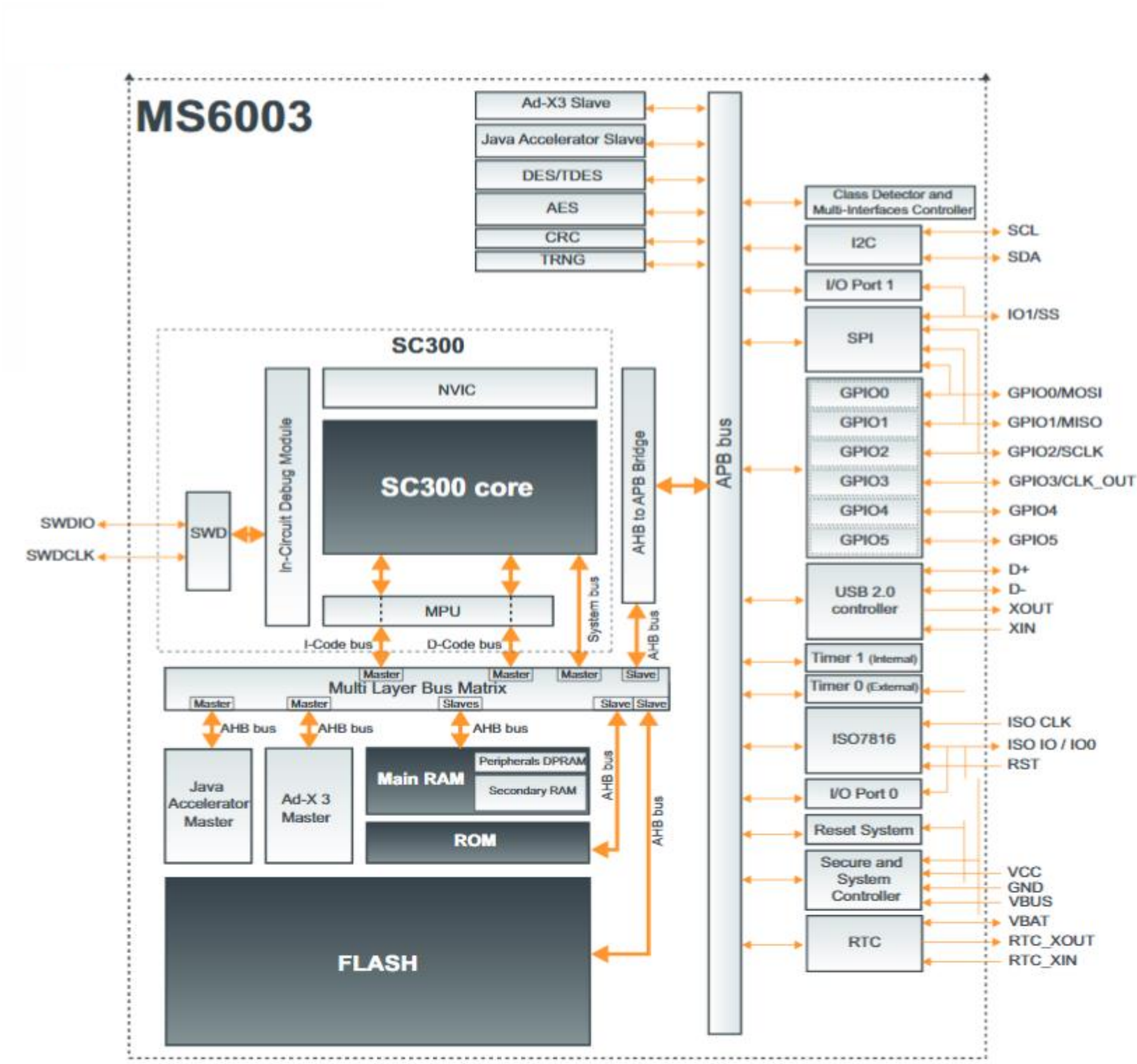


Figure 1 - *Block Diagram of the MS6003 TOE*

The Target of Evaluation (TOE) is a Secure Microcontroller (Security IC) composed of a processing unit, security components, I/O port, ROM, FLASH, and RAM memories.

The TOE will contain software elements during its life cycle. This software falls into 2 distinct categories:

- IC Dedicated Software comprising
  - IC Dedicated Test Software
  - IC Dedicated Support Software (Cryptographic Support Software / Wear Levelling)
- Security IC Embedded Software (Composite Product)

IC Dedicated Software:

- IC Dedicated Test Software

Test Software includes the test programs that are produced as evidence to support the ATE class for the evaluation of the TOE. WISeKey Engineering Code is provided to facilitate testing of the device; this Engineering Code is applicable to Phases 2 and 3 of the TOE life Cycle. To further aid testing of the TOE, additional test programs may be loaded into the FLASH. In addition to the Test Software, the TOE also includes dedicated hardware to perform testing. To allow the ITSEF to perform testing of the TOE, the TOE is delivered with a WISeKey Engineering Code and some simple test routines stored in the FLASH. It must be noted that this **Engineering Code and associated Test Software is not part of the TOE.** The entry and abuse of test modes (hardware) must be verified after TOE Delivery: this is evaluated according to the Common Criteria assurance family AVA_VAN. Refer to TOE Summary Specification for further information.

IC Dedicated Support Software

- **Cryptographic Support Software (Toolbox):** The TOE where applicable also consists of a Cryptographic Toolbox provided by WISEKEY. This Toolbox is stored in ROM and is embedded on the TOE. The user of this document should refer to the TOE Summary specification of this document for the full details. **The WISEKEY Toolbox is considered part of the TOE.**
- **Wear Levelling Library:** The TOE where applicable also consists of a Wear Levelling library provided by WISEKEY. This Library is stored in ROM and is embedded on the TOE. The user of this document should refer to the TOE Summary specification of this document for the full details. **The WISEKEY Wear Levelling Library is considered part of the TOE.**
- **Secure Bootloader:** The TOE where applicable also consists of a Cryptographic Bootloader. When the Bootloader is loaded onto the chip, it is used to download the application code and data in Flash in the composite product manufacturer premises. This dedicated software is optional and becomes unusable at the end of the successful loading of the application. When not used, it is replaced by the end-user application (which, in this case is loaded onto the chip's Flash memory by Wisekey). The **Secure Bootloader** resides in Flash. **The Secure bootloader is considered part of the TOE.**

Security IC Embedded Software:

The final version of the MS6003 device also includes embedded software; this final version of the product is referred to as a Composite Product. The Security IC Embedded Software will be stored in FLASH memory. However, some parts of it (called supplements for the Security IC Embedded Software, refer to [PP]) may also be stored in non-volatile programmable memory (FLASH). All data managed by the Security IC Embedded Software is called User Data. In addition, Pre-personalisation Data [PP] belongs to the User Data.

The Composite Product comprises

- the TOE
- the Security IC Embedded Software comprising
  - Security IC Embedded Software (stored in FLASH)
  - User Data (especially personalisation data and other data generated and used by the Security IC Embedded Software)

The **Security IC Embedded Software** and the User Data are developed separately to the hardware TOE by the WISeKey Customers. **The Security IC Embedded Software is not part of the TOE**.

Note: even though the Security IC Embedded Software is not part of the TOE, the documentation delivered as evidence for the AGD Class (**Guidance Documentation**) aid the developer to ensure the correct operation of the device and more importantly the security functionality of the device. Therefore, the **Guidance Documentation is considered part of the TOE.**

Therefore, the TOE comprises:

- the circuitry of the IC (hardware including the physical memories)
- initialisation data related to the IC Dedicated Software and the behaviour of the security functionality [a]

[a] This may also be coded in specific circuitry of the IC; for a definition refer to the Glossary.

- the associated guidance documentation
- Cryptographic and Wear Levelling Support Software
- The secure Bootloader

The TOE is designed and generated by the TOE manufacturer.

The TOE is intended to be used for a Secure Microcontroller product (Security IC), independent of the physical interface and the way it is packaged. Generally, a Security IC product may include other optional elements (such as specific hardware components, batteries, capacitors, antennae) but these are not in the scope of this Security Target.

Note that the Security IC is usually packaged. However, the way it is packaged is not specified here.

1.4.3.1 Cryptographic Toolbox Software

The TOE contains the 06.04.01.05 WISeKey Toolbox which contains the following algorithms:

Algorithm
AIS31 Online Test
Selftest
SHA-1
SHA-224
SHA-256
SHA-384
SHA-512
RSA
RSA with CRT
Prime Gen (Miller Rabin)
FIPS PrimeGen (Miller Rabin)
ECDSA over Zp
EC-DH over Zp
ECDSA over GF(2n)
EC-DH over GF(2n)

Table 5 - Toolbox algorithm

1.5 TOE Life Cycle

This Security Target is fully conformant to the claimed PP, section 2.3, the full details of the Security IC life cycle is shown in the PP. This Security Target gives a short summary of the information given in the PP. Information is also given within this Security Target to expand on the applicable phases of the life cycle of the TOE.

1.5.1 Overview of the Composite Product 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 TOE (IC) development and production:

- The IC Development (Phase 2):
  - IC design
  - IC Dedicated Software development
  - Secure Bootloader Development
- The IC Manufacturing (Phase 3):
  - integration and photomask fabrication
  - IC production
  - IC testing
  - Preparation
  - Pre-personalisation if necessary
  - Loading of the embedded software or the secure bootloader
- The IC Packaging (Phase 4)
  - Package manufacturing
  - Final test on package

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

- Security IC Embedded Software Development (Phase 1) (not part of the TOE)
- the IC Packaging (Phase 4)
- the Composite Product finishing process, preparation and shipping to the personalisation line for the Composite Product (Composite Product Integration Phase 5) [a]
- the Composite Product personalisation and testing stage where the User Data is loaded into the Security IC's memory (Personalisation Phase 6)

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

[a] It should be noted that as the TOE contains FLASH memory Phase 1 Security IC Embedded Software Development can also take place in Phase 5 that is prior to personalisation and finishing. In theory loading of the FLASH embedded software could take place later in the life cycle that is it could be loaded once shipped to the end user. This is not part of the life cycle.

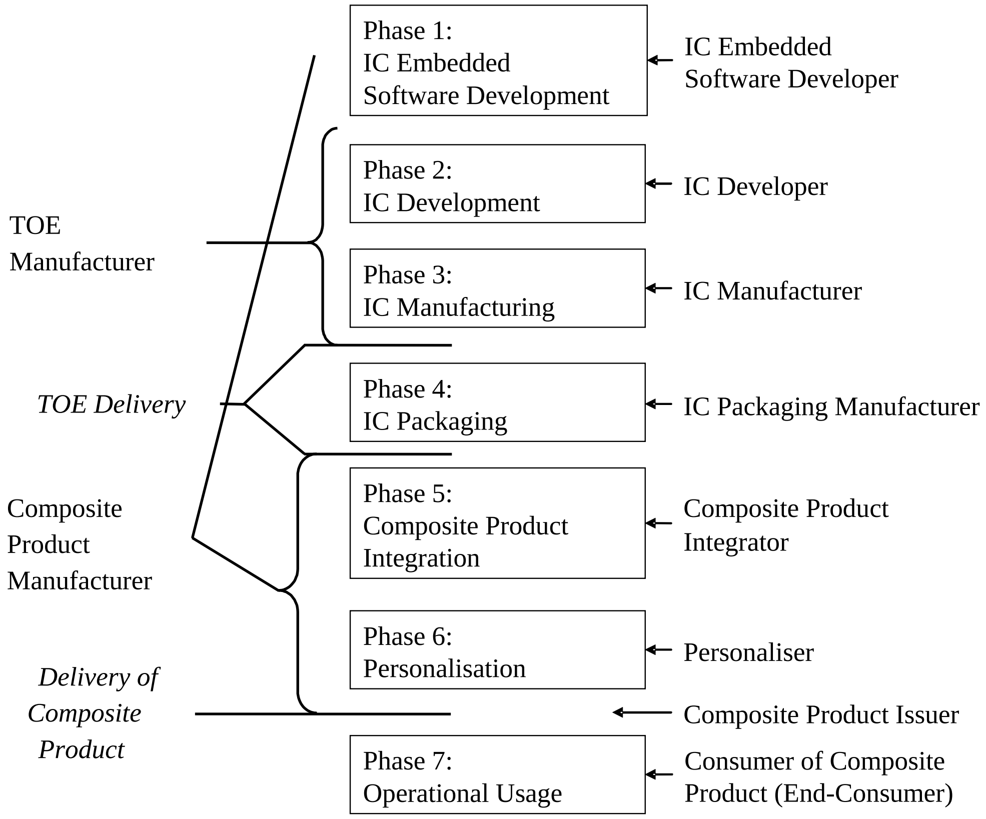


Figure 2 - *Definition of “TOE Delivery” and responsible Parties*

The Security IC Embedded Software is developed outside the TOE development between Phase 1 and 5 (as the TOE contains FLASH memory). The TOE is developed in Phase 2 and produced in Phase 3. Then the TOE can be delivered in the form of wafers or sawn wafers (dice) or industry standard package.

In the following the term “TOE Delivery” (refer to Figure 2 -) is uniquely used to indicate:

- the TOE is delivered after
  - Phase 3 in the form of wafers or sawn wafers (dice)
  - Phase 4 in the form of industry standard package
- the Security Target uniquely uses the term “TOE Manufacturer” (refer to Figure 2 -) which includes the following roles:
  - the IC Developer (Phase 2) and the IC Manufacturer (Phase 3)

Hence, the “TOE Manufacturer” comprises all roles beginning with Phase 2 and before “TOE Delivery”. Starting with “TOE Delivery”, another party takes over the control of the TOE.

The Security Target uniquely uses the term “Composite Product Manufacturer” which includes all roles (outside TOE development and manufacturing) except the End-consumer as user of the Composite Product (refer to Figure 2 -) which are the following:

- Security IC Embedded Software development (Phase 1)
- the IC Packaging Manufacturer (Phase 4)
- the Composite Product Manufacturer (Phase 5) and the Personaliser (Phase 6).

1.5.2 Phase 1 of the TOE Life Cycle

Although the pertinent phases of the Life Cycle associated with the TOE and this Security Target are Phases 2 and 3, it should be noted that parts of the TOE and this Security Target relate to Phase 1 of the TOE life Cycle. The user of this document should note the following:

- Development Samples
- Guidance Documents
- Code Entry (Security IC Embedded Software Delivery)

Development Samples: To aid with the development of the Security IC Embedded Software, development samples (with JTAG debug interface) can be delivered by WISeKey. The development samples are treated with the same level of protection by WISeKey as the final IC and are managed through the already approved controlled sample process.

Guidance Documents: To ensure that the end Composite Product is fully protected and that the SFR enforcing mechanisms cannot be tampered with or bypassed, user guidance is delivered in Phase 1 to the Security IC Embedded Software Developer. Delivery procedures are in place to ensure the confidentiality of the sensitive information contained in this documentation set, including secure courier delivery with traceability is followed. Also all parties are covered with NDA before any information is delivered (this also is applicable to Tools and Emulator).

