MN67S150 Smart Card IC Security Target (ST-Lite; BSI-DSZ-CC-0935-2015)

Version: 1.8

Date: 9 March 2015

Panasonic Semiconductor Solutions Co., Ltd.

Document History

Version	Date	Changes
1.0	2013-11-25	Public version
1.1	2013-12-20	1. Modification of version and form of delivery of guidance documents (Section 1.2: Table 1) 2. Addition of the description of TOE cryptographic functionality (Section 6.1.2) 3. Deletion of the note (Section 6.1.2 (1)) 4. Revision of the explanation of “Note” (Section 6.1.2 (1) (c)) 5. Revision of the explanation (Section 7.3.5, 7.3.8) 6. Revision of the referenced documents (Section 8.3)
1.2	2014-03-13	1. Addition of the definition of ADV_SPM (Section 6.2.2)
1.3	2014-04-22	1. Revision of the RV (Section 1.2: Table 1)
1.4	2014-06-27	1. Revision of TOE name and deletion of the TOE version (Section 1.2) 2. Revision of the RV (Section 1.2: Table 1)
1.5	2014-07-23	1. Revision of the table (Section 1.2: Table1) 2. Revision of the abbreviated name of guidance (Section 8.3)
1.6	2014-07-31	1. Revision of the table (Section 1.2: Table 1) 2. Revision of the table (Section 6.1.2: Table 6) 3. Addition of [FIPS46-3] reference (Section 8.3)
1.7	2014-12-04	1. Revision of the version of guidance document (Section 1.2: Table 1)
1.8	2015-03-09	1. Revision of the title (cover, header, Section 1.1)

Table of Contents

1 ST Introduction

1.1 ST Reference

Title:	MN67S150 Smart Card IC Security Target (ST-Lite; BSI-DSZ-CC-0935-2015)
Version:	Version 1.8
Date:	9 March 2015
Produced by:	Panasonic Semiconductor Solutions Co., Ltd.
Author:	Mitsuyoshi Ohya
CC version used:	Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 4, September 2012, CCMB-2012-09-001. Part 2: Security Functional Requirements, Version 3.1, Revision 4, September 2012, CCMB-2012-09-002. Part 3: Security Assurance Requirements, Version 3.1, Revision 4 September 2012, CCMB-2012-09-003. (CC V3.1), part 1 to 3
PP used:	Security IC Platform Protection profile PP0035, Version 1.0, BSI-CC-PP-0035-2007, 2007-06-15

This document is compiled from MN67S150 Smart Card IC Security Target (BSI-DSZ-CC-0935-2015) as public version (hereafter ST-Lite). Proprietary information (e.g. about design) is removed in accordance with regulations of [JIL]

1.2 TOE Reference

TOE:	MN67S150 Smart Card IC Version RV08 including IC Dedicated Software
Developed by:	Panasonic Semiconductor Solutions Co., Ltd.

The TOE consists of:

Table 1: TOE identification

Item Type	Name	Version	Form of delivery
Hardware	MN67S150 Smart Card IC	RV08	Sawn wafers (dice)
Software	MN67S150 Smart Card IC - IC Dedicated Software	FV0C	Encrypted in electronic form, Object file (.rf), executable format file (.ex) or HEX format file (.hex)
Document	[AGD-SES]	1.4	Encrypted in electronic form
	[AGD-CM]	1.2	
	MN101C/MN101E Series Installation Manual	11640-080E	
	MN101C/MN101E/MN103L Series In-Circuit Emulator Installation Manual	19940-015E	
	PCI/PC Card Installation Manual	19942-101E	
	MN101C00 Series LSI User's Manual	21499-030E	
	Debug Factory Builder Version 4 Tutorial	1999002-010	
	MN101C/MN101E Series Cross Assembler User's Manual	11410-230E	
	MN101C Series Instruction Manual	11450-041E	
	MN101C/MN101E Series C Compiler User's Manual Library Reference	11422-060E	
	MN101C/MN101E Series C Compiler User's Manual Language Description	11421-090E	
	MN101C/MN101E/MN101L Series C Compiler User's Manual Usage Guide	11420-210E	
	MN67S150 Software Library Specification	1.20	

1.3 TOE Overview

1.3.1 TOE Class and main security function

The TOE is the smart card integrated circuit (IC) called MN67S150, developed by Panasonic Semiconductor Solutions Co., Ltd. using 0.18µm process. TOE is composed of hardware including a processing unit, Cryptographic Hardware, security components, RF interface, and volatile and non-volatile memories. The TOE also includes IC Dedicated Software and documentation. The IC Dedicated Software is used for test purposes during production but also provide additional services to facilitate usage of hardware.

The IC is delivered in form of sawn wafers (dice). After making into module by Composite Product Manufacturer, it is embedded in a credit card-sized plastic package.

The TOE is intended to be used for the applications requiring high security such as transportation and fare collection applications (the Commuter ticket), access control applications (ID cards), and government applications (the Basic Resident Register, health cards and driver license).

The security features implemented by the MN67S150 are:

- True random number generator;
- Security sensors (temperature, frequency, voltage, light);
- Physical countermeasures (such as sensing shield);
- Cryptography (Triple-DES, DES, AES); and
- Countermeasures against DFA, DPA, and SPA attacks.

In addition, the security of the development and manufacturing environments has been designed to provide high assurance in the security of the MN67S150 product right through to its delivery to customers.

1.3.2 Required non-TOE hardware/software/firmware

The TOE requires a reader/writer device which supplies the power and performs transmission and reception of data commands via the protocol defined in [JISX6319-4].

1.4 TOE Description

1.4.1 TOE Physical Scope

The Target of Evaluation (TOE) is a *Smart card integrated circuit* which is composed of hardware such as a processing unit, Cryptographic Hardware, security components, RF Interface and volatile and non-volatile memories (Figure 1). The TOE also includes IC Designer/Manufacturer proprietary *IC Dedicated Software* (Figure 2). Such software (also known as IC firmware) is used for test purposes during production but also provides additional services to facilitate usage of hardware. In addition to the IC Dedicated Software the Smart Card Integrated Circuit also includes hardware to perform testing. All other software is called Security IC Embedded Software, which is not part of the TOE.

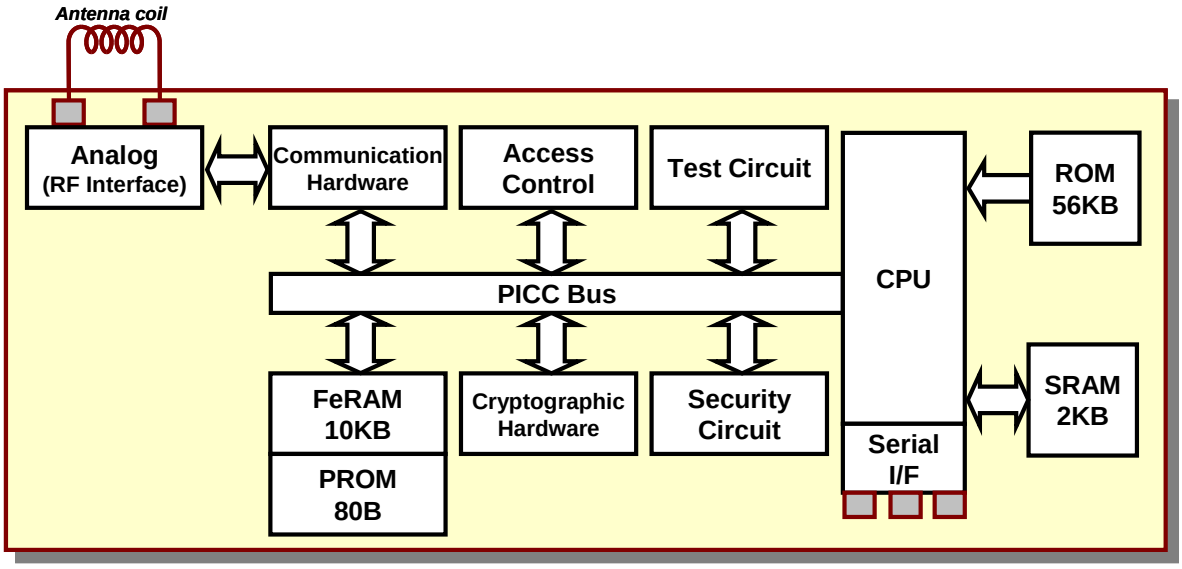


Figure 1: Block Diagram 1 of MN67S150 Smart Card IC - Hardware -

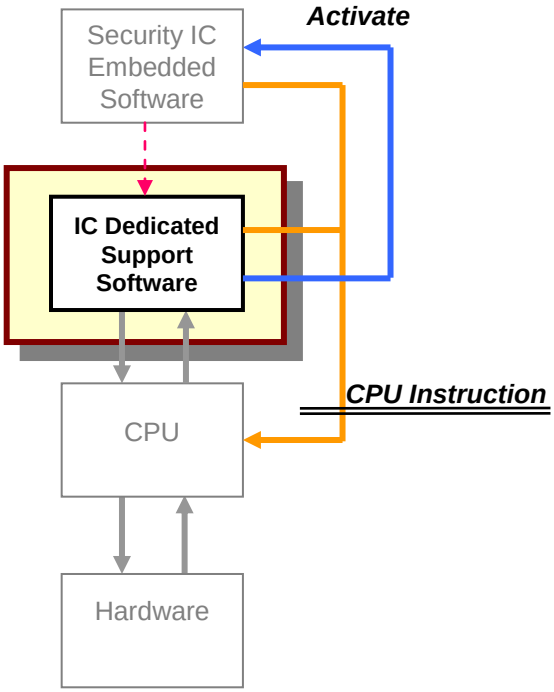


Figure 2: Block diagram 2 of MN67S150 Smart Card IC -IC Dedicated Software in Normal operation-

1.4.1.1 Hardware

As depicted in Figure 1, the TOE includes the following components.

(1) Analog

Analog is RF Interface in conformity to [JISX6319-4], and realizes the following functions.

- Power reception using a rectifier
- Demodulation of ASK-modulated receive signals
- Transmission of modulated signals using a load switch
- Generation of stabilized power supply voltage VDD
- Generation of FeRAM supply voltage VPP
- Generation of reference clock signal from 13.56MHz carrier
- Generation of power-on reset signal

Besides, it contains various security logics such as RNG, random current generator, sensor/filter, and sensing shield.
The generated random numbers are used internally and can be used by the Security IC Embedded Software for e.g. the generation of cryptographic keys.
Sensor/filter includes the followings.

- Voltage sensors (High & low)
- Voltage glitch sensor
- Low frequency sensor
- Light sensor
- Clock filters (High frequency & glitch)
- Reset filter
- Temperature sensors (High & low)

The sensing shield covers the whole chip surface with shield lines, which are connected to sensors.

(2) Communication Hardware

The Communication Hardware controls data transmission/reception from/to RF Interface in conformity to [JISX6319-4].

(3) Memory

The device has memories consisting of the following:

- ROM: 56Kbytes
- SRAM: 2Kbytes
- FeRAM: 10Kbytes, PROM: 80Bytes

In ROM, IC Dedicated Software and Security IC Embedded Software are stored. FeRAM can be accessed as both data memory and program memory. In PROM, data, not allowed to be overwritten is stored.

(4) Cryptographic Hardware

The Cryptographic Hardware is capable of realizing AES functionality and DES/TDES.

(5) Access Control

There are two modes for access control, and areas to which accesses are possible vary depending on each mode.

- User mode
- API mode

(6) Security Circuit

Security Circuit contains various control circuits to control the Security logic (refer to 1.4.1.1(1)).

(7) CPU

CPU contains the Core AM13E, Interrupt function that processes interrupt.

The main features of AM13E CPU are:

- Simple and highly efficient instruction set
  (Number of basic instructions: 37; number of addressing modes: 9)
- Configuration that can increment variable instruction length by 4 bits based on minimum instruction length of 1 byte
- Minimum instruction execution time of 1 clock cycle
- Support for linear address space of up to 256KBytes

The interrupt function speeds up interrupt response with circuitry that automatically loads the branch address to the corresponding interrupt processing program from an interrupt vector table, and processes non-maskable interrupts (NMI) and level interrupts.

(8) PICC Bus

Via the PICC Bus data between CPU and each device (Test Circuit, FeRAM, Cryptographic Hardware, Communication Hardware, Access Control, and Security Circuit) are exchanged.

(9) Test Circuit

Test Circuit controls Test mode operation to execute the manufactural defective tests of

IC during Phase 3.