Code Entry: Guidance documents and a delivery tool (SmartACT) are delivered to the Security IC Embedded Software Developer. The guidance document [ACT] describes how to use the SmartACT tool and how to securely transmit the final code to WISeKey for embedding on the final device. As part of the code delivery a Customer Option Form [COF] is also delivered to the Code entry team in MEY, this gives details of the options that the customer may choose for the MS6003 device.

Guidance Documents and Code Entry documents are also delivered as evidence for the AGD class, to allow the ITSEF to use these as part of the search for vulnerabilities during the Vulnerability Assessment part of the evaluation.

1.5.3 Phases 2, 3 and 4 of the TOE Life Cycle

1.5.3.1 Phase 2 IC Development

The development of the TOE is applicable to phase 2 of the life cycle and can be split into two sections:

- IC design
- Support Software Development (Cryptographic Toolbox and Wear Levelling Library)
- Secure Bootloader Development

IC design: IC design takes place on the design centre in Meyreuil France (MEY). The main project design team is located in MEY. Any sharing of information (data transfer) is achieved through a secure FTP link.

Support Software Development: The development takes place within the WISeKey Design Centre.

Secure Bootloader Development: The development takes place within the WISeKey Design Centre. The Validation takes place within GlobalLogic Inc(GL)

To ensure security of the design centre, IC design takes place within a secure environment; access is controlled with full traceability. A dedicated security person is on site at all times. The IC, Toolbox and secure bootloader development is achieved using appropriate development tools running on a secure network. All access to tools and data are controlled using appropriate restrictions and passwords. The full details are shown within the evidence provided for the ALC class. On completion of the design database, the data is transferred internally from MEY Design to MEY Dataprep to allow for generation of the Photo masks used to manufacture the TOE.

1.5.3.2 Phase 3 IC Manufacturing

The IC manufacturing falls into three sections

- Dataprep and Mask Shop
- Wafer Fab
- Testing

Dataprep and Mask Shop: The design database is delivered from the design centre to the Dataprep team within WISeKey Meyreuil France (MEY). This delivery and acceptance process and associated outputs are delivered as part of the evidence provided for the ALC class. The Photo masks used to manufacture the TOE are created by the Mask Shop. Data is transferred from MEY to the Mask Shop by secure FTP. Once created the Photo masks are transferred to the Wafer Fab by a secure approved carrier. This transfer includes tamper evidence and full traceability.

Wafer Fab: The TOE is manufactured within the Wafer Fabrication facility. The fabrication process occurs within the secure facility, as with the protection mechanisms in place in Phase 2 access to the fabrication facility is restricted. The batches are controlled using a tracking database to ensure that there is traceability of wafers at all times (including rejected wafers/dies). On completion of the fabrication process, the wafers are transferred to the test facility for test and pre-personalisation. Transfer is by a secure carrier, includes tamper evidence, and has full traceability.

Testing: This stage of the process includes production testing (refer to ATE evidence), pre-personalisation, configuration of the security functionality and optionally, wafer thinning and saw. The test facility has a controlled environment, access is restricted with full traceability, and dedicated security personnel are on site at all times.

1.5.3.3 Phase 4 IC Packaging

The IC Packaging falls into two sections

- Assembly
- Final test

Assembly: The TOE is coming in wafer to perform the following stage of the process: storage of security wafers and raw materials, assembly in final package.

Final test: The TOE is tested with a final test to verify that the TOE is still functional after the assembly.

1.5.4 Phase 3 and 5 of the TOE Life Cycle

1.5.4.1 Security IC Embedded Software Loading

The TOE is a FLASH product and the application Software is loaded during either Phase 3 or Phase 5 of the TOE life cycle. When performed in phase 3, this loading takes place in a secure environment as detailed in section 1.4.2.3., under direct control of the TOE manufacturer (either in the Test Centre or the WISeKey development (design) centre). The controls listed in section 1.5.2 are also applicable to the Software loading operation.

The loading of application software is performed by 2 means:

- Using the hardware loader during phase 3. This is performed by the TOE manufacturer. The hardware loader is part of the Test Mode and is therefore only accessible to WISeKey authenticated engineers. The hardware loader is disabled as soon as the device is configured in User Mode. Access to the hardware loader is not provided to the Composite Product manufacturer.

- Using the secure bootloader during phase 5. The secure bootloader is in this case loaded using the hardware loader during phase 3. The secure bootloader is used by the composite product manufacturer to download securely the application software.

Therefore, 2 kinds of software can be loaded during phase 3:

- The secured bootloader (package 2), allowing the loading of the final software during all phases starting at phase 4 (IC packaging) up to phase 6.
- The application software.

Only one kind of software can be loaded during phase 5:

- The application software, using the secure bootloader (which has been previously loaded in phase 3).

At the end of phase 5, if the secure bootloader has been selected as an application software method, it automatically become unusable (at the same moment it transfers the control to the application software, it “kills” itself).

1.5.5 State of the TOE between sites

The following table (figure 3) details the state and protection applicable to the TOE between sites as detailed in the Life Cycle flow.

Phase	Function	Protection methods
3	Wafer Manufacturing	Test Mode - Protected[1], Unconfigured, no loader Present
3	Test Centre	when entering : Unconfigured, Test Mode - Protected[1] when exiting : Configured, protected by authentication when exiting.
3	Delivery (wafers)	Protected by authentication
4	Assembly Manufacturing	Protected by authentication
4	Delivery (packaged chips)	Protected by authentication
5 to 7	Warehouse	User Mode, Configured, Protected by authentication
7	Customer defined	Protected by authentication

Table 6 - Definition of “TOE Delivery” and responsible Parties

[1] Protected by Authentication Process

1.5.6 Modes of Operation and Life Cycle Phases

The TOE has two distinct modes of operation:

Test Mode This mode is designed to allow authenticated test engineers access to Test features of the TOE. This mode of operation is applicable to the full life cycle of the TOE, however this is only applicable to the TOE Manufacturer via the authentication mechanism available to authenticated engineers. When entering into Test mode, the entire FLASH content is automatically erased.

User Mode This is the Mode of operation that the end Security IC (composite product) is intended to be used in. This mode of operation is dependent on the ROM and NVM code loaded. This mode of operation is available throughout the life cycle of the TOE.

2 CONFORMANCE CLAIMS

This chapter contains details the conformance claims for the TOE.

2.1 CC Conformance Claim

This Security Target claims to be conformant to the Common Criteria Version 3.1, Revision 5, April 2017.

Furthermore, it claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in the Protection Profile.

2.2 Package Claim

The TOE is evaluated to EAL5 level augmented with AVA_VAN.5 and ALC_DVS.2.

2.3 PP Claim

This Security Target is strictly conformant to the Protection Profile BSI-CC-PP-0084-2014 “Security IC Platform Protection Profile with Augmentation Packages”. The following augmentation packages from PP have been included.

- Package 1: Loader dedicated for usage in secured environment only.
  - P.Lim_Block_Loader
  - O.Cap_Avail_Loader
  - OE.Lim_Block_Loader
  - FMT_LIM.1/Loader
  - FMT_LIM.2/Loader
- Package for Masquerade “Authentication of the Security IC”
  - T.Masquerade_TOE
  - FIA_API.1
  - O.Authentication
  - OE.TOE_Auth
- Package 2: Loader for usage by authorized users only
  - P.Ctrl_Loader
  - O.Ctrl_Auth_Loader
  - OE.Loader_Usage
  - FTP_ITC.1
  - FDP_UTC.1
  - FDP_UIT.1
  - FDP_ACC.1/Loader
  - FDP_ACF.1/Loader

2.4 PP Refinements

Refinements are made to the PP within this security target relating to the Cryptographic Operations.

Refinements are made to the following Security objectives for the environment:

- OE.Resp-Appl
- OE.Lim_Block_Loader
- OE.Loader_Usage
- OE.TOE_Auth

2.5 PP Additions

The following threats, organisational security policies, assumptions, security objectives, and security functional requirements have been added.

- P.Add-Functions
- A.Key-Function
- O.Add-Functions
- FCS_COP.1
- T.Open_Samples_Diffusion
- O.Prot_TSF_Confidentiality

2.6 PP Claims Rationale

The differences between this Security Target and the BSI-CC-PP-0084-2014 that is the addition of:

- Threat
- Organisational Security Policy
- Assumptions
- Security Objectives for the TOE
- Security Functional Requirements for the TOE

Do not affect the conformance claim of this Security Target. The Rationale for these additions is given in section 6 and section 7 of this ST.

For each addition, the appropriate section clearly shows the addition, that is, section 3, Section 4 and section 6.

Although the PP recommends an EAL4 certification level with augmentations, the TOE claims an EAL5 plus certification level. This ST maintains the conformance to BSI-CC-PP-0084-2014, the rationale for this is given in sections 6.2.1 and 6.3.3.

All the Protection Profile requirements have been shown to be satisfied within this Security Target.

3 SECURITY PROBLEM DEFINITION

This chapter describes the security aspects of the environment in which the TOE is intended to be used. As this security target is conformant to BSI-CC-PP-0084-2014, this section contains only the relevant details and a summary where applicable. For complete details, refer to the Protection Profile.

3.1 Description of Assets

Assets regarding the Threats

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

- the User Data
- the Security IC Embedded Software, 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 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

According to this 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 (SC1 to SC4) the TOE shall self-protect its TSF. Therefore, critical information about the TSF shall be protected by the development environment and the operational environment. Critical information may include:

- 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 photo masks

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

3.2 Threats

The threats are listed in BSI-CC-PP-0084-2014, only a summary is provided in this Security target.

The standard threats to the TOE are shown below.

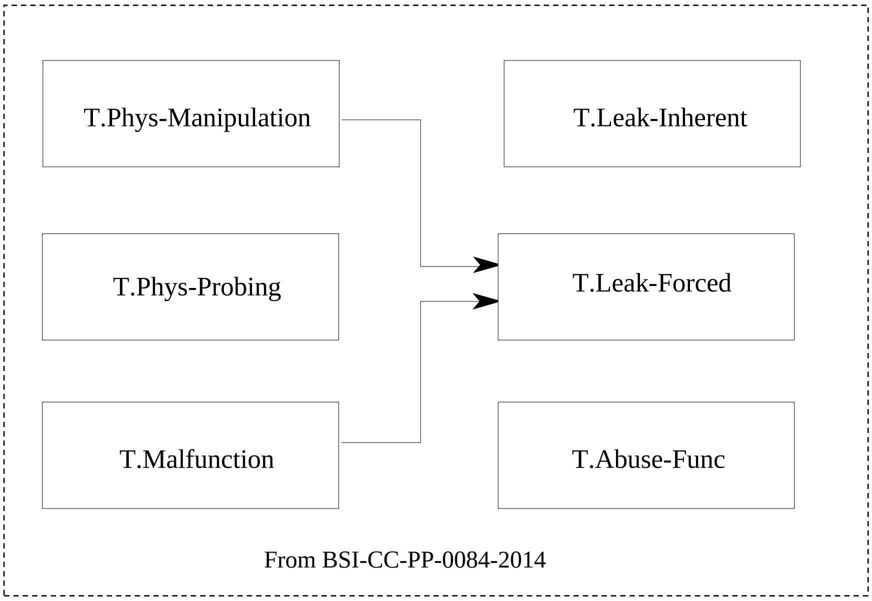


Figure 3 - *Standard Threats*

The threats relating to specific security services are shown below

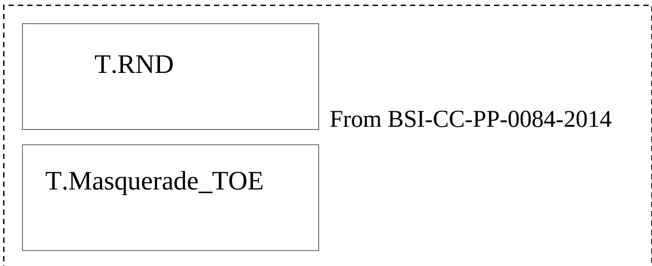


Figure 4 - *Threats related to security service*

The threats relating to diffusion of open samples are shown below.

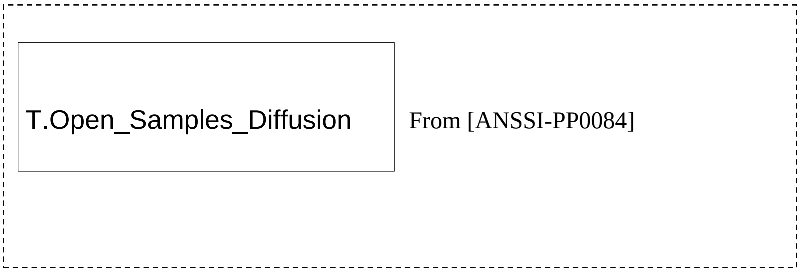


Figure 5 - *Threats related to diffusion of open samples*

The TOE shall avert the threat “Diffusion of open samples (T.Open_Samples_Diffusion)” as specified below.

T.Open_Samples_Diffusion	Diffusion of open samples An attacker may get access to open samples of the TOE and use them to gain information about the TSF (loader, memory management unit, ROM code). He may also use the open samples to characterize the behaviour of the IC and its security functionalities (for example: characterization of side channel profiles, perturbation cartography). The execution of a dedicated security features (for example: execution of a DES computation without countermeasures or by de-activating countermeasures) through the loading of an adequate code would allow this kind of characterization and the execution of enhanced attacks on the IC.

The Security IC Embedded Software may be required to contribute to preventing the threats. At least it must not undermine the security provided by the TOE. For detail refer to the assumptions regarding the Security IC Embedded Software specified in Section 3.4

The above security concerns are derived from considering the operational usage by the end-consumer (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.

3.3 Organisational Security Policies

The following figure shows the policies applied in this Security Target.

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

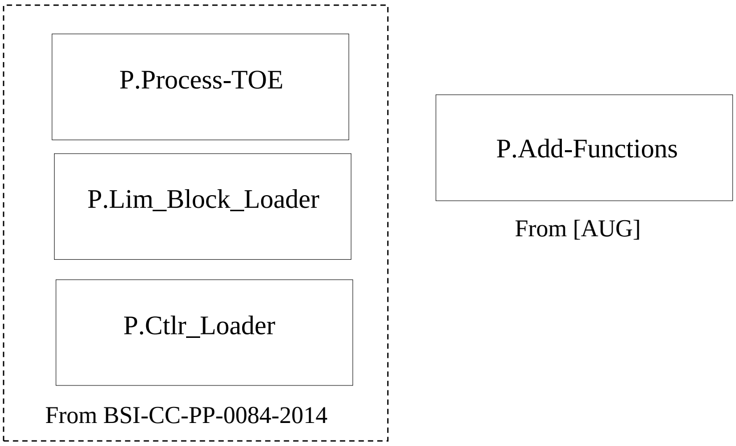


Figure 6 - *Policies*

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.