1.4.1.2 Firmware and Software

The TOE includes the following IC Dedicated Software stored in ROM. It consists of IC Dedicated Support Software and IC Dedicated Test Software.

Table 2: IC Dedicated Software

Sorting of IC Dedicated Software	Purpose
IC Dedicated Support Software	To facilitate the use of hardware
IC Dedicated Test Software	To execute the functional tests after production

The Security IC Embedded Software is not part of the TOE but the interface for delivery of it is included in the TOE.

1.4.1.3 Interface of the TOE

(1) Electrical Interface / data interface

The electrical interface of the TOE to the external environment is the coil pads to which the RF antenna is connected.
Besides, the pads which are used at the test execution at Phase 3 are also electrical interface.

(2) Hardware Interface

For the interface to hardware, there is CPU Instruction Set.

(3) Firmware Interface

There are the following interfaces according to the modes.

- In the Test mode (during test at Phase 3)
  - In case that the functionality is tested directly from pad: None
  - In case that the test is conducted by using the Contact Test Software: test instruction set
- In the Normal mode (API mode or User mode)
  - The set of function for controlling hardware

(4) Software Interface

For Software Interface, there is the Security IC Embedded Software main function call from IC Dedicated Support Software.

(5) Physical Interface

Although not used for normal operation, the IC surface is an additional physical interface of the TOE that might be used by an attacker.

(6) Test Pads

The pads which are used at the test execution at Phase 3 are also electrical interface.

1.4.1.4 Guidance Documentation

The TOE includes the following guidance documentation:

- [Guide-SES]: This documentation is provided for users who develop Security IC Embedded Software.
- [Guide-CM]: This documentation is provided for users who manufacture card using TOE.

1.4.2 TOE Life Cycle

As described in [PP, 1.2.3 & 7.1.1], the life cycle of TOE is separated into 7 phases.

Phase 1: Security IC Embedded Software Development
Phase 2: IC Development
Phase 3: IC Manufacturing and Testing
Phase 4: IC Packaging
Phase 5: Security IC Product Finishing Process
Phase 6: Security IC Personalization
Phase 7: Security IC End-usage

This Security Target addresses Phase 2-3. This also includes the interfaces to the other phases where information and material is being exchanged with the partners of the development/manufacturer of the TOE.

The IC is delivered in form of sawn wafer (dice) after the production test. TOE delivery can therefore be at the end of Phase 3.

1.4.2.1 TOE Logical Phases

Just after the power-on, the IC is in Normal mode. The IC can enter the test mode by the predefined procedures. When the power is off, the IC returns to the Normal mode. If all the requested tests are successfully done, the transition to the Test mode falls into disuse by the predefined control.

1.5 TOE Environments

The development and manufacturing environments of the TOE are separated into four areas.

- Design sites
- Mask manufacture site
- Manufacturing sites
- Defective products processing site

1.5.1 TOE Development Environment

1.5.1.1 Design sites

Panasonic's design sites are managed as defined in the "Information Security Management System Manual" for the following confidential information.

Table 3: Confidential information at the design site

Confidential information	
Assets (related to standard functionality)	The Security IC Embedded Software
	The security services provided by the TOE for the Security IC Embedded Software
Assets (Security service)	The generation of random numbers by means of a physical Random Number Generator
Critical information to protect above assets	Logical design data
	Physical design data
	IC Dedicated Software
	Configuration data
	Pre-personalization Data
	Specific development aids
	Test and characterization related data
	Material for software development support
	Wafer/Development samples for testing
	Related documentation

Clearly defined physical, personnel, and IT processes and procedures within the scope of evaluation ensure the security in the development environment.

1.5.2 TOE Production Environment

1.5.2.1 Mask Manufacture site

Mask manufacturer subcontracted with Panasonic is forced to securely handle the following confidential information with NDA.

- MN67S150 mask processing data (EB data)
- Photomasks
- Related documentation

1.5.2.2 Manufacturing sites

In Panasonic's manufacturing sites the following confidential information is managed

securely as defined in the “Information Security Management System Manual”

- Pre-personalization data
- Masks and wafers (including sawn wafer),
- Test and characterization related data
- Wafer/Development samples for testing
- Wafer/Chip defectives
- Related documentation

As with in the development environment, clearly defined processes and procedures ensure security in the production environment.

1.5.2.3 Defective Products processing site

Defective products processing company subcontracted with Panasonic is forced to securely handle the following confidential information with NDA.

- Wafer/Chip defectives
- Related documentation

The wafer/chip defectives are transported from manufacturing sites to defective products processing company, and securely discarded.

1.5.3 Initialization and pre-personalization Data

During testing at Phase 3, certain data to uniquely identify the IC is injected in the write lock area of FeRAM.

2 Conformance Claims

2.1 CC Conformance Claim

This Security Target and the TOE claim conformance to Common Criteria for Information Technology Security Evaluation; Version 3.1, revision 4, Part 1, Part2, and Part 3.
This Security Target claims conformance for:
Common Criteria Part 2 extended and Common Criteria Part 3 conformant.

2.2 PP claim

This Security Target claims strict conformance to the following Protection Profile.

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

2.3 Package Claim

This Security Target does not claim conformance to a package of the [PP].
The assurance level is **EAL6 augmented** with the following components:

- ASE_TSS.2

2.4 Conformance Rationale

In this Security target, strict conformance to [PP] is claimed. This is fulfilled by including all the security objectives and requirements from [PP] (as shown in the relevant sections). The additional aspects added in this ST are consistent with [PP] as argued in section 6.3, and hence no further rationale is required.

The followings are the additional assumptions, organizational security policies, security objectives, and requirements added in this Security target.

PP addition	Added section	Rationale section
T.Mem-Access	to Section 3.2.3	Section 4.4
P.Add-Functions	to Section 3.3.2	
A.Key-Function	to Section 3.4.2	
A.Interpreter		
OE.Interpreter	to Section 4.2.1.2	
O.Mem-Access	to Section 4.1.3	
O.Add-Functions		
FCS_COP.1/TDES	to Section 6.1.2 (1) (a)	Section 6.3.1
FCS_COP.1/AES	to Section 6.1.2 (1) (b)	
FCS_COP.1/DES	to Section 6.1.2 (1) (c)	
FDP_ACC.1	to Section 6.1.2 (2)	
FDP_ACF.1		
FMT_MSA.3		
FMT_MSA.1		
FMT_SMF.1		
FCS_RNG.1	to Section 6.1.2 (3)	Section 6.3.2
ASE_TSS.2	to Section 7.3	Section 6.3.4

3 Security Problem Definition

The assets, threats, organizational security policies, and assumptions given in [PP] apply to the MN67S150. The description below is therefore adopted from [PP, 3].
In addition, the MN67S150 implements cryptographic functions for which relevant threats, organizational security policies, and assumptions have been added.

3.1 Description of Assets

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

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

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

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

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

[PP] requires the TOE to provide one security service: the generation of random numbers by means of a physical Random Number Generator. The Security Target may require additional security services. It is essential that the TOE ensures the correct

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

According to [PP] there is the following high-level security concern related to security service:

SC4 deficiency of random numbers.

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

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

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

Note that there are many ways to manipulate or disclose the User Data: (i) An attacker may manipulate the Security IC Embedded Software or the TOE. (ii) An attacker may cause malfunctions of the TOE or abuse Test Features provided by the TOE. Such attacks usually require design information of the TOE to be obtained.
They pertain to all information about (i) the circuitry of the IC (hardware including the physical memories), (ii) the IC Dedicated Software with the parts IC Dedicated Test Software (if any) and IC Dedicated Support Software (if any), and (iii) the configuration data for the security functionality. The knowledge of this information enables or supports attacks on the assets. Therefore the TOE Manufacturer must ensure that the development and production of the TOE (refer to [PP, 1.2.3]) is secure so that no information is unintentionally made available for the operational phase of the TOE.

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

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

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

as long as they are generated, stored, or processed by the TOE Manufacturer. Explanations can be found in [PP, 7.1.2].

3.2 Threats

3.2.1 Standard Threats from [PP]

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

T.Leak-Inherent Inherent Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential User Data as part of the assets.

No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. One example is Differential Power Analysis (DPA). This leakage may be interpreted as a covert channel transmission but is more closely related to measurement of operating parameters, which may be derived either from direct (contact) measurements (Numbers 6 and 7 in [PP, Figure 8) or measurement of emanations (Number 5 in [PP, Figure 8) and can then be related to the specific operation being performed.

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

T.Phys-Probing Physical Probing

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

disclosing or manipulating the User Data or the Security IC Embedded Software.

Physical probing requires direct interaction with the Security IC internals (Numbers 5 and 6 in [PP, Figure 8]). Techniques commonly employed in IC failure analysis and IC reverse engineering efforts may be used. Before that hardware security mechanisms and layout characteristics need to be identified (Number 3 in [PP, Figure 8]). Determination of software design including treatment of User Data may also be a prerequisite.

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

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

T.Malfunction Malfunction due to Environmental Stress

An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to (i) modify security services of the TOE or (ii) modify functions of the Security IC Embedded Software (iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the User Data or the Security IC Embedded Software. This may be achieved by operating the Security IC outside the normal operating conditions (Numbers 1, 2 and 9 in [PP, Figure 8]).

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

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

T.Phys-Manipulation Physical Manipulation

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

The modification may be achieved through techniques commonly employed in IC failure analysis (Numbers 1, 2 and 4 in [PP, Figure 8)) and IC reverse engineering efforts (Number 3 in [PP, Figure 8)). The modification may result in the deactivation of a security feature. Before that hardware security mechanisms and layout characteristics need to be identified. Determination of software design including treatment of User Data may also be a pre-requisite. Changes of circuitry or data can be permanent or temporary.

In contrast to malfunctions (refer to T.Malfunction) the attacker requires to gather significant knowledge about the TOE's internal construction here (Number 3 in [PP, Figure 8]).

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

T.Leak-Forced Forced Information Leakage

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

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

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

T.Abuse-Func Abuse of Functionality

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

manipulating the User Data or the Security IC Embedded Software.

3.2.2 Threats related to Security Services (from [PP])

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

T.RND Deficiency of Random Numbers

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

An attacker may gather information about the random numbers produced by the TOE security service. Because unpredictability is the main property of random numbers this may be a problem in case they are used to generate cryptographic keys. Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the TOE. Malfunctions or premature ageing are also considered which may assist in getting information about random numbers.

3.2.3 Augmented Threats

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

T.Mem-Access Memory Access Violation

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

3.3 Organizational Security Policies

3.3.1 Organizational Security Policies from [PP]

The following organizational Security Policy is taken from [PP, 3.3].

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

P.Process-TOE Protection during TOE Development and Production

An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.

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

3.3.2 Augmented Organizational Security Policies

The following specific security functionality is listed which is not derived from threats identified for the TOE's environment because it can only be decided in the context of the smart card application, against which threats the Security IC Embedded Software will use the specific security functionality.

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

P.Add-Functions Additional Specific Security Functionality

The TOE shall provide the following specific security functionality to the Security IC Embedded Software:

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

3.4 Assumptions

3.4.1 Assumptions from [PP]

The following descriptions and assumptions are taken from [PP, 3.4].

The intended usage of the TOE is twofold, depending on the Life Cycle Phase: (i) The Security IC Embedded Software developer uses it as a platform for the Security IC software being developed. The Composite Product Manufacturer (and the consumer) uses it as a part of the Security IC. The Composite Product is used in a terminal which supplies the Security IC (with power[1]) and (at least) mediates the communication with the Security IC Embedded Software.

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

[1] In case of contactless card the terminal does not supply the clock.

Appropriate "Protection during Packaging, Finishing and Personalization (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 Personalization

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 unauthorized use).

This means that the Phases after TOE Delivery (refer to [PP, 1.2.2 & 7.1]) are assumed to be protected appropriately. For a preliminary list of assets to be protected refer to the following paragraph.

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-personalization and personalization data including specifications of formats and memory areas, test related data,
- the User Data and related documentation, and
- material for software development support

as long as they are not under the control of the TOE Manufacturer.

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

A.Plat-Appl Usage of Hardware Platform

The Security IC Embedded Software is designed so that the requirements from the following documents are met:

(i) *[Guide-SES],* and

(ii) Findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the

certification report.