P.Add-Functions	Additional Specific Security Functionality The TOE shall provide the following specific security functionality to the Security IC Embedded Software:

- TDES [a]
- AES [a]
- RSA without CRT [b]*
- RSA with CRT [b] *
- PrimeGen (Miller Rabin algorithm) [b] *
- Secure Hash (SHA) [b] *
- ECDSA over Zp [b] *
- EC-DH over Zp [b] *
- ECDSA over GF(2n) [b] *
- EC-DH over GF(2n) [b] *
- Lucas Test [b] *

[a] The functions TDES and AES are based on a hardware dedicated part of the TOE and are applicable to all versions of the TOE

[b] The functions marked * are applicable to toolbox versions 06.04.01.02

The organisational security policy “limiting and blocking the loader functionality (P.Lim_Block_Loader)” applies to Loader dedicated for usage in secured environment. The Composite Product Manufacturer or Production Test Environment must apply this security policy.

P.Lim_Block_Loader — Limiting and Blocking the Loader Functionality

The composite manufacturer uses the Loader for loading of IC Dedicated Software, user data of the Composite Product or Installation of 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.Ctrl_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

Full details of the assumptions are listed in BSI-CC-PP-0084-2014, only a summary is provided in this Security Target. Full details are given for the additional assumption taken from [AUG].

Figure 7 shows the assumptions applied in this Security Target.

Figure 7 - *Assumptions*

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

A.Process-Sec-IC — Protection during Packaging, Finishing and Personalisation

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-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 (refer to Section 1.5) are assumed to be protected appropriately. For a list of assets to be protected, see below.

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
- material for software development support

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

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

The developer of the Security IC 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 Security IC Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced). Note that here the routines which may compromise keys when being executed are part of the Security IC Embedded Software. In contrast to this, the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines, which are part of the TOE and (ii) the processing of User Data including cryptographic keys.

4 SECURITY OBJECTIVES

The full details of the Security Objectives are listed in BSI-CC-PP-0084-2014, only a summary is provided in this Security target.

4.1 Security Objectives for the TOE

The user has the following standard high-level security goals related to the assets:

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

The Security IC may not distinguish between User Data which is publicly known or requires being confidential. Therefore, the Security IC shall protect the confidentiality and integrity of the User Data 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 8). Note that the integrity of the TOE is a means to reach these objectives.

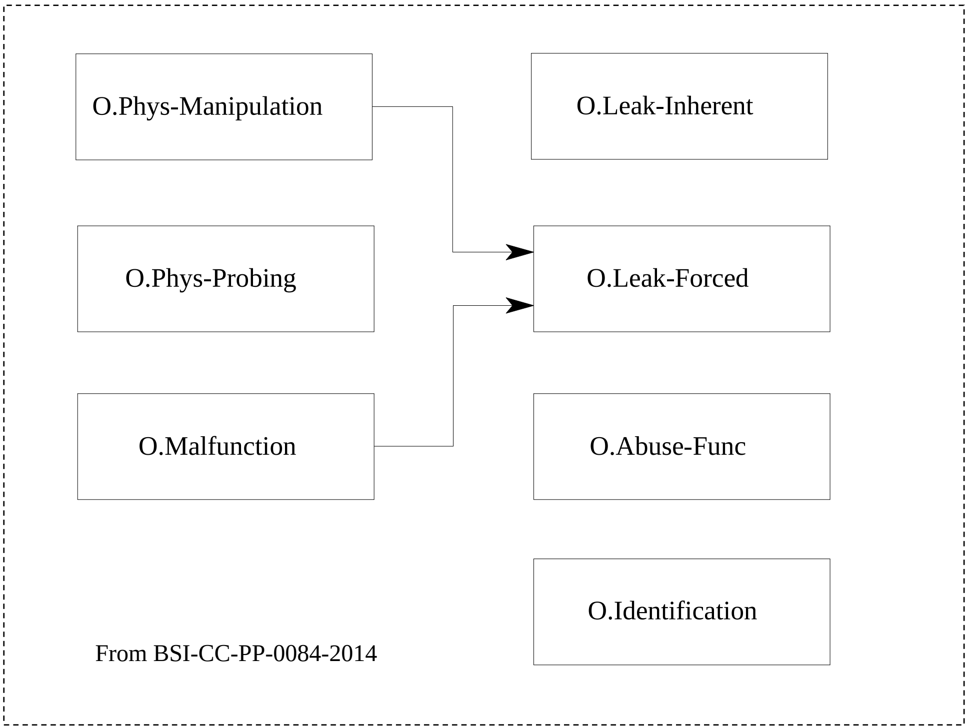


Figure 8 - *Standard Security Objectives*

According to this Security Target there is the following high-level security goal related to specific functionality:

- SG4 provide true random numbers.

The additional high-level security considerations are refined below by defining security objectives as required by the Common Criteria (refer to Figure 9).

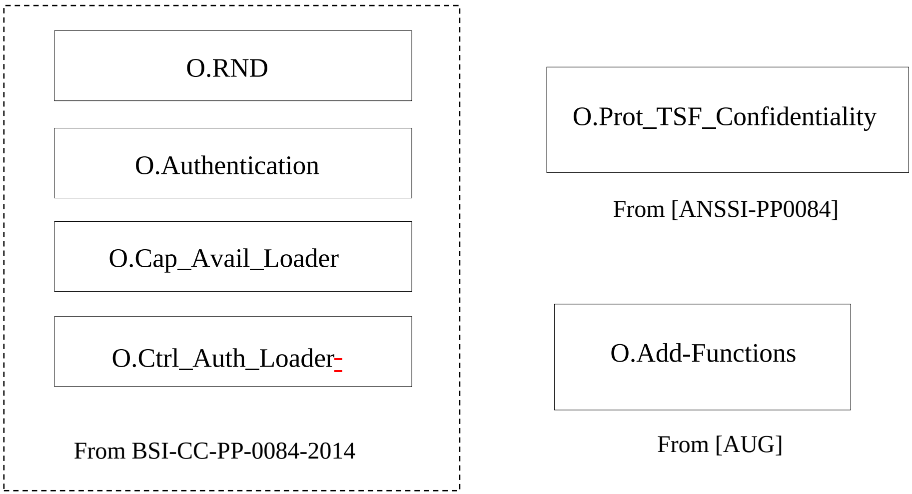


Figure 9 - *Security Objectives related to Specific Functionality*

Security Objectives related to Specific Functionality.

The TOE shall provide “Additional Specific Security Functionality (O.Add-Functions)” [AUG] as specified below.

O.Add-Functions	Additional Specific Security Functionality The TOE shall provide the following specific security functionality to the Security IC Embedded Software: • TDES • AES As per [AUG] • RSA without CRT • RSA with CRT • ECDSA over Zp • EC-DH over Zp • ECDSA over GF(2n) • EC-DH over GF(2n) • Secure HASH (SHA) As per WISeKey • PrimeGen (Miller Rabin algorithm) • Lucas Test

The TOE shall provide "Protection of confidentiality of the TSF (O.Prot_TSF_Confidentiality)" as specified below.

O.Prot_TSF_Confidentiality	Protection of confidentiality of the TSF The TOE must provide protection against disclosure of confidential operations of the Security IC (loader, memory management unit, …) through the use of a dedicated code loaded on open samples

4.2 Security Objectives for the Security IC Embedded Software

The development of the Security IC Embedded Software is outside the development and manufacturing of the TOE (cf. section 1.5). The Security IC Embedded Software defines the operational use of the TOE. This section describes the security objective for the Security IC Embedded Software.

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

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

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.

By definition, cipher or plain text data and cryptographic keys are User Data. The Security IC Embedded Software shall treat this 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 the 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 not practical 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 [AUG].

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 Phase 6.

OE.Process-Sec-IC	Protection during composite product manufacturing Security procedures shall be used after TOE Delivery up to delivery to the end-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 (refer to Section 1.5) must be protected appropriately. For a preliminary list of assets to be protected, refer to (Section 3.4, A.Process Sec IC).

The operational environment of the TOE shall provide “secure communication and usage of the loader (OE.Loader_Usage)” as specified below:

OE.Loader_Usage	Secure communication and usage of the Loader The authorized user must support the trusted communication channel with the TOE by confidentiality protection and authenticity proof of the data to be loaded and fulfilling the access conditions required by the Loader.

The operational environment shall provide “External entities authenticating of the TOE (OE.TOE_Auth)” as specified below.

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

Phase 3 and 5

The operational environment of the TOE shall provide “Limitation of capability and blocking the Loader (OE.Lim_Block_Loader)” as specified below.

OE.Lim_Block_Loader	Limitation of capability and blocking the Loader The Composite Product Manufacturer will protect the loader functionality against misuse, limit the capability of the Loader and terminate irreversibly the Loader after intended usage of the Loader.

4.4 Security Objectives Rationale

The following table shows how the assumptions, threats, and organisational security policies are addressed by the objectives. The text following the table justifies this in detail.

Assumption, Threat or Organisational Security Policy	Security Objective	Notes
A.Resp-Appl	OE.Resp-Appl	Phase 1
A.Key-Function	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 O.Cap_Avail_Loader	
T.RND	O.RND	
T.Masquerade	O.Authentication OE.TOE_Auth	
T.Open_Samples_Diffusion	O.Prot_TSF_Confidentiality O.Leak-Inherent O.Leak-Forced	
P.Add-Functions	O.Add-Functions	
P.Lim_Block_Loader	O.Cap_Avail_Loader OE.Lim_Block_Loader	Bootloader Package 1
P.Ctrl_Loader	O.Ctrl_Auth_Loader OE.Loader_Usage	Bootloader Package 2

Table 7 - Security Objectives versus Assumptions, Threats or Policies

The justification related to the assumption "Usage of Key-dependent Functions (A.Key-Function)" is as follows:

Since OE.Resp-Appl requires the Security IC Embedded Software developer to implement those measures assumed in A.Key-Function, the assumption is covered by the objective.

The justification related to the assumption "Treatment of user data of the Composite TOE (A.Resp-Appl)" is as follows:

Since OE.Resp-Appsl requires the developer of the Security IC 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 60. 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 (Table 7 - Security Objectives versus Assumptions, Threats or Policies) are stated in a way that directly corresponds to the description of the threat (Refer to section 3.2). It is clear from the description of each objective (refer to Section 4.1), 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 security objective “T_Open_Samples_Diffusion” is as follows:

The threat “Diffusion of open samples” (T.Open_Samples_Diffusion) is directly covered by the TOE security objective “Protection of the confidentiality of the TSF” (O.Prot_TSF_Confidentiality) based on the self-protection of the TOE and the authentication mechanism of the Loader. Additionally to O.Prot_TSF_Confidentiality (Protection of the confidentiality of the TSF), T.Open_Samples_Diffusion threat is countered by O.Leak-Inherent (Protection against Inherent Information Leakage) and O.Leak-Forced (Protection against Forced Information Leakage) from the PP.

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

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

Nevertheless, the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced define how to implement the specific security functionality required by P.Add-Functions (Note that these objectives support that the specific security functionality is provided in a secure way as expected from P.Add-Functions). Especially O.Leak-Inherent and O.Leak-Forced refer to the protection of confidential data (User Data or TSF Data (section 7.1) in general. User Data are also processed by the specific security functionality required by P.Add-Functions.

The following text gives details of the clarification added to OE.Resp-Appl. By definition cipher or plain text data and cryptographic keys, are defined as User Data. Therefore, the Security IC Embedded Software will protect such data if required and use keys and functions appropriately in order to ensure the strength of cryptographic operation. Strength 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. These measures make sure that the assumption A.Resp-Appl is still covered by the security objective OE.Resp-Appl although additional functions are being supported according to P.Add-Functions.

The justification related to the organisational security policy “Limitation of capability and blocking of the Loader (P.Lim_Block_Loader) is as follows.

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)” and the security objective for the TOE environment “Secure communication and usage of the Loader (OE.Loader_Usage)”.

The justification of the additional threat (T.Open_Samples_Diffusion), policy (P.Add-Functions) and assumption (A.Add-Functions) do not contradict the rationale already given in the Protection Profile for assumptions, policy and threats defined in the PP and within this Security Target. These additions come from [AUG_v1.0] and [ANSSI_PP0084].

5 EXTENDED COMPONENTS DEFINITION

The extended components:

- FCS_RNG.1
- FMT_LIM.1
- FMT_LIM.2
- FAU_SAS.1
- FDP_SDC.1
- FIA_API.1

The above are defined within the Protection Profile [PP] that this Security Target is strictly conformant to.

6 IT SECURITY REQUIREMENTS

The standard Security Requirements are shown in Figure 10. These security components are listed and explained below.

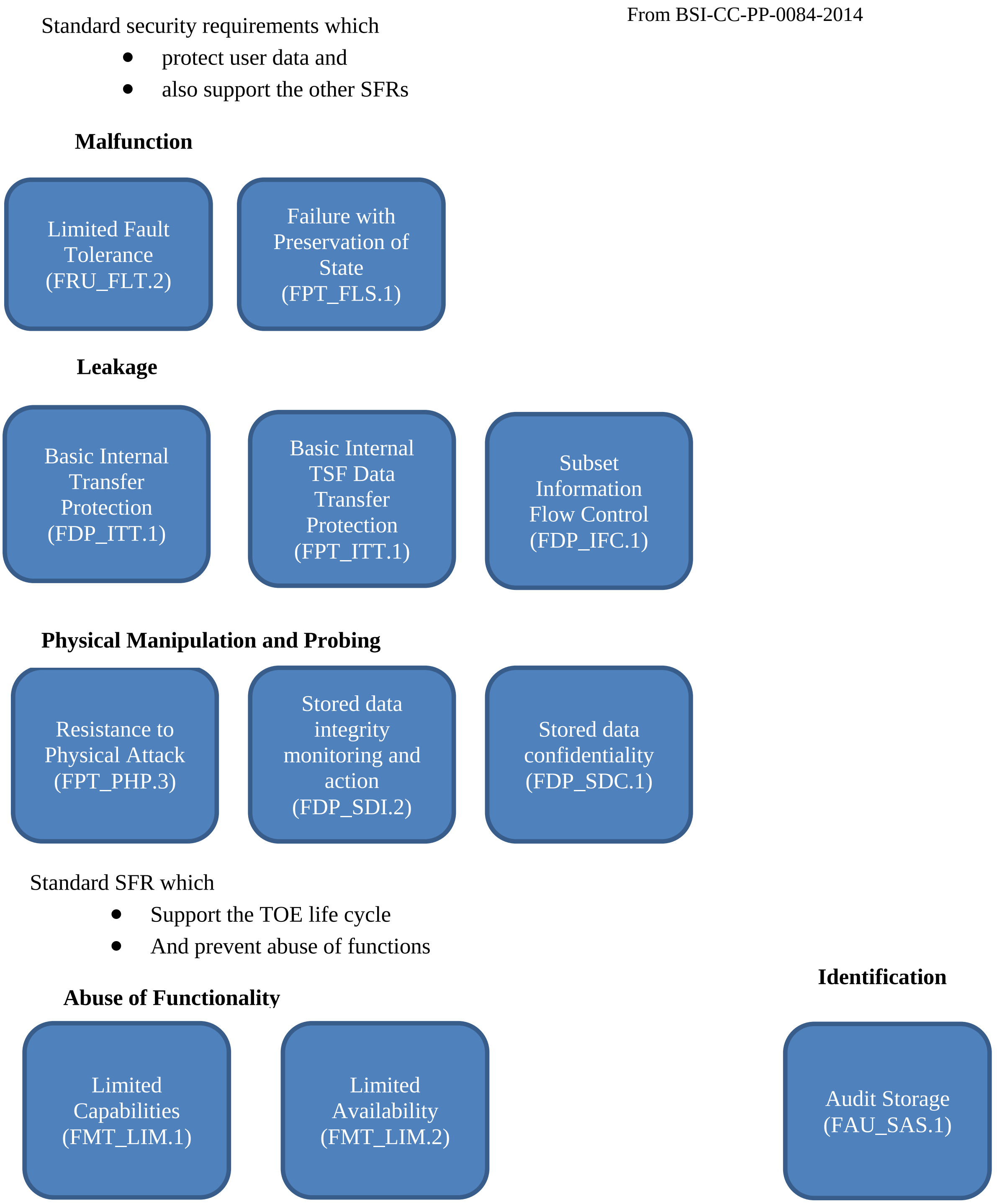


Figure 10 - *Standard Security Requirements*

The Security Functional Requirements related to Specific Functionality are shown in Figure 11. These security functional components are listed and explained below.