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

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

A.Resp-Appl Treatment of User Data

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

The application context specifies how the User Data shall be handled and protected.
The evaluation of the Security IC according to [PP] is conducted on generalized application context. The concrete requirements for the Security IC Embedded Software shall be defined in [PP] respective Security Target for the Security IC Embedded Software. The Security IC can not prevent any compromise or modification of User Data by malicious Security IC Embedded Software. The assumption A.Resp-Appl ensures that the Security IC Embedded Software follows the security rules of the application context. Examples are given in [PP, 7.2.1], all being directly related to and covered by A.Resp-Appl.

3.4.2 Augmented Assumption

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

A.Key-Function Usage of Key-dependent Function

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.

The developer of Security IC Embedded Software must ensure the appropriate "Implementation of command interpreter (A.Interpreter)" while developing this software in Phase 1 as specified below.

A.Interpreter Implementation of command interpreter

It is assumed that the command interpreter used for the tests in Phase 4 and 5 shall be implemented.
To prevent that an attacker abuses the test commands, it is required to authenticate sufficiently before executing these test commands. Besides, the test commands shall be deactivated after completing all of the tests.

4 Security Objectives

The security objectives described below are drawn from [PP, 4].

4.1 Security Objectives for the TOE

4.1.1 Security Objectives for the TOE from [PP]

The following Security Objectives for the TOE are taken from [PP, 4.1].

There are the following standard high-level security goals related to the assets:

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

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

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

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

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

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.

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.

Standard Security Objectives

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

O.Leak-Inherent Protection against Inherent Information Leakage

The TOE must provide protection against disclosure of confidential data stored and/or processed in the Security IC

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

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

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

O.Phys-Probing Protection against Physical Probing

The TOE must provide protection against disclosure of User Data, against the disclosure/reconstruction of the Security IC Embedded Software or against the disclosure of other critical information about the operation of the TOE. This includes protection against

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

with a prior

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

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

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

O.Malfunction Protection against Malfunctions

The TOE must ensure its correct operation.

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

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

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

O.Phys-Manipulation Protection against Physical Manipulation

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

- reverse-engineering (understanding the design and its properties and functions),

- manipulation of the hardware and any data, as well as

- controlled manipulation of memory contents (Application Data).

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

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

O.Leak-Forced Protection against Forced Information Leakage

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

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

- by a physical manipulation (refer to "Protection against Physical Manipulation (O.Phys-Manipulation)".

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

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

O.Abuse-Func Protection against Abuse of Functionality

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

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

O.Identification TOE Identification

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

4.1.2 Security Objectives related to Specific Functionality (referring to SG4)

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

O.RND Random Numbers

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

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

4.1.3 Augmented Security Objectives for the TOE

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

O.Add-Functions Additional Specific Security Functionality

The TOE must provide the following specific security functionality to the Security IC Embedded Software.

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

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

O.Mem-Access Area based Memory Access Control

The TOE must provide the following specific security functionality to the Security IC Embedded Software. The TOE must provide the Security IC Embedded Software with the capability to define restricted access memory areas. The TOE must then enforce the partitioning of such memory areas so that

access of software to memory areas is controlled as required, for example, in a multi-application environment.

4.2 Security Objectives for the Security IC Embedded Software development Environment

In this section Security Objectives for Security IC Embedded Software development Environment are taken from [PP, 4.2]. But to explicitly cover cryptographic algorithms the clarification are made for OE.Plat-Appl and OE.Resp-Appl.
Furthermore OE.Interpreter is added as TOE-specific objective.

4.2.1 Phase 1

4.2.1.1 Security Objectives from [PP]

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

OE.Plat-Appl Usage of Hardware Platform

To ensure that the TOE is used in a secure manner the Security IC Embedded Software shall be designed so that the requirements from the following documents are met:

(i) hardware data sheet for the TOE,

(ii) data sheet of the IC Dedicated Software of the TOE,

(iii) TOE application notes, other guidance documents *[Guide-SES]*, and

(iv) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report.

Because the TOE supports cipher schemes as additional specific security functionality (O.Add-Functions), these security objectives for environment (OE.Plat-Appl) are clarified as follow.
If required the Security IC Embedded Software shall use these cryptographic services of the TOE and their interface as specified. When key-dependent functions implemented in the Security IC Embedded Software are just being executed, the Security IC Embedded Software must provide protection against disclosure of confidential data (User Data) stored and/or processed in the TOE by using the methods described under "Inherent Information Leakage (T.Leak-Inherent)" and "Forced Information Leakage (T.Leak-Forced)".

The Security IC Embedded Software shall provide "Treatment of User Data

(OE.Resp-Appl)" as specified below.

OE.Resp-Appl Treatment of User Data

Security relevant User Data (especially cryptographic keys) are treated by the 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 to unauthorized users or processes when communicating with a terminal.

Because the TOE supports cipher schemes as additional specific security functionality (O.Add-Functions), these security objectives for environment (OE.Resp-Appl) are clarified as follow.
By definition cipher or plain text data and cryptographic keys are User Data. The Security IC Embedded Software shall treat these data appropriately, use only proper secret keys (chosen from a large key space) as input for the cryptographic function of the TOE and use keys and functions appropriately in order to ensure the strength of cryptographic operation.

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

4.2.1.2 Augmented security objectives

The Security IC Embedded Software shall provide "Implementation of command interpreter (OE.Interpreter)" as specified in below.

OE.Interpreter Implementation of command interpreter

It is assumed that the command interpreter used for the tests in Phase 4 and 5 shall be implemented.
To prevent that an attacker abuses the test commands, it is required to authenticate sufficiently before executing these test commands. Besides, the test commands shall be deactivated after completing all of the tests.

4.3 Security Objectives for the operational Environment

4.3.1 TOE Delivery up to the end of Phase 6

Appropriate "Protection during Packaging, Finishing and Personalization

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

OE.Process-Sec-IC Protection during composite product manufacturing

Security procedures shall be used after TOE Delivery up to delivery to the 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 unauthorized use).

This means that Phases after TOE Delivery up to the end of Phase 6 (refer to Section 1.4.2) must be protected appropriately. For a preliminary list of assets to be protected refer to section 3.1.

4.4 Security Objectives Rationale

Table 4 gives an overview, how the assumptions, threats, and organizational security policies are addressed by the objectives.
The rationale justified in [PP] is not changed. Hereinafter, only the additional aspects and aspects added to the TOE (identified by the use of **bold type**) are justified in detail.

Table 4: Security Objectives versus Assumptions, Threats or Policies

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

The following rationale is added for the augmented parts.

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

> Since O.Add-Function 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) in general. User Data are also processed by the specific security functionality required by P.Add-Functions.

Compared to [PP] a clarification has been made for the security objective "Usage of Hardware Platform (OE.Plat-Appl)": If required the Security IC Embedded Software shall use these cryptographic services of the TOE and their interface as specified. In addition, the Security IC Embedded Software must implement functions which perform operations on keys (if any) in such a manner that they do not disclose information about confidential data. This means that this objective covers A.Key-Function in that Key-dependent Functions ensure confidential data or information is protected against leakage attacks. This addition ensures that the assumption A.Plat-Appl is still covered by the objective OE.Plat-Appl although additional functions are being supported according to O.Add-Functions.
For the separation of different applications the Security IC Embedded Software may implement a memory management scheme based upon security mechanisms of the TOE as required by the security policy defined for the specific application context.

Compared to [PP] a clarification has been made for the security objective "Treatment of User Data (OE.Resp-Appl)": By definition cipher or plain text data and cryptographic keys are User Data. So, the 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. It can be concluded from the above that this objective covers A.Key-Function since it ensures that any keys in use are protected from any compromises by adoption of the cryptographic functions. Quality and confidentiality must be maintained for keys that are imported and/or derived from other keys. This implies that appropriate key management has to be realized 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 treatment of User Data is still required when a multi-application operating system is implemented as part of the Security IC Embedded Software on the TOE. In this case the multi-application operating system should not disclose security relevant user data

of one application to another application when it is processed or stored on the TOE.

Rationale added to the TOE is presented below.

The justification related to the security objective "Implementation of command interpreter **(OE.Interpreter)**" is as follows:

> Since OE.Interpreter requires the Security IC Embedded Software developer to implement the interpreter assumed in A.Interpreter, the assumption is covered by the objective.

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

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

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

5 Extended Components Definition

There are three extended components defined for the TOE:

- The family FCS_RNG at the class FCS Cryptographic Support
- The family FMT_LIM at the class FMT Security Management
- The family FAU_SAS at the class FAU Security Audit

The extended components FMT_LIM.1, FMT_LIM.2 and FAU_SAS.1 are used as defined and described in [PP, 5].
Even though the extended family FCS_RNG with its component FCS_RNG.1 is also defined in [PP], it was chosen to take the definition from [KS2011] instead, because in the German certification scheme the [AIS31] is mandatory. Therefore the functional requirement FCS_RNG.1 is seen as a refinement of the one defined in [PP]. The definition of the family and component is given in the next section 5.1.

5.1 Definition of the Family FCS-RNG

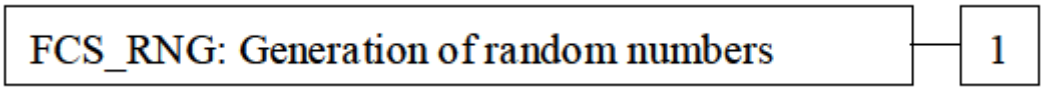
This section describes the functional requirements for the generation of random numbers, which may be used as secrets for cryptographic purposes or authentication. The IT security functional requirements for a TOE are defined in an additional family (FCS_RNG) of the Class FCS (Cryptographic support).

5.1.1 FCS_RNG Generation of random numbers

Family behaviour

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

Component levelling:



FCS_RNG.1	Generation of random numbers requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.
Management:	FCS_RNG.1 There are no management activities foreseen.
Audit:	FCS_RNG.1

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

6 IT Security Requirements

The Security Functional Requirements are shown in Table 5. The additional Security Functional Requirements are shown in **bold type.** These security functional components are listed and explained below.

Table 5: Security Functional Requirements

Security functional requirement		Refined in [PP]
FRU_FLT.2	Limited fault tolerance	Yes
FPT_FLS.1	Failure with preservation of secure state	Yes
FDP_ITT.1	Basic internal transfer protection	Yes
FPT_ITT.1	Basic internal TSF data transfer protection	Yes
FDP_IFC.1	Subset information flow control	No
FPT_PHP.3	Resistance to physical attack	Yes
FMT_LIM.1	Limited capabilities	No
FMT_LIM.2	Limited availability	No
FAU_SAS.1	Audit storage	No
FCS_RNG.1	**Random number generation (Class PTG.2)**	**No**
FCS_COP.1/TDES	**Cryptographic operation**	**-**
FCS_COP.1/AES	**Cryptographic operation**	**-**
FCS_COP.1/DES	**Cryptographic operation**	**-**
FDP_ACC.1	**Subset access control**	**-**
FDP_ACF.1	**Security attribute based access control**	**-**
FMT_MSA.3	**Static attribute initialization**	**-**
FMT_MSA.1	**Management of security attributes**	**-**
FMT_SMF.1	**Specification of management functions**	**-**

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. The refinements are described below the associated SFR.

6.1.1 Security Functional Requirements for the TOE from [PP]

The following Functional Requirements for the TOE are taken from [PP, 6.1].

(1) Malfunctions

There are different ranges of operating conditions such as supply voltage, external frequency and temperature. The TOE can be operated within the limits visualized as the inner dashed rounded rectangle in Figure 3 and must operate correctly there. The limits have been reduced to ensure correct operation. This is visualized by the outer dotted rounded rectangle in the figure.