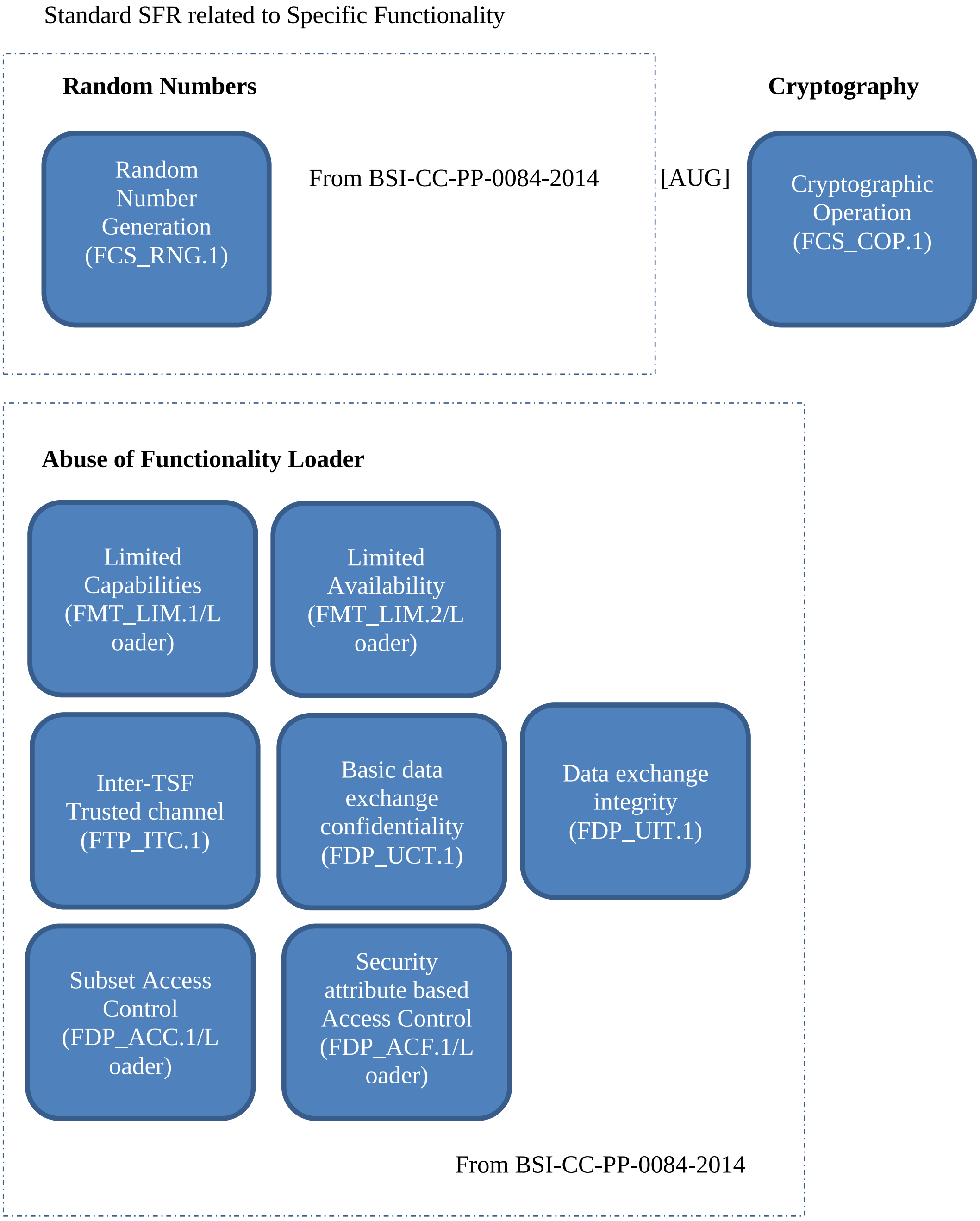


Figure 11 - *Security Functional Requirements related to Specific Functionality*

6.1 Security Functional Requirements for the TOE

In order to define the Security Functional Requirements Part 2 of the Common Criteria was used. However, some Security Functional Requirements have been refined (please refer to the Protection Profile [PP]).

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 Degraded fault tolerance
Dependencies:	FPT_FLS.1 Failure with preservation of secure state.
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)***[a]**.**
Refinement:	**The term "failure" above also covers "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.
Dependencies:	No dependencies.
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***[b]**.**
Refinement:	**The term "failure" above also covers "circumstances". The TOE prevents failures for the "circumstances" defined above.**

Refinement Note	Environmental conditions include but are not limited to power supply, clock, and other external signals (e.g. reset signal) necessary for the TOE operation.

[a] The TOE operates in a stable way within this operating window, this is verified during the development and manufacturing phase of the life cycle. This is verified by the ITSEF during the ATE Assurance Class analysis.

[b] TSF_ENV_PROTECT details the operating conditions that are not tolerated by the TOE (namely Voltage and temperature out of bounds, and internal frequency following below a defined level). The TOE takes action through TSF_AUDIT_ACTION to ensure the TOE fails in a 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.
Dependencies:	FMT_LIM.2 Limited availability.
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***[a]**.**

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.
Dependencies:	FMT_LIM.1 Limited capabilities.
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***[b].

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.
Dependencies:	No dependencies.
FAU_SAS.1.1	The TSF shall provide ***the test process before TOE Delivery***[c] with the capability to store ***the Initialisation Data and/or Pre-personalisation Data and/or supplements of the Security IC Embedded Software***[d] in the ***Non-Volatile Memory***.

[a] TSF_TEST details the Limited capability and availability policy.

[b] TSF_TEST details the Limited capability and availability policy.

[c] The code entry process allows the Security IC Embedded Software developer to deliver pre-personalisation data, details are given in the SmartACT manual [ACT]. Some configuration of the TOE is allowed using the [COF].

[d] The Security IC Embedded Software Developer may deliver data during the code entry process [ACT].

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 ***ROM, RAM and Flash***.

The TOE shall meet the requirement "Stored Data Integrity monitoring and action (FDP_SDI.1)" as specified below.

FDP_SDI.2	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1	The TSF shall monitor user data stored in containers controlled by the TSF for ***integrity errors*** on all objects, based on the following attributes: ***ROM, RAM and Flash content.***
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall ***automatically invoke a violation.***

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.
Dependencies:	No dependencies.
FPT_PHP.3.1	The TSF shall resist ***physical manipulation and physical probing*** [a] to the ***TSF***[b] by responding automatically such that the SFRs are always enforced.
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.**

Note: The TOE provides the ability to perform an automatic response when a violation is detected. To allow the Security IC Embedded Software developer to choose an appropriate response the TOE allows some configuration of this response mechanism (refer to TSF_AUDIT_ACTION). Further details of the automatic response mechanisms can be found in [GEN_TD] (section 10.1.3 Violation reactions).

[a] Direct Probing, manipulation by operating the TOE, out with the specified operating conditions [TD].

[b] The TSF are detailed in TOE Summary Specification Section.

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.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_ITT.1.1	The TSF shall enforce the ***Data Processing Policy*** [a] to prevent the ***disclosure or modification*** of user data when it is transmitted between physically separated parts of the TOE.
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.
Dependencies:	No dependencies.
FPT_ITT.1.1	The TSF shall protect TSF data from ***disclosure or modification*** when it is transmitted between separate parts of the TOE.
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.

[a] The user of this document should refer to TSF_LEAK_PROTECT for the SFP: Data Processing Policy

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.
Dependencies:	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1	The TSF shall enforce the ***Data Processing Policy***[a] on ***all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software***[b].

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

User Data of the Composite TOE and TSF data shall not be accessible from the TOE except when the Security IC Embedded Software decides to communicate the User Data of the Composite TOE 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

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

FCS_RNG.1	**Random number generation – AIS31 PTG.2**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
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 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.***
(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 continuously. The online test is***

[a] The user of this document should refer to TSF_LEAK_PROTECT for the SFP: Data Processing Policy

[b] The sensitive information that must be protected includes information when transferred from one memory location to another by the user or Security IC Embedded Software or being operated on by the hardware processors. This information must be protected as it would allow an attacker to gain knowledge of the functions of the TOE TSF, or gain access to cryptographic key information.

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 ***octets of bits*** that meet:

(PTG.2.6) ***Test procedure A 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.***

Notes on RNG

Definition of the Security Functional Requirement FCS_RNG.1 has been taken from [BSI-CC-PP-0084-2014] and is further refined according to [Functionality classes for random number generators, Version 2.0, 18. September 2011].

The average Shannon entropy 0.997 per internal random bit compares to 7.984 per octet.

The RNG is evaluated against AIS31 without post-processing so the internal random number defined is the RNGDAS output.

Cryptography

The TOE shall meet the requirement “Cryptographic Operation – TDES (FCS_COP.1/TDES)” as specified below.

FCS_COP.1/TDES	**Cryptographic operation – TDES**
Hierarchical to:	**No other components.**
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/TDES	The TSF shall perform ***hardware TDES encryption and decryption*** in accordance with a specified cryptographic algorithm: ***Triple Data Encryption Standard (TDES)*** with cryptographic key sizes: ***112-bit and 168-bit*** that meet the following ***NIST SP 800-67 [8], NIST SP 800-38A [9]***

Note on TDES	TDES Cryptographic operation based on a hardware dedicated part of the TOE and is applicable to all versions of the TOE. This is not based on the PP-0084.

FCS_COP.1/AES	**Cryptographic operation – AES**
Hierarchical to:	No other components.
Dependencies:	([FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/AES	The TSF shall perform ***hardware AES encryption and decryption*** in accordance with a specified cryptographic algorithm: ***Advanced Encryption Standard (AES)*** with cryptographic key sizes: ***128-bit, 192-bit and 256-bit*** that meet the following: ***FIPS 197 November 26, 2001 [7], NIST SP 800-38A [9].***

Note on AES	AES Cryptographic operation based on a hardware dedicated part of the TOE and is applicable to all versions of the TOE. This is not based on the PP-0084.

FCS_COP.1/RSA without CRT	**Cryptographic operation – RSA without CRT**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform ***data encryption and decryption*** in accordance with a specified cryptographic algorithm: ***RSA without CRT*** and cryptographic key sizes: ***between 96 bits and 5120 bits*** that meet the following: ***PKCS#1 V2.2, 27th October, 2012 [12]***.

FCS_COP.1/RSA with CRT **Cryptographic operation – RSA with CRT**

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall perform ***data encryption and decryption*** in accordance with a specified cryptographic algorithm: ***RSA with CRT data*** and cryptographic key sizes: ***between 192 bits and 5120 bits*** that meet the following: ***PKCS#1 V2.2, 27th October, 2012 [12]***.

FCS_COP.1/ECDSA over Zp **Cryptographic operation – ECDSA over Zp**

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall perform ***signature generation and verification*** in accordance with a specified cryptographic algorithm: ***EC-DSA over Zp*** and cryptographic key sizes: ***between 192 bits and 521 bits*** that meet the following: ***FIPS 186-4 [11].***

FCS_COP.1/Prime Generation **Cryptographic operation – Prime Generation**

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1 The TSF shall perform ***Miller-Rabin Prime Generation*** in accordance with a specified cryptographic algorithm: ***PrimeGen*** and cryptographic key sizes: ***between 96 bits and 5120 bits*** that meet the following: ***FIPS 186-4 [11].***

FCS_COP.1/Lucas Test **Cryptographic operation – Lucas Test**

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall perform ***Lucas Test*** in accordance with a specified cryptographic algorithm: ***Lucas Test*** and cryptographic key sizes: ***between 96 bits and 5120 bits*** that meet the following: ***FIPS 186-4 [11].***

FCS_COP.1/EC-DH over Zp **Cryptographic operation – EC-DH over Zp**

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall perform ***signature generation and verification*** in accordance with a specified cryptographic algorithm: ***EC-DH over Zp*** and cryptographic key sizes: ***between 192 bits and 521 bits*** that meet the following: ***ISO/IEC 11770-3:2008*.**

FCS_COP.1/ECDSA over GF(2n)	**Cryptographic operation – ECDSA over GF(2n)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1	The TSF shall perform ***signature generation and verification*** in accordance with a specified cryptographic algorithm: ***ECDSA over GF(2n)*** and cryptographic key sizes: ***between 163 bits and 571 bits*** that meet the following: ***FIPS 186-4 [11].***

FCS_COP.1/EC-DH over GF(2n)	**Cryptographic operation – EC-DH over GF(2n)**
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1	The TSF shall perform ***signature generation and verification*** in accordance with a specified cryptographic algorithm: ***EC-DH over GF(2n)*** and cryptographic key sizes: ***between 163 bits and 571 bits*** that meet the following: ***ISO/IEC 11770-3:2008*****.**

FCS_COP.1/SHA-1	**Cryptographic operation**
Hierarchical to:	No other components.
FCS_COP.1.1	The TSF shall perform ***data signing*** in accordance with a specified cryptographic algorithm: ***SHA-1*** and cryptographic key sizes: ***no cryptographic key size*** that meet the following: ***Secure Hash Standard, FIPS 180-4, 2015 August*****.**
Dependencies:	No dependency as no Key are used

FCS_COP.1/SHA-224 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform ***data signing*** in accordance with a specified cryptographic algorithm: ***SHA-224*** and cryptographic key sizes: ***no cryptographic key size*** that meet the following: ***Secure Hash Standard, FIPS 180-4, 2015 August***.

Dependencies: No dependency as no Key are used

FCS_COP.1/SHA-256 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform ***data signing*** in accordance with a specified cryptographic algorithm: ***SHA-256*** and cryptographic key sizes: ***no cryptographic key size*** that meet the following: ***Secure Hash Standard, FIPS 180-4, 2015 August***.

Dependencies: No dependency as no Key are used

FCS_COP.1/SHA-384 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform ***data signing*** in accordance with a specified cryptographic algorithm: ***SHA-384*** and cryptographic key si ***no cryptographic key size*** that meet the following: ***Secure Hash Standard, FIPS 180-4, 2015 August***.