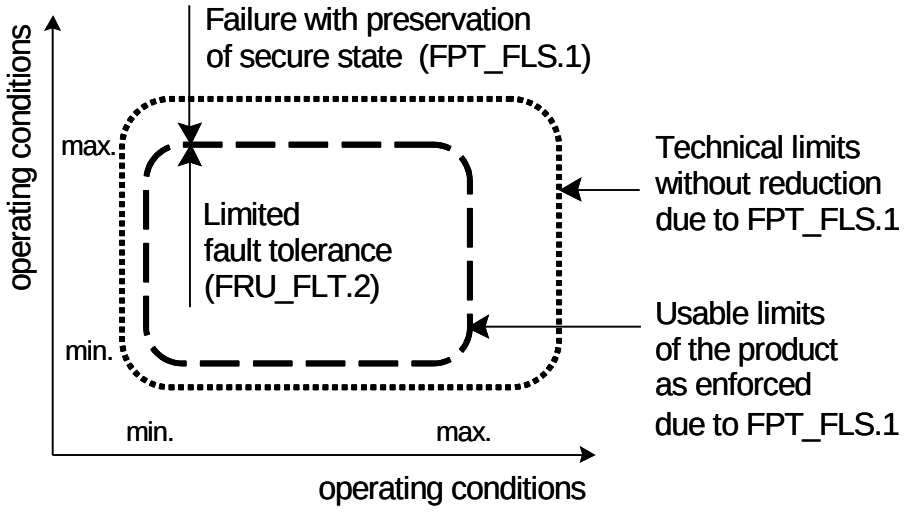


Figure 3: Paradigm regarding Operating Conditions

Figure 3 must not be understood as being two-dimensional and defining static limits only. Reality is multi-dimensional and includes a variety of timing aspects. Note that the limit of the operating conditions visualized by the inner dashed rounded rectangle in Figure 3 is not necessarily exactly reflected by the limits identified in the TOE's data sheet. Instead this limit marks the boundary between the "tolerance reaction" of the TOE and the "active reaction" of sensors (and perhaps other circuitry).

The security functional component has been selected in order to address the robustness within some limit (as shown by the inner dashed rectangle in Figure 3) before active reaction takes place to reach a failure with preservation of secure state.
Note that the TOE does not (in most cases) actually detect faults or failures and then correct them in order to guarantee further operation of all the TOE's capabilities. This is the way software would implement Limited fault tolerance (FRU_FLT.2). Instead the TOE will achieve exactly the same by (i) stable functional design within the limits of operational conditions (e.g. temperature) and (ii) eliminating the cause for possible faults and by being resistant against influences (e.g. robustness against glitches of the power supply by means of filtering). In the case of the TOE the "reaction to a failure" is replaced by the "reaction to operating conditions" which could cause a malfunction without the reaction of the TOE's countermeasure addressed by the security functional component FPT_FLS.1.

If the TOE is exposed to other operating conditions this may not be tolerated. Then the TOE must detect that and preserve a secure state (use of detectors and cause a reset for instance). The security functional component (FPT_FLS.1) has been selected to ensure that. The way the secure state is reached depends on the implementation. Note that the TOE can monitor both external operating conditions and other internal conditions and then react appropriately. Exposure to specific "out of range" external operating conditions (environmental stress) may actually cause failure conditions internally which can not be tolerated by FRU_FLT.2. Referring to external operating conditions the TOE is expected to respond if conditions are detected which may cause a failure. Examples for implementations of the security functional requirement FPT_FLS.1 are a voltage detector (external condition) and a circuitry which detects accesses to address areas which are not used (internal condition).

Those parts of the TOE which support the security functional requirements "Failure with preservation of secure state (FPT_FLS.1)" and "Limited Fault tolerance (FRU_FLT.2)" shall be protected from misconfiguration of and by-passing by means of the Security IC Embedded Software. These aspects are addressed by the security assurance requirements Architectural design (ADV_ARC.1).

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
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)[2].

Dependencies: FPT_FLS.1 Failure with preservation of secure state

Refinement: The term "failure" above means "circumstances". The TOE prevents failures for the "circumstances" defined above.

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

FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: *exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur*[3].

Dependencies: No dependencies.

Refinement: The term "failure" above also covers "circumstances". The TOE prevents failures for the "circumstances" defined above.

(2) Abuse of Functionality

During testing at the end of Phase 3 before TOE Delivery, the TOE shall be able to store some data (for instance about the production history or identification data of the individual die or other data to be used after delivery). Therefore, the security functional component **Audit storage (FAU_SAS.1)** has been added. The security functional component FAU_SAS.1 has been newly created (refer to [PP, 5.3]) and is used instead of FAU_GEN.1 which is too comprehensive to be applicable in this context.

The requirement FAU_SAS.1 shall be regarded as covering the injection of Initialization Data and/or Pre-personalization Data and of supplements of the Security IC Embedded Software as described in [PP, 7.1.1]. After TOE Delivery the identification data (injected as part of the Initialization Data) and the Pre-personalization Data are available to the Security IC Embedded Software. These data are protected by the TOE as all other User Data. It's up to the Security IC Embedded Software to use these data stored and provided by the TOE.

[2] [assignment: list of type of failures]

[3] [assignment: list of types of failures in the TSF]

Each instantiation of the TOE has to undergo exhaustive testing at clearly defined stages of the production process where the correct functioning and properties are ascertained and also if necessary information might be stored in the EEPROM/Flash. This task is done by a specialized group of people of the TOE manufacturer called "test-personnel". The test-personnel is the first user of the TOE and their identity may be assumed as default user for FAU_SAS.1. If the Initialization Data, Pre-personalization Data and supplements of the Security IC Embedded Software can be written only once the test-personnel will be the only user able to store these data.

The TOE shall prevent functions (provided by hardware features) from being abused after TOE Delivery in order to compromise the TOE's security. (All such functions are called "Test Features" below.) This includes but is not limited to: disclose or manipulate User Data and bypass, deactivate, change or explore security features or functions of the TOE. Details depend on the capabilities of the Test Features provided by the IC Dedicated Test Software and/or the hardware.

This can be achieved (i) by limiting the capabilities of these Test Features after Phase 3, (ii) by limiting the availability of these Test Features after Phase 3 or (iii) by a combination of both. The security functional components **Limited capabilities (FMT_LIM.1)** and **Limited availability (FMT_LIM.2)** have been newly created (refer to [PP, 5.2]) to address this.

Examples of the technical mechanism used in the TOE are user authentication ("passwords"), non-availability (for instance through removal or disabling by "fusing") or a combination of both. A detailed technical specification would unnecessarily disclose details and is beyond the scope of the Security Target.

The TOE is tested after production in Phase 3 (refer to [PP, 7.1.1]) using means provided by the IC Dedicated Software and/or specific hardware. The IC Dedicated Software is considered as being a test tool delivered as part of the TOE and used before TOE Delivery only. It does not provide functions in later phases of the Security IC's life-cycle. Therefore, no security functional requirement is mandatory according to this Security Target regarding these testing capabilities except FPT_LIM.1 and FPT_LIM.2.