Dependencies: No dependency as no Key are used

FCS_COP.1/SHA-512 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform ***data signing*** in accordance with a specified cryptographic algorithm: ***SHA-512*** and cryptographic key sizes: ***no cryptographic key size*** that meet the following: ***Secure Hash Standard, FIPS 180-4, 2015 August***.

Dependencies: No dependency as no Key are used

Abuse of Functionality Loader

The TOE shall meet the requirement "Limited capabilities – Loader (FMT_LIM.1/Loader)" is specified below:

FMT_LIM.1/Loader	**Limited capabilities - Loader**
Hierarchical to:	No other components.
Dependencies:	FMT_LIM.2 Limited availability - Loader.
FMT_LIM.1.1/Loader	The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: **Deploying Loader functionality after *Activate command of the Loader, the loader is disabled and no more available to* store user data or manipulate by an unauthorized user [18].**

The TOE shall meet the requirement "Limited availability – Loader (FMT_LIM.2/Loader)" as specified below:

FMT_LIM.2/Loader	**Limited availability -Loader**
Hierarchical to:	No other components.
Dependencies:	FMT_LIM.1 Limited capabilities - Loader.
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 *reception of the activate command[19]. The loader is disabled and the application loaded is activated.***

The TOE shall meet the requirement "Limited availability – Loader (FTP_ITC.1)" as specified below:

FTP_ITC.1	**Inter-TSF trusted channel**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1.1	The TSF shall provide a communication channel between itself and ***the Composite Product Manufacturer*** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit ***another trusted IT product*** to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for ***deploying Loader load binary command***.

[18] TSF_FLASH details the Limited capability and availability policy for a Loader.

[19] TSF_FLASH details the Limited capability and availability policy for a Loader.

The TOE Functional Requirement "Basic data exchange confidentiality (FDP_UCT.1)" is specified as follows:

FDP_UCT.1	**Basic data exchange confidentiality**
Hierarchical to:	No other components.
Dependencies:	FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path FDP_ACC.1/Loader Subset access control, or FDP_IFC.1 Subset information flow control
FDP_UCT.1.1	The TSF shall enforce ***the Loader SFP*** to ***receive*** user data in a manner protected from unauthorised disclosure.

FDP_UIT.1	**Data exchange integrity**
Hierarchical to:	No other components.
Dependencies:	FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control
FDP_UIT.1.1	The TSF shall enforce the ***Loader SFP*** to ***receive*** user data in a manner protected from ***modification, deletion, insertion*** errors.
FDP_UIT.1.2	The TSF shall be able to determine on receipt of user data, whether ***modification, deletion, insertion*** has occurred.

FDP_ACC.1/Loader	**Subset access control – Loader**
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1/Loader Security attribute based access control.
FDP_ACC.1.1/Loader	The TSF shall enforce the ***Loader SFP*** on (1) ***the subjects Composite Product Manufacturer***, (2) ***the objects user data in Flash memory***, (3) ***the operation deployment of Loader***

FDP_ACF.1/Loader	**Security attribute based access control - Loader**
Hierarchical to:	No other components.
Dependencies:	FDP_ ACC.1/Loader Subset access control FMT_MSA.3 Static attribute initialisation
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 ***Composite Product Manufacturer*** with security attributes ***Knowledge of the shared secret key (diversified by customer) used for the mutual authentication,*** (2) the objects user data in ***Flash memory*** with security attributes ***successful authentication done***

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: ***Knowledge of the shared secret key (diversified by customer) used for the mutual authentication***.
FDP_ACF.1.3/ Loader	FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: ***Perform a successful authentication using a mutual authentication mechanism.***
FDP_ACF.1.4/ Loader	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: ***Try counter mechanism manage a kill card***.

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 mutual authentication of the bootloader*** to prove the identity of the ***TOE*** to an external entity.

6.2 Security Assurance Requirements for the TOE

This Security Target is evaluated according to Security Target evaluation (Class ASE)

The “Security Assurance Requirements for the TOE”, for the evaluation of the MS6003 TOE are those taken from the Evaluation Assurance Level 5 (EAL5) and augmented by taking the following components:

- ALC_DVS.2 and AVA_VAN.5

The assurance requirements are (augmentation from EAL5+ highlighted)

Class ADV: Development

Architectural design	(ADV_ARC.1)
Functional specification	(ADV_FSP.5)
Implementation representation	(ADV_IMP.1)
Well-structured internals	(ADV_INT.2)
TOE design	(ADV_TDS.4)

Class AGD: Guidance documents

Operational user guidance	(AGD_OPE.1)
Preparative user guidance	(AGD_PRE.1)

Class ALC: Life-cycle support

CM capabilities	(ALC_CMC.4)
CM scope	(ALC_CMS.5)
Delivery	(ALC_DEL.1)
Development security	(ALC_DVS.2)
Life-cycle definition	(ALC_LCD.1)
Tools and techniques	(ALC_TAT.2)

Class ASE: Security Target evaluation

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

Class ATE: Tests

Coverage	(ATE_COV.2)
Depth	(ATE_DPT.3)
Functional tests	(ATE_FUN.1)
Independent testing	(ATE_IND.2)

Class AVA: Vulnerability assessment

Vulnerability analysis	(AVA_VAN.5)

6.2.1 Refinements of the TOE Assurance Requirements

The Protection Profile BSI-CC-PP-0084-2014 defines refinements to the Security Assurance requirements defined in CC V3.1 Part 3. The TOE is assessed to EAL5 Level with additional augmentations which are taken into account in this analysis.

The [PP] allows the TOE to be evaluated above the EAL4+ requirements given in the [PP], therefore the fact that this Security Target is assessed to EAL5 level, it still maintains the conformance claim to [PP]. The refinements stated in [PP] remain consistent with the EAL5 package claims of this Security Target.

The full details of the Assurance Requirement refinements are listed in BSI-CC-PP-0084-2014.

6.3 Security Requirements Rationale

6.3.1 Rationale for the security functional requirements

Table 8 below gives an overview of how the security functional requirements are combined to meet the security objectives. The detailed justification follows 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”
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”
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 “Basic internal transfer protection” FPT_ITT.1 “Basic internal TSF data transfer protection” FDP_IFC.1 “Subset information flow control” plus those listed for O.Malfunction and O.Phys-Manipulation FRU_FLT.2 “Limited fault tolerance” FPT_FLS.1 “Failure with preservation of secure state” FPT_PHP.3 “Resistance to physical attack”
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 “Basic internal transfer protection” FPT_ITT.1 “Basic internal TSF data transfer protection” FDP_IFC.1 “Subset information flow control” FPT_PHP.3 “Resistance to physical attack” FRU_FLT.2 “Limited fault tolerance” FPT_FLS.1 “Failure with preservation of secure state”
O.Identification	FAU_SAS.1 “Audit storage”
O.RND	FCS_RNG.1 “Quality metric for random numbers” plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced FDP_ITT.1 “Basic internal transfer protection” FPT_ITT.1 “Basic internal TSF data transfer protection” FDP_IFC.1 “Subset information flow control” FPT_PHP.3 “Resistance to physical attack” FRU_FLT.2 “Limited fault tolerance” FPT_FLS.1 “Failure with preservation of secure state”
O.Add-Functions	FCS_COP.1 “Cryptographic Operation”
O.Cap_Avail_Loader	FMT_LIM.1 “Limited capabilities - Loader” FMT_LIM.2 “Limited availability - Loader”
OE.Resp-Appl	not applicable
OE.Process-Sec-IC	not applicable

Objective	TOE Security Functional and Assurance Requirements
OE.Lim_Block_Loader	not applicable
O.Authentication	FIA_API.1 “Authentication Proof of Identity”
OE.TOE_Auth	not applicable
O.Ctrl_Auth_Loader	FTP_ITC.1 “Inter-TSF Trusted channel” FDP_UCT.1 “Basic data exchange confidentiality” FDP_UIT.1 “Data exchange integrity” FDP_ACC1/Loader “Subset Acces Control” FDP_ACF.1/Loader “Security attribute based Acces Control”
OE.Loader_Usage	not applicable
O.Prot_TSF_Confidentiality	FDP_ACC.1/Loader “Subset Acces Control” FDP_ACF.1/Loader “Security attribute based Acces Control”

Table 8 - 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. Thus, attackers cannot discover such data by measurements of emanations, power consumption or other behaviour 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 may be possible if the processing time of algorithms implemented in the software depends on the secret data). 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). 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 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). This support must be addressed in the Guidance Documentation. Together with this FPT_PHP.3 is suitable to meet the objective.

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

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

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

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

Other security functional requirements which prevent attackers from circumventing the functions implementing these two security functional requirements (e.g. FPT_PHP.3) also support this objective. The relevant objectives are also listed in Table 8 - Security Requirements versus Security Objectives.

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

The justification related to the security objective “TOE Acces control and authenticity for the loader (O.Ctrl_Auth_Loader) “ is as follows:

The TSF provides trusted communication channel with authorized user, supports confidentiality protection and authentication of the user data to be loaded and access control for usage of the Loader functionality.

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

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

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

The justification related to the security objective “TOE Authentication to external entities (O.Authentication) “ is as follows:

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.

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

FCS_RNG.1 requires the TOE to provide random numbers of good quality, and to specify a quality metric defined within this Security Target.

Other security functional requirements, which prevent physical manipulation and malfunction of the TOE (see the corresponding objectives listed in the table Table 8 -) support this objective because they prevent attackers from manipulating or otherwise affecting the random number generator.

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

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

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

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

The security functional requirements “Cryptographic operation (FCS_COP.1)” exactly requires those functions to be implemented which are demanded by O.Add-Functions. Therefore, FCS_COP.1 is suitable to meet the security objective.

Depending on the functionality of the end composite device, the Security IC Embedded Software will have to support the objective by using the additional functions as specified by the [CC]. The user data processed by the functions relating to FCS_COP.1 is protected as defined for the end application. The Embedded Software will have to support the objective O.Add-Functions by implementing the security functional requirements below:

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

The justification related to the security objective “Capability and availability of the Loader (O.Cap_Avail_Loader)” is as follows:

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 justification related to the security objective Protection of the confidentiality of the TSF (O.Prot_TSF_Confidentiality) is covered by the SFR as follows:

- The SFR FDP_ACC.1/Loader defines the subjects, objects and operations of the Loader SFP enforced by the FDP_ACF.1/Loader.
- The SFR FDP_ACF.1/Loader requires the TSF to implement authentication mechanism for the Protection of the confidentiality of the TSF

6.3.2 Dependencies of security functional requirements

Table 9 below lists the security functional requirements defined in this Security Target, their dependencies and whether they are satisfied by other security requirements defined in this Security Target. The text following the table discusses the remaining cases.

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this ST
FRU_FLT.2	FPT_FLS.1	**Yes**
FPT_FLS.1	None	**No dependency**
FMT_LIM.1	FMT_LIM.2	**Yes**
FMT_LIM.2	FMT_LIM.1	**Yes**
FAU_SAS.1	None	**No dependency**
FPT_PHP.3	None	**No dependency**
FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1	**Yes**
FDP_IFC.1	FDP_IFF.1	**See discussion below**
FPT_ITT.1	None	**No dependency**
FDP_SDC.1	None	**No dependency**
FDP_SDI.2	None	**No dependency**
FCS_RNG.1	None	**No dependency**
FCS_COP.1	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) FCS_CKM.4	**See discussion below**
FIA_API.1	None	**No dependency**
FDP_ACC.1/Loader	FDP_ACF.1/Loader	**Yes**
FDP_ACF.1/Loader	FDP_ACC.1/Loader FMT_MSA.3	**Yes** **See discussion below**
FMT_LIM.1/Loader	FMT_LIM.2/Loader	**Yes**
FMT_LIM.2/Loader	FMT_LIM.1/Loader	**Yes**
FTP_ITC.1	None	**No dependency**
FDP_UCT.1	FTP_ITC.1 or FTP_TRP.1 FDP_ACC.1/Loader or FDP_IFC.1	**Yes** **Yes**
FDP_UIT.1	FTP_ITC.1 or FTP_TRP.1 FDP_ACC.1/Loader or FDP_IFC.1	**Yes** **Yes**

Table 9 - 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).

The dependencies for FCS_COP.1 cannot be satisfied by the TOE the dependencies for key management must be met by the Security IC Embedded Software, they are dependent on the end usage of the Security IC.

For FCS_COP/SHA_xxx, because no key is used, there is no need for key import as required by dependency to FDP_ITC.1, FDP_ITC.2 or key generation as required by dependency to FCS_CKM.1 or destruction as required by dependency to FCS_CKM.4.

As Table 9 shows, all other dependencies of functional requirements are fulfilled by security requirements defined in this Protection Profile.

The discussion in Section 6.3.1 has shown how the security functional requirements support each other in meeting the security objectives of this Protection Profile. In particular, the security functional requirements providing resistance of the hardware against manipulations (e.g. FPT_PHP.3) support all other more specific security functional requirements (e.g. FCS_RNG.1) because they prevent an attacker from disabling or circumventing the latter.

For FDP_ACF.1/Loader, because the security attributes used to enforce the Loader SFP are fixed by the IC manufacturer and no new objects under control of the Loader SFP are created, there is no need for Static attribute initialisation as required by dependency to FMT_MSA.3.

For FDP_UCT.1, because an Inter-TSF trusted channel (FTP_ITC.1) and Subset access control (FDP_ACC.1) is used to enforce the Loader SFP, there is no need for trusted path and subset information flow control as required by dependency to FTP_TRP.1 and FDP_IFC.1.

For FDP_UIT.1, because an Inter-TSF trusted channel (FTP_ITC.1) and Subset access control (FDP_ACC.1) is used to enforce the Loader SFP, there is no need for trusted path and subset information flow control as required by dependency to FTP_TRP.1 and FDP_IFC.1.

6.3.3 Rationale for the Assurance Requirements

Although [PP] requires EAL4 the TOE is assessed against the EAL5 requirements, this gives the additional assurance that the TOE is developed and tested in a structured and methodical way, part of the TOE development is described in semi-formal terms to allow the ITSEF and Certification Body to understand the TOE to a detailed level.

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

An assurance level of EAL5 with the augmentations AVA_VAN.5 and ALC_DVS.2 are required for this type of TOE since it is intended to defend against sophisticated attacks. This evaluation assurance package was selected to permit a developer to gain maximum assurance from positive security engineering based on good commercial practices. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defence against such attacks, the evaluators should have access to the low level design and source code. The user of this document should refer to [PP] for further understanding on the requirement for the augmentations.

6.3.4 Security Requirements are Internally Consistent

The discussion of security functional requirements and assurance components in the preceding sections has shown that consistencies 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.

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 identified in Section 3.1. Therefore, the security functional requirement FPT_PHP.3 is not only required to meet the security objective O.Phys-Manipulation. In addition, 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 identified in Section 3.1. Therefore, the security functional requirements FRU_FLT.2 and FPT_FLS.1 are not only required to meet the security objective O.Malfunction. In addition, 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 prevent the disclosure of primary assets identified in Section 3.1 it is important that the security functional requirements preventing 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 prevent forced leakage 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 prevent the disclosure of primary assets identified in Section 3.1. 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 prevent the disclosure of primary assets identified in Section 3.1. 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 is 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 not controllable by the Security IC Embedded Software (e.g. test features). 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 services of the TOE or of the Security IC Embedded Software or (iii) to enable other attacks on the assets. Therefore, the binding between the 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 services which limit the availability could be bypassed, deactivated or changed by physical manipulation or a malfunction caused by an attacker. Therefore, if Limited Availability (FMT_LIM.2) is vulnerable[20], 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 (test software[21]) 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 services of the TOE or of the Security IC Embedded Software or to enable other attacks on the assets. Therefore, if an attacker could benefit from using such functions[22], 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 prevent 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

[20] or, in the extreme case, not being provided

[21] Test Software is not included in the TOE refer to section 1.4.2.2

[22] the capabilities are not limited in a perfect way (FMT_LIM.1)

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 addition of the security functional requirement FCS_COP.1 and how it relates to the security objective O.Add-Functions is detailed in 6.3.1. It should be noted that any assets related to the cryptographic operations (e.g. cryptographic keys) are protected by the objectives relating to “Leakage”, “Physical Manipulation and Probing” and “Malfunction”.

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 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.

7 TOE SUMMARY SPECIFICATION

This section demonstrates how the TOE matches the Security Functional requirements as detailed in section 6.1 (Security functional Requirements).

It gives a description of the TSF elements of the TOE to allow an understanding of how the security of the TOE matches the SFR of section 6.1, and also how they TOE protects itself against tampering, interfering and bypass of the TSF Features of the TOE.

7.1 Description of TSF Features of the TOE

7.1.1 TSF_TEST Test Interface

- Test Mode (TME)
- Serial Number Registers Write

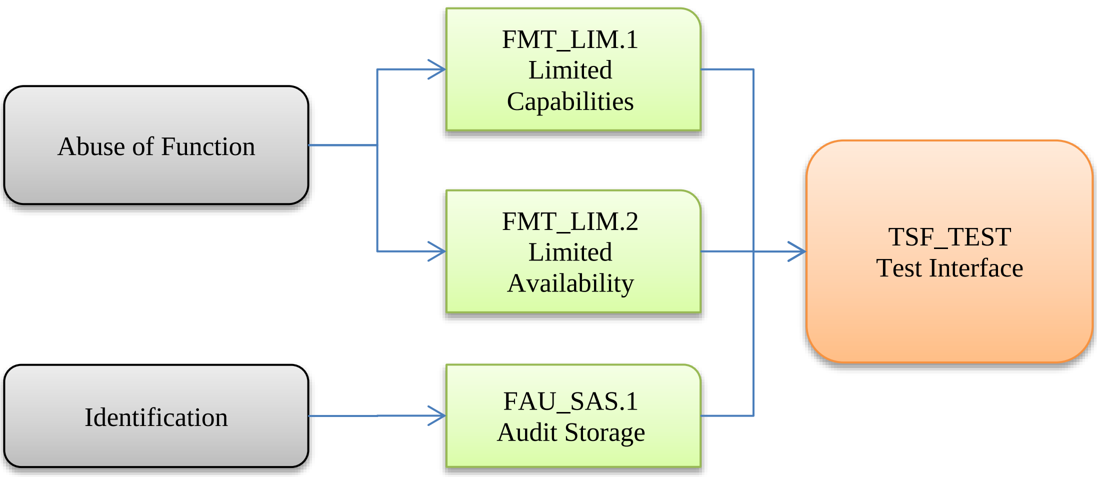
The TOE has an engineering test mode (TME).

Test Mode Entry: TME is protected by a test mode entry condition and is only accessible to authenticated test engineers. This mechanism is strong enough to protect the test functionality of the TOE. Furthermore, enter in the test mode causes the full erasing of the Flash.

Serial Number Register Write: In Test Mode it is possible to store pre-personalisation data. The serial number information is also written at this time.

SFP: Limited capability and availability Policy	The TOE Test features are only available to authenticated Wisekey engineers with the knowledge of the Test Mode Entry sequence.

7.1.1.1 SFR to TSF Test Interface



7.1.2 TSF_ENV_PROTECT Environmental Protection

- Hardware Protection (Active Shield)
- Voltage Monitor
- Frequency Monitor
- Temperature Monitor
- Light Scan Detector
- Memory Encryption (Scramblers)
- Bus Encryption (Protection)
- Structure and Layout

Hardware Protection: The TOE has an active shield that covers the top of the chip, this provides tamper evidence protection.

Voltage Monitor: The VCC and GND lines to the TOE are monitored to protect the TOE from the supply going out of bounds.

Frequency Monitor: The internal frequency is monitored to protect the internal clock falling below a defined level.

Temperature Monitor: The operating temperature of the TOE is monitored to prevent the TOE from being operated out-with the correct operating conditions.

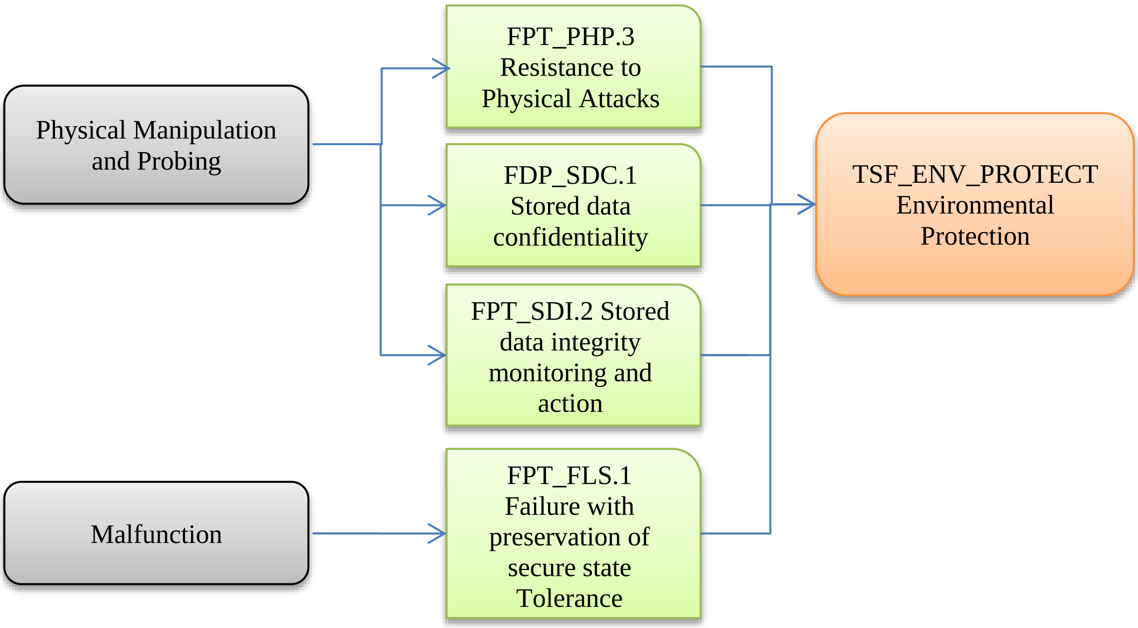
Light Scan Detector: The TOE provides a Light scan Detector (LSD) to protect against laser (or focused light) scanning of the TOE.

Memory encryption: The memories and register file are encrypted.

Bus Encryption: Layout structures are implemented to make internal bus probing difficult. The TOE contains no visible bus structures.

Structure and Layout: This provides complexity to any attack that involves identifying specific areas of the TOE.

7.1.2.1 SFR to TSF_ENV_PROTECT



7.1.3 TSF_LEAK_PROTECT Leakage Protection

- Internal Clock (VFO)
- VFO Jitter
- Dummy Interrupt
- Random Branch Insertion
- Frequency Divider
- CRC Power Scrambling
- Dummy Flash write
- Bus Polarity
- Uniform Data Dependency Timing
- Uniform Branch Timing
- Trash Register Write
- Clock Gating Randomisation

Internal Clock: The TOE provides an internal Variable Frequency Oscillator (VFO).

VFO Jitter: The VFO frequency offers variances of the frequency through time (Jitter) to help against side channel leakage analysis.

Dummy Interrupt: The TOE can trigger Dummy Interrupts.

Random Branch Insertion: The TOE can insert a branch-to-self instruction at a random interval.

Frequency Divider: The VFO clock can be varied by dividing the clock; this can also be set up by the IC embedded software to perform this subdivision on the fly.

CRC Power Scrambling: CRC Power scrambling introduces a random component into the power signature of the chip.

Dummy Flash write: This allows the Security IC embedded Software to cause a dummy write of the Flash.

Bus Polarity: When enabled, can mask to power signature of the bus.

Uniform Data Dependency Timing: Ensures that arithmetic operations take a fixed/repeatable number of cycles.

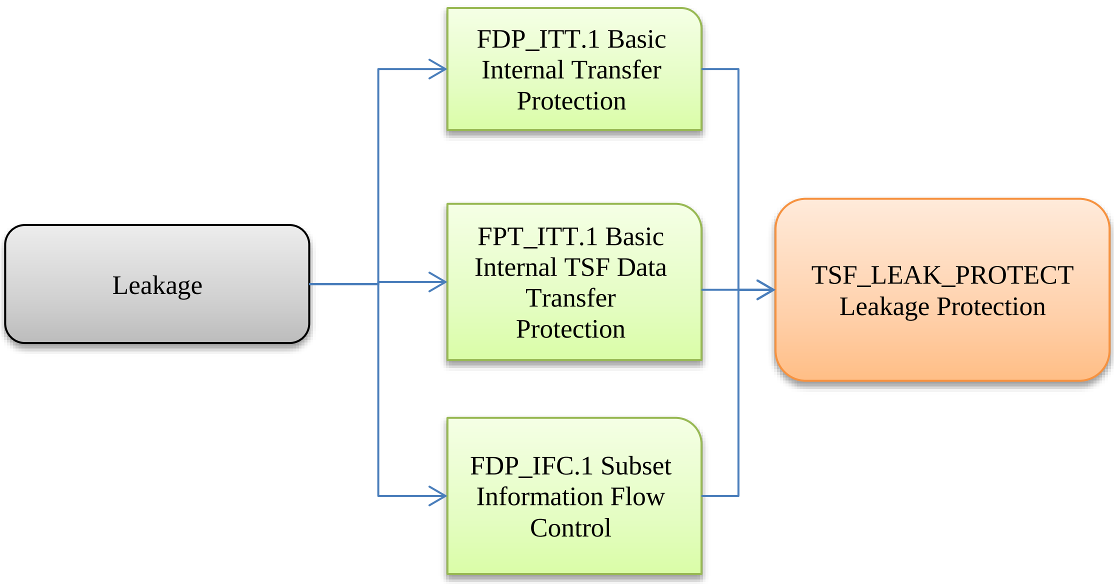
Uniform Branch Timing: is used to ensure that branch timing is the same for all branch paths.

Trash Register Write: when enabled it allows selected operations that normally end without writing to write the current value to a trash register. Therefore the power signature associated with whether or not a register write does or does not occur is masked.

Clock Gating Randomisation: This feature causes the hardware to modify in a random way the power signature of the core.

SFP: Data Processing Policy	When processing or moving information within the TOE, the TOE should not leak any specific information that would allow an attacker to gain sufficient knowledge to gain access to secret information stored within the TOE memories.

7.1.3.1 SFR to TSF_LEAK_PROTECT



7.1.4 TSF_DATA_PROTECT Data Protection

- Memory Access Protection
- CRC Accelerator
- Parity Checker
- Mirroring
- Enhanced Protection Object (EPO)
- Program stack Checker
- Glitch Detectors
- Secure Bridge
- Memory Protection Unit
- CPU Lockup Protection
- Flash Lock
- Filters on Power Supply

Secure Memory Management: The TOE provides a Memory Protection Unit.

CRC Accelerator: The TOE provides a Cyclic Redundancy Check (CRC32 or CRC16).

Parity Checker: The TOE features parity checking on the ROM, Registers and RAM. If a fault is injected by modifying a data bit the parity check will be able to detect it and generate a violation.

Mirroring: Some of the internal security registers have been duplicated/ mirrored. A violation is triggered if the register and its mirror differ.

Enhanced Protection Object: The NVM read is protected against attempted perturbations.

Program Stack Checker: The SC300 core provides the IC Embedded Software a Program Stack Checker.

Secure Bridge: When enabled, any byte/halfword access will generate error response, and any fetch access will also generate an error response.

Glitch Detectors: The Glitch Detectors can detect a glitch on the Vcc signal. This protects against attempted perturbations.

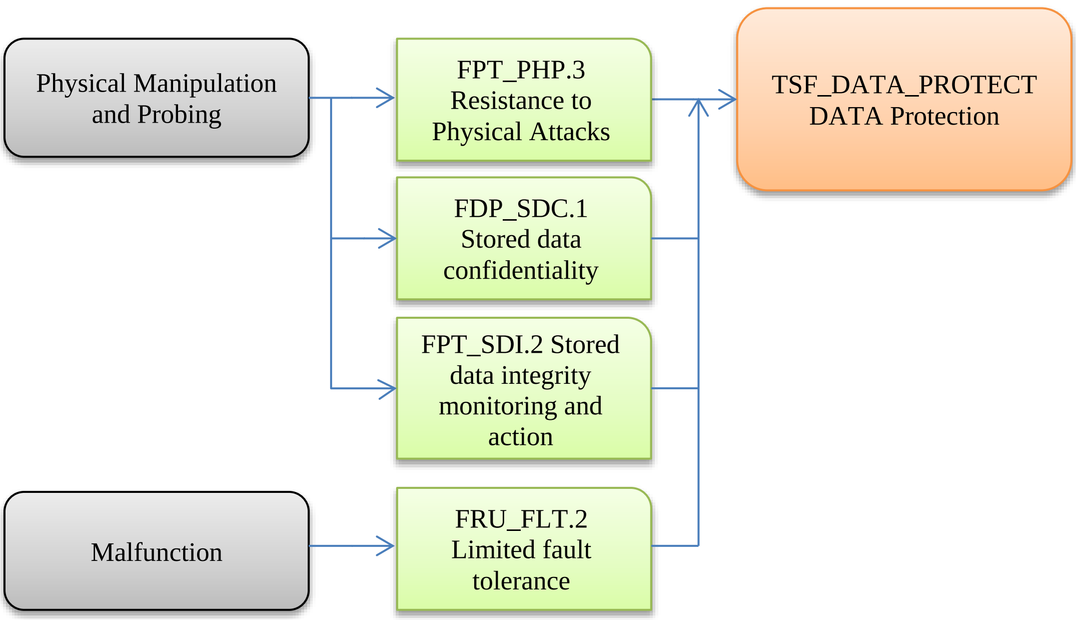
Memory Protection Unit: Is a component for memory protection, supports regions and sub-regions, overlapping protection regions, access permissions, possibility to define memory access characteristics.

CPU Lockup Protection: The CPU is able to detect some incoherent code, identified as lockup state.

Flash Lock: Allows a part of the flash to be locked from any write or erase.

Filters on Power Supply: The filters on the power supply protect and provide robustness against glitches.

7.1.4.1 SFR to TSF_DATA_PROTECT



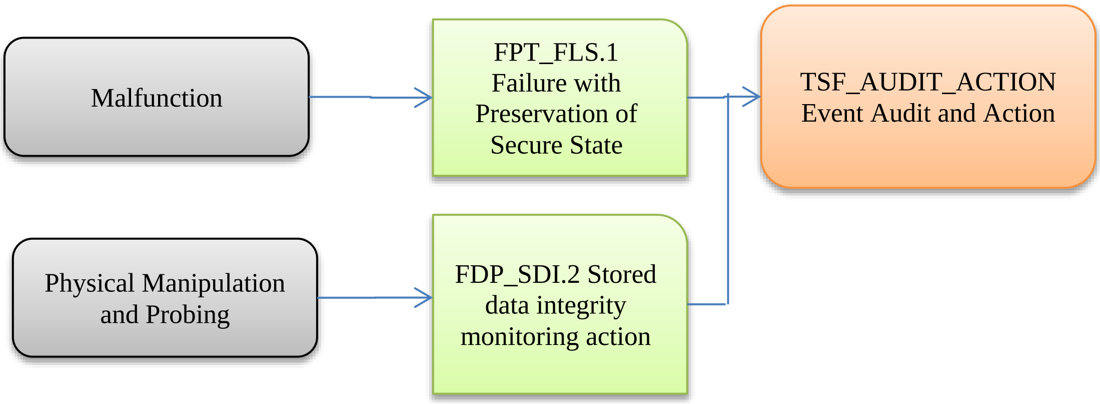
7.1.5 TSF_AUDIT_ACTION Event Audit and Action

- Reset System
- Security Registers

Reset System: The TOE allows the Security IC Embedded Software to select the response the TOE makes to a security violation. Several reactions are possible depending upon how the TOE is configured by the Security IC Embedded Software.

Security registers: The TOE includes several registers to report failures (violations) detected by the security mechanisms of the TOE.

7.1.5.1 SFR to TSF_AUDIT_ACTION



7.1.6 TSF_RNG Random Number Generator

- True RNG
- RNG Status Register
- RNGDAS
- RDWDR

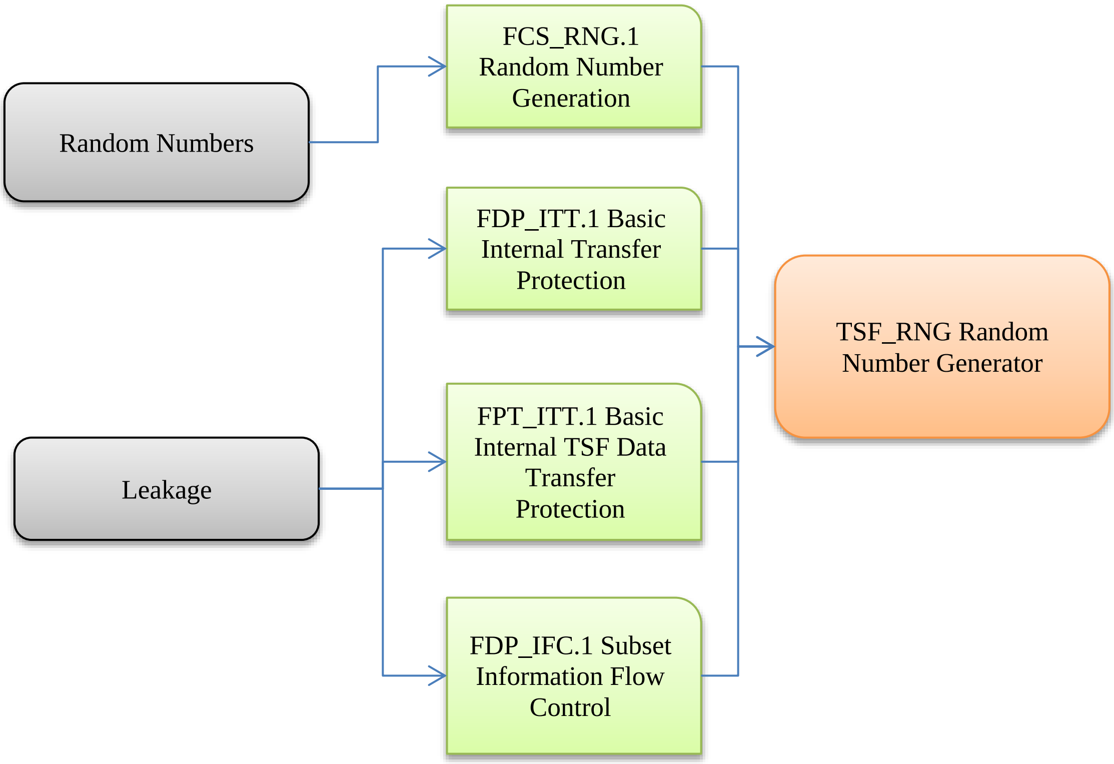
True RNG: The TOE has an analogue noise source that can be used to provide random numbers when required by the Security IC Embedded Software.

RNG Status Register: The TOTFAIL bit is set if the analogue noise source fails. The Security IC Embedded Software can monitor this security flag and the software can then take appropriate action.

RNGDAS: The Analogue Noise Source is sampled to create a digitized analogue source that is accessible to the Security IC Embedded Software through the RNGDAS register.

RDWDR: The digital analogue source from RNGDAS can be post processed using a seeded LFSR. The result of the post-processed data is accessible to the Security IC Embedded Software through the RDWDR register

7.1.6.1 SFR to TSF_RNG



7.1.7 TSF_FLASH FLASH Loader Protection

- Mutual Authentication
- Bootloader availability

Mutual Authentication: Valid mutual authentication must be performed before flash loading

Bootloader availability: When the secure bootloader activate command is received, the secure bootloader is no more available.

SFP: Limited capability and availability Policy for a loader	The code loading capability must be protected from un-authorised use, to ensure that loading only performed in the composite manufacturer premises as defined in the Life Cycle. The activate command disable the secure bootloader function.

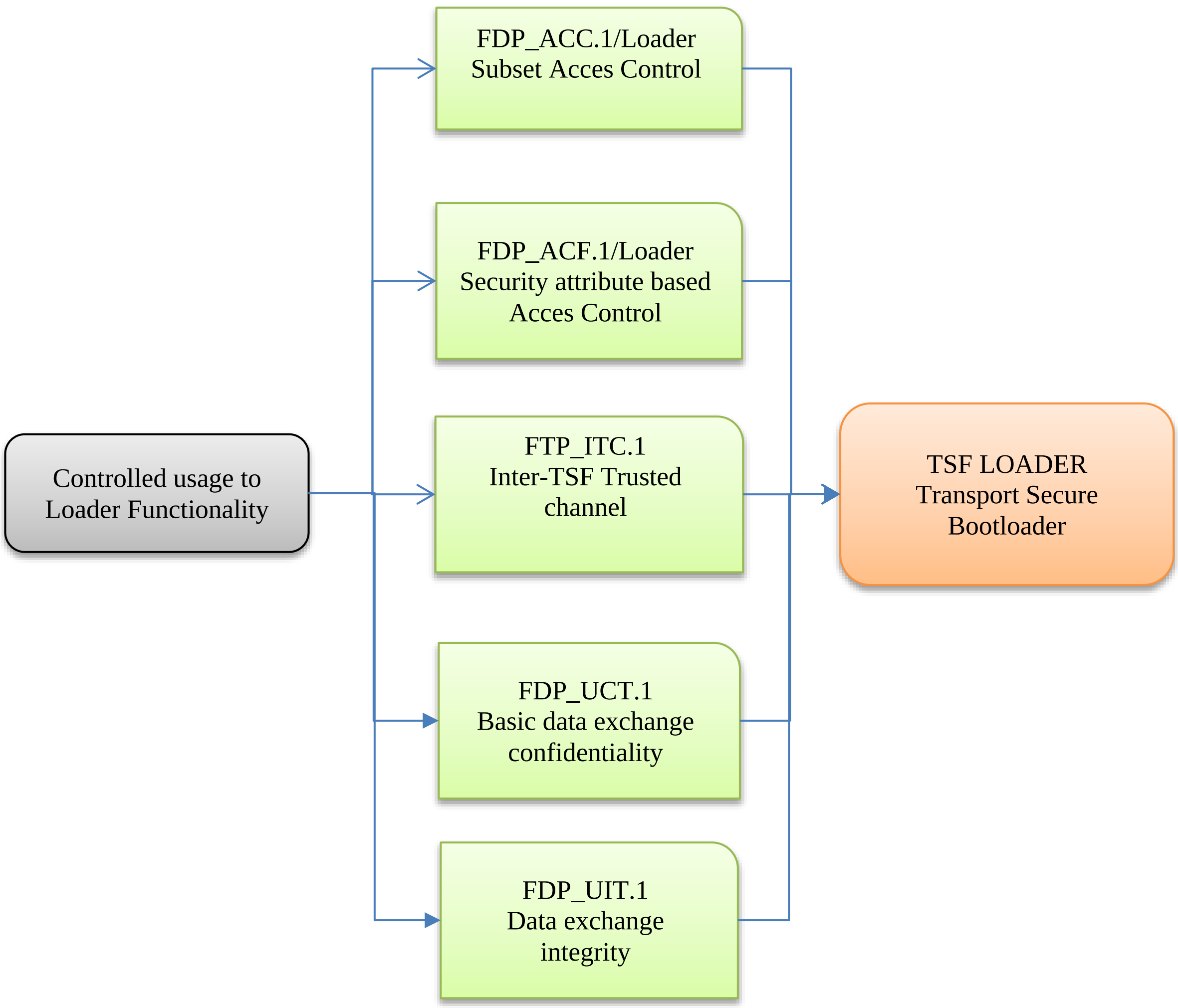
7.1.7.1 SFR to TSF_FLASH

7.1.8 TSF LOADER Transport Secure Bootloader

- Mutual Authentication
- Encryption
- Integrity Check
- Try counter mechanism

Mutual Authentication: The TOE performs mutual authentication following SCP03 specifications.
Encryption: The TOE performs a computation using the hardware AES.
Integrity Check: The TOE performs a 16 bits CRC computation using the hardware CRC.
Try counter mechanism: The TOE performs a try counter mechanism

7.1.8.1 SFR to TSF LOADER



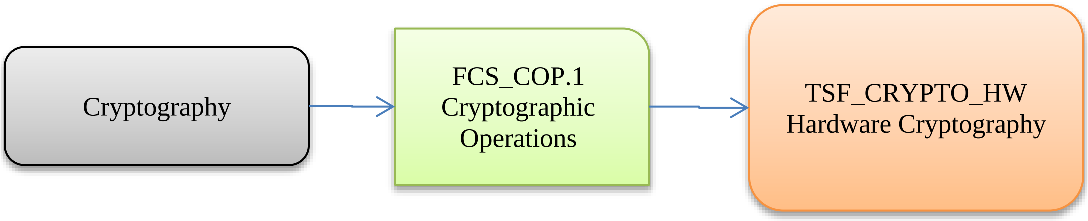
7.1.9 TSF_CRYPTO_HW Hardware Cryptography

- Hardware Triple DES
- Hardware AES

Hardware Triple DES: The TOE provides a hardware DES / TDES engine that enables fast cryptographic computations.

Hardware AES: The TOE provides a hardware AES engine which enables fast cryptographic computations.

7.1.9.1 SFR to TSF_CRYPTO_HW



7.1.10 TSF_CRYPTO_SW Toolbox Cryptography

- AIS31 Online Test
- RSA
- RSA with CRT
- PrimeGen (Miller Rabin)
- Lucas Test
- ECC Multiply over GF(P)
- ECC Multiply over GF(2n)
- ECDSA generation and verification over GF(2n)
- ECDSA generation and verification over GF(P)
- Self-Test
- HASH (SHA)

Self-Test: The TOE can perform a test of the crypto toolbox at the request of the Security IC Embedded Software

AIS31 Online Test: The TOE provides the ability to run online tests of the random numbers provided to the RNGDAS register. The test performed is a χ^2 (chi squared) test to check the randomness of the data.

RSA without CRT: The TOE provides RSA without CRT (Modular Exponentiation), data encryption and decryption functions.

RSA with CRT: The TOE provides RSA with CRT, data encryption and decryption functions.

PrimeGen: The TOE provides RSA cryptographic key generation capability using Miller Rabin algorithm with confidence criteria (t parameter) between 0 and 255.

Lucas Test: The TOE provides a Lucas Test for Primality

ECC Multiply over GF(P): The TOE provides a Multiplication on an Elliptical Curve

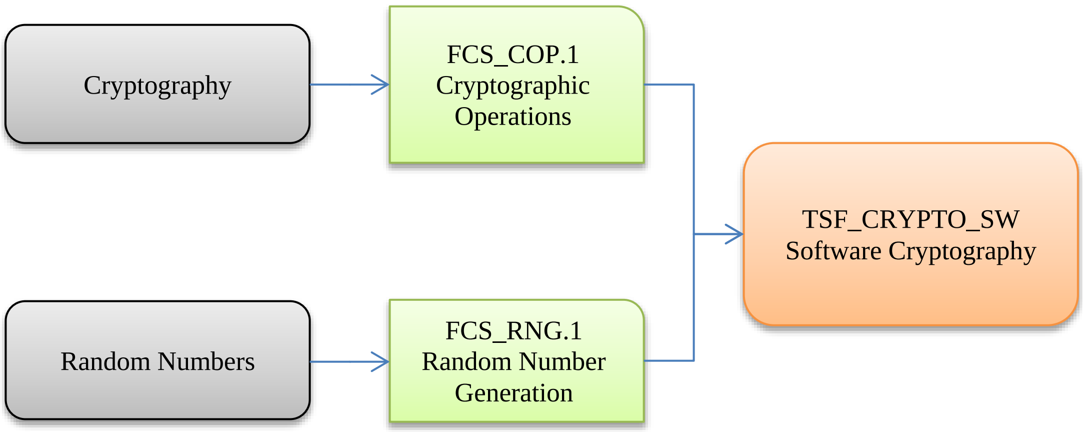
ECC Multiply over GF(2n): The TOE provides a Multiplication on an Elliptical Curve

ECDSA over GF(P): The TOE provides ECDSA over GF(P) cryptographic signature and verification

ECDSA over GF(2n): The TOE provides ECDSA over GF(2n) cryptographic signature and verification

HASH: The TOE provides Secure Hash (SHA) data signing capability

7.1.10.1 SFR to TSF_CRYPTO_SW

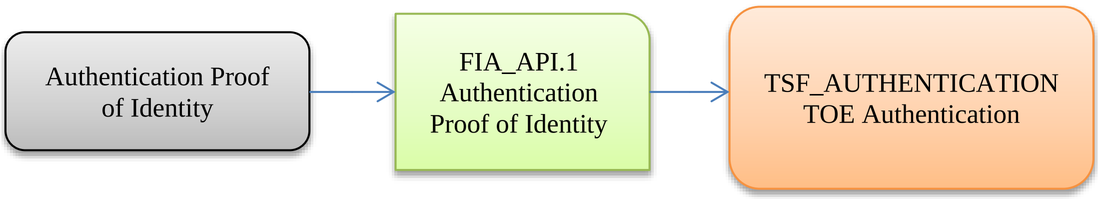


7.1.11 TSF_AUTHENTICATION Identification of the TOE

- Authentication

Authentication: The authentication of the TOE requires that the composite product manufacturer should perform a mutual authentication using SCP03 authentication mechanism to prove authentication and a "Get_Data" command over the secure bootloader to prove the identification of the TOE (Bootloader version, chip serial number that includes the hardware version in SNB0 and chip Id).

7.1.11.1 SFR to TSF_AUTHENTICATION



7.2 Rationale for TSF

This section demonstrates how the TSF contribute and work together to fulfil the SFR defined in section 6.

7.2.1 Summary of TSF to SFR

The following table gives an overview of the TSF that contribute to the SFRs.

		Security Functional Requirements																				
		Malfunctions		**Leakage**			**Physical Manipulation and Probing**			**Abuse of Functionality**		**Identification**	**Random Number Generation**	**Cryptography[23]**	**Proof of Identity**	**Abuse of Functionality**		**Controlled usage to Loader Functionality**				
		FRU_FLT.2	FPT_FLS.1	FDP_ITT.1	FPT_ITT.1	FDP_IFC.1	FPT_PHP.3	FDP_SDC.1	FDP_SDI.2	FMT_LIM.1	FMT_LIM.2	FAU_SAS.1	FCS_RNG.1	FCS_COP.1	FIA_API.1	FMT_LIM.1/Loader	FMT_LIM.2/Loader	FDP_ACC.1/Loader	FDP_ACF.1/Loader	FTP_ITC.1	FDP_UCT.1	FDP_UIT.1
TSF Features	TSF_TEST									X	X	X										
	TSF_ENV_PROTECT		X				X	X	X													
	TSF_LEAK_PROTECT			X	X	X																
	TSF_DATA_PROTECT	X					X	X	X													
	TSF_AUDIT_ACTION		X						X													
	TSF_RNG			X	X	X							X									
	TSF_CRYPTO_HW													X								
	TSF_CRYPTO_SW												X	X								
	TSF_AUTHENTICATION														X							
	TSF_LOADER																	X	X	X	X	X
	TSF_FLASH															X	X					

Table 10 - Dependencies of the TOE Security Features

[23] *Refer to Table 6 which gives Cryptographic Functions Overview*

7.2.2 Rationale for the TSF Features of the TOE

The justification for the SFRs relating to Malfunctions that is FRU_FLT.2 and FPT_FLS.1 is as follows:

The SFR "FRU_FLT.2 Limited fault tolerance" and "FPT_FLS.1 Failure with preservation of secure state" relate to the TSF Features "TSF_ENV_PROTECT Environmental Protection", "TSF_AUDIT_ACTION Event Audit and Action" and "TSF_DATA_PROTECT". The TSF_ENV_PROTECT defines an operating window that the TOE safely works within. If the TOE is subjected to operating conditions out-with this operating range, the TSF mechanisms of TSF_ENV_PROTECT (Voltage Monitor, Frequency Monitor, and Temperature Monitor) will set a violation through TSF_AUDIT_ACTION mechanism Security Registers. The mechanism Reset System can take appropriate action to ensure the TOE fails in a secure state (FPT_FLS.1).

The justification for the SFRs relating to Abuse of Functionality that is FMT_LIM.1 and FMT_LIM.2 is as follows:

The SFR "FMT_LIM.1 Limited capabilities" and "FMT_LIM.2 Limited availability" relates to the TSF Feature TSF_TEST. The security mechanism Test Mode Entry protects the test feature.

The justification for the SFRs relating to Authentication of Proof of Identity that is FIA_API.1 is as follows:

The SFR "FIA_API.1 Authentication of Proof of Identity" relates to the TSF feature TSF_AUTHENTICATION, this provides proof of the identity of the TOE, an object or an authorized user or role to an external entity. The "Get_Data" command over the secure bootloader can be used to prove the identification of the TOE.

The justification for the SFRs relating to Controlled usage to Loader Functionality that is FDP_ACC.1, FDP_ACF.1, FTP_ITC.1, FDP_UCT.1, and FDP_UIT.1 is as follows:

The SFR "FDP_ACC.1 Subset access control", "FDP_ACF.1 Security attribute based access control", "FTP_ITC.1 Inter-TSF trusted channel", "FDP_UCT.1 Basic data exchange confidentiality", and "FDP_UIT.1 Data exchange integrity" relates to the TSF feature TSF_LOADER. The mutual authentication is covered by the mutual authentication mechanism and the trusted channel.

The justification for the SFRs relating to Identification that is FAU_SAS.1 is as follows:

The SFR "FAU_SAS.1 Audit Storage" relates to the TSF Feature TSF_TEST. The mechanism Serial Number Register Write allows the storage of Initialisation data, Pre-personalisation data, supplements of the Security IC Embedded Software and unique identification information for the TOE when in Test Mode.

The justification for the SFRs relating to Physical Manipulation and Probing that is FPT_PHP.3, FDP_SDC.1 and FDP_SDI.2 is as follows:

The SFR "FPT_PHP.3 Resistance to physical attack", "FDP_SDC.1 Stored Data Confidentiality" and "FDP_SDI.2 Stored data integrity monitoring and action" relate to the TSF Features TSF_ENV_PROTECT and TSF_DATA_PROTECT. The mechanisms of TSF_ENV_PROTECT (Voltage Monitor, Frequency Monitor and Temperature Monitor) detect when the TOE has been manipulated to try to operate it out-with its operating conditions. To protect against direct probing using galvanic contacts, the mechanisms of TSF_ENV_PROTECT prevent this attack. Hardware Protection has an active shield that is monitored to detect any violation or removal. The Structure and Layout and also Bus Encryption make any attempt to identify important structures difficult for an attacker. Any attempt to directly identify or probe memory contents is also made difficult through the mechanism Memory Encryption of TSF_ENV_PROTECT. Attempts to probe the TOE in an indirect way (without galvanic contacts) for example using a laser to identify registers are countered by the TSF_ENV_PROTECT mechanism Light Scan Detector and the TSF_DATA_PROTECT mechanism Register Mirroring. Attempts to use Fault Injection to indirectly probe or gather information from the memory contents is countered by the mechanism Memory Encryption of TSF_ENV_PROTECT and the mechanisms Secure Memory Management, CRC Accelerator, Parity Checker ROM/RAM/Registers, Register Mirroring, Enhanced Protection Object, Program Stack Checker, Glitch Detectors of TSF_DATA_PROTECT. In addition the monitoring and action requirement of “FDP_SDI.2 Stored data integrity monitoring and action” is met by TSF_AUDIT_ACTION which will set a violation and allow embedded software to react as required.

The justification for the SFRs relating to Leakage that is FDP_ITT.1, FPT_ITT.1 and FDP_IFC.1 is as follows:

The SFR “FDP_ITT.1 Basic internal transfer protection”, “FPT_ITT.1 Basic internal TSF data protection” and “Subset information flow control” relates to TSF_LEAK_PROTECT and TSF_RNG. The TSF feature TSF_LEAK_PROTECT provides mechanisms to help prevent side channel analysis through both power and electromagnetic emissions. The mechanisms Dummy Interrupt, Random Branch Insertion, VFO Jitter, and Frequency Divider, help spread the information content of any power signature emanating from the TOE. The mechanism Dummy NVM write allows the Security IC Embedded Software to mask when the NVM is being written. The Internal Clock (VFO) mechanism helps prevent any clock pulse synchronisation to aid the attacker when setting up or timing the study of the emanations. The mechanism True RNG of TSF_RNG can also be used to add noise to the leakage from the TOE. The TSF Features TSF_LEAK_PROTECT and TSF_RNG combine to comply with the Data Processing Policy defined by FDP_IFC.1.

The justification for the SFR relating to Random Number Generation FCS_RNG.1 for is as follows:

The SFR “FCS_RNG.1 Random number generation” relates to TSF_RNG and TSF_CRYPTO_SW. The SFR requires a physical random number generator this is provided by the mechanism True RNG of TSF_RNG. The total failure test of the noise source is provided by the mechanism RNG Status Register. If the Security IC Embedded Software requires performing an online test of the random data FCS_COP.1 provides the mechanism AIS31 Online Test (see Table 11). FCS_RNG.1 also requires the random data to be compliant to a quality metric - TSF_RNG allows data to be gathered using the mechanism RNGDAS for AIS31 compliant data. It is also possible for the end user of the TOE to apply post processing to the random data and gather the resulting data through mechanism RDWDR.

The justification for the SFRs relating to Abuse of Functionality for a Loader that is FMT_LIM.1/Loader and FMT_LIM.2/Loader is as follows:

The SFR “FMT_LIM.1 Limited capabilities” and “FMT_LIM.2 Limited availability” relates to the TSF Feature TSF_FLASH. FLASH Lock mechanism once enabled disables the Loader mechanism.

The justification for the SFR relating to Cryptography FCS_COP.1 is as follows:

The SFR “FCS_COP.1 Cryptographic Operation” relates to TSF_CRYPTO_HW and TSF_CRYPTO_SW. The SFR requires cryptographic operations to be performed with certain key lengths and to a specific standard. To understand how the mechanisms of the TSF features contribute to this, a map is shown in Table 11.

FCS_COP.1 requirement	TSF Feature	Mechanism
/TDES	TSF_CRYPTO_HW	Triple DES
/AES	TSF_CRYPTO_HW	AES
/SHA-1	TSF_CRYPTO_SW	Secure Hash (SHA-1)
/SHA-224	TSF_CRYPTO_SW	Secure Hash (SHA-224)
/SHA-256	TSF_CRYPTO_SW	Secure Hash (SHA-256)
/SHA-384	TSF_CRYPTO_SW	Secure Hash (SHA-384)
/SHA-512	TSF_CRYPTO_SW	Secure Hash (SHA-512)
/RSA without CRT	TSF_CRYPTO_SW	RSA Without CRT PrimeGen
/RSA with CRT	TSF_CRYPTO_SW	RSA with CRT PrimeGen
/ECDSA over Zp	TSF_CRYPTO_SW	ECDSA over Zp
/EC-DH over Zp	TSF_CRYPTO_SW	EC-DH over Zp
/ECDSA over GF(2n)	TSF_CRYPTO_SW	ECDSA over GF(2n)
/EC-DH over GF(2n)	TSF_CRYPTO_SW	EC-DH over GF(2n)
N/A (support for FCS_RNG.1)	TSF_CRYPTO_SW	AIS31 Online test
/Lucas Test	TSF_CRYPTO_SW	Lucas Test
/PrimeGen	TSF_CRYPTO_SW	Prime Generation

Table 11 - Cryptographic Functions Overview

7.2.3 Note on ADV_ARC.1

The Assurance component ADV_ARC.1 states that the TOE should be self-protected against any tampering or bypassing of the TSF of the TOE.

The TSF Features TSF_ENV_PROTECT, TSF_AUDIT_ACTION and TSF_DATA_PROTECT contain mechanisms that fully protected the TOE against any external tamper or bypass.

The Security Mechanisms applicable to this protection are:

- Hardware Protection (Active Shield)
- Voltage Monitor
- Frequency Monitor
- Temperature Monitor
- Glitch Detectors
- Memory Encryption
- Reset System

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 un-personalised 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 the form of wafers or sawn wafers (dice,) he has the role of the IC Packaging Manufacturer (Phase 4) in addition.
End-consumer	The customer of the TOE Manufacturer who receives the TOE during TOE Delivery. The Composite Product Manufacturer includes the Security IC Embedded Software developer and all roles after TOE Delivery up to Phase 6. User of the Composite Product in Phase 7.
IC Dedicated Software	IC proprietary software embedded in a Security IC (also known as IC firmware) and developed by the IC Developer. Such software is required for testing purpose (IC Dedicated Test Software) but may provide additional services to facilitate usage of the hardware and/or to provide additional services (IC Dedicated Support Software).
IC Dedicated Test Software	That part of the IC Dedicated Software (refer to above) which is used to test the TOE before TOE Delivery but which does not provide any functionality thereafter.
IC Dedicated Support Software	That part of the IC Dedicated Software (refer to above) which provides functions after TOE Delivery. The usage of parts of the IC Dedicated Software might be restricted to certain phases.
Initialisation Data	Initialisation Data defined by the TOE Manufacturer to identify the TOE and to keep track of the Security IC's production and further life-cycle phases are considered as belonging to the TSF data. These data are for instance used for traceability and for TOE identification (identification data).
Integrated Circuit (IC)	Electronic component(s) designed to perform processing and/or memory functions.

Pre-personalisation Data	Any data supplied by the Card Manufacturer that is programmed into the non-volatile memory by the Integrated Circuits manufacturer (Phase 3). This data is for example used for traceability and/or to secure shipment between phases.
Security IC	(as used in this Protection Profile) Composition of the TOE, the Security IC Embedded Software, User Data and the package (the Security IC carrier).
Security IC Embedded Software	The 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
Test Features	All features and functions (implemented by the IC Dedicated Test Software and/or hardware) which are designed to be used before TOE Delivery only and delivered as part of the TOE.
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 E^2PROM) 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 Literature

[CC_PART1]

Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 5, April 2017

[CC_PART2]

Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1, Revision 5, April 2017

[CC_PART3]

Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 5, April 2017

[CEM]

Common Methodology for Information Technology Security Evaluation (CEM), Part 2: Evaluation Methodology; Version 3.1, Revision 5, April 2017

[JHAS]

Supporting Document, Mandatory Technical Document: Application of Attack Potential to Smartcards, April 2019, Version 3.0

[COMP]

Supporting Document: Composite product evaluation for Smart Cards and similar devices, version 1.5.1, May 2018

[PP]

Security IC Platform Protection Profile with Augmentation Packages, BSI-CC-PP-0084-2014, V1.0

[ANSSI-PP0084]

PP0084 : Interpretations, PP0084.03, 1st June 2016

[AIS31]

AIS31: Functionality classes for random number generators, Version 2.0, 18. September 2011

[AUG]

Smartcard Integrated Circuit Augmentations Version 1.0, March 2002, registered under the German Certification Scheme BSI-AUG-2002

8.3 List of Abbreviations

CC	Common Criteria.
EAL	Evaluation Assurance Level.
IC	Integrated circuit.
IT	Information Technology.
PP	Protection Profile.
ST	Security Target.
TOE	Target of Evaluation.
TSC	TSF Scope of Control.
TSF	TOE Security Functionality.

Headquarters	Product Contact		
WISeKey	*Web Site*	*Technical Support*	*Sales Contact*
Arteparc Bachasson, Bat A Rue de la carrier de Bachasson CS70025 13590 Meyreuil France Tel: +33 (0)4-42-370-370 Fax: +33 (0)4-42-370-024	**www.wisekey.com**	**dl-ms6xxx@wisekey.com**	**sales@wisekey.com**