All necessary information about the capabilities of the Test Features (including the IC Dedicated Software) must be provided by TOE Design (ADV_TDS). The TOE Design (ADV_TDS) shall describe the mechanisms and the Security Architecture (ADV_ARC) shall describe the security architecture design and implementation to limit the availability of the Test Features. The Vulnerability Assessment (AVA) shall analyse the effectiveness of the security mechanisms to enforce FMT_LIM.1 and FMT_LIM.2. For further information on how to handle the Test Features refer to Section 6.2.1.

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

FMT_LIM.1 **Limited capabilities**

Hierarchical to: No other components.

FMT_LIM.1.1 The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: *Deploying Test Features after TOE Delivery does not allow User Data 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*[4].

Dependencies: FMT_LIM.2 Limited availability.

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

FMT_LIM.2 **Limited availability**

Hierarchical to: No other components.

FMT_LIM.2.1 The TSF shall be designed and implemented in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: *Deploying Test Features after TOE Delivery does not allow User Data 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*[5].

Dependencies: FMT_LIM.1 Limited capabilities.

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

FAU_SAS.1 **Audit storage**

Hierarchical to: No other components.

Dependencies: No dependencies.

[4] [assignment: Limited capability and availability policy]

[5] [assignment: Limited capability and availability policy]

FAU_SAS.1.1 The TSF shall provide *the test process before TOE Delivery*[6] with the capability to store *the Initialization Data and/or Pre-personalization Data and/or supplements of the Security IC Embedded Software*[7] in the *FeRAM*[8].

(3) Physical Manipulation and Probing

The TOE can be subject to "tampering" which here pertains to (i) manipulation of the chip hardware and its security features with (ii) prior reverse-engineering to understanding the design and its properties and functions, (iii) determination of critical data through measuring using galvanic contacts, (iv) determination of critical data not using galvanic contacts and (v) calculated manipulation of memory contents.

The TOE is not always powered and therefore not able to detect, react or notify that it has been subject to tampering. Nevertheless, its design characteristics make reverse-engineering and manipulations etc. more difficult. This is regarded as being an "automatic response" to tampering. Therefore, the security functional component **Resistance to physical attack (FPT_PHP.3)** has been selected. The TOE may also provide features to actively respond to a possible tampering attack which is also covered by FPT_PHP.3

The TOE may also leave it up to the Security IC Embedded Software to react when a possible tampering has been detected. Comprehensive guidance (refer to Common Criteria assurance class AGD) will be given for the developer of the Security IC Embedded Software in this case. Taking the assumption "Usage of Hardware Platform (A.Plat-Appl)" into consideration this case shall therefore also be covered by FPT_PHP.3.

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*

6 [assignment: list of subjects]

7 [assignment: list of audit information]

8 [assignment: type of persistent memory]

probing[9] to the *TSF*[10] 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.

(4) Leakage

When the Security IC processes User Data and/or TSF Data, information about these data may be leaked by signals which can be measured externally (e.g. the ISO contacts of the Smartcard). An attacker may also cause malfunctions or perform manipulations of the TOE in order to cause the TOE to leak information. The analysis of those measurement data can lead to the disclosure of User Data and other critical data. Examples are given in [PP, 7.3].

The security functional requirements "Basic internal transfer protection (FDP_ITT.1)" and "Basic internal TSF data transfer protection (FPT_ITT.1)" have been selected to ensure that the TOE must resist leakage attacks (both for User Data and TSF data). The corresponding security policy is defined in the security functional requirement "Subset information flow control (FDP_IFC.1)". These security functional requirements address inherent leakage. With respect to forced leakage they have to be considered in combination with the security functional requirements "Limited fault tolerance (FRU_FLT.2)" and "Failure with preservation of secure state (FPT_FLS.1)" on the one hand and "Resistance to physical attack (FPT_PHP.3)" on the other.

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

FDP_ITT.1 Basic internal transfer protection

Hierarchical to: No other components.

FDP_ITT.1.1 The TSF shall enforce the *Data Processing Policy*[11] to prevent

[9] [assignment: physical tampering scenarios]

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

[11] [assignment: access control SFP(s) and/or information flow control SFP(s)]

the *disclosure*[12] of user data when it is transmitted between physically-separated parts of the TOE.

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

Refinement: The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as physically-separated parts of the TOE.

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

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

FPT_ITT.1.1 The TSF shall protect TSF data from *disclosure*[13] when it is transmitted between separate parts of the TOE.

Dependencies: No dependencies.

Refinement: The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE.

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

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

FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

FDP_IFC.1.1 The TSF shall enforce the *Data Processing Policy*[14] on *all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software*[15].

12 [selection: disclosure, modification, loss of use]

13 [selection: disclosure, modification]

14 [assignment: information flow control SFP]

15 [assignment: list of subjects, information, and operations that cause controlled information to flow to and from

Dependencies: FDP_IFF.1 Simple security attributes

The following Security Function Policy (SFP) **Data Processing Policy** is defined for the requirement "Subset information flow control (FDP_IFC.1)":
User Data and TSF data shall not be accessible from the TOE except when the Security IC Embedded Software decides to communicate the User Data via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the Security IC Embedded Software.

controlled subjects covered by the SFP]

6.1.2 Augmented Security Functional Requirements for the TOE

The strength of the cryptographic algorithms was not rated in the course of the product certification (see BSIG Section 9, Para. 4, Clause 2). But Cryptographic Functionalities with a security level of lower than 100 bits can no longer be regarded as secure without considering the application context. Therefore for these functionalities it shall be checked whether the related crypto operations are appropriate for the intended system. Some further hints and guidelines can be derived from the 'Technische Richtlinie BSI TR-02102', www.bsi.bund.de.

Any Cryptographic Functionality that is marked in column 'Security Level above 100 Bits' of the following table with 'no' achieves a security level of lower than 100 Bits (in general context).

Table 6: TOE cryptographic functionality

Purpose	Cryptographic Mechanism	Standard of Implementation	Key Size in Bits	Security Level above 100 Bits
Cryptographic Primitive	DES	[FIPS46-3] (DES) [SP 800-38A] (CBC, ECB)	|k| = 56	No
	TDES	[SP-800-67] (TDEA) [SP 800-38A] (ECB)	|k| = 112	No
	AES	[FIPS197] (AES) [SP 800-38A] (OFB, ECB, CBC) [SP 800-38B] (CMAC)	|k| = 128	Yes
	Physical True RNG PTG.2	[AIS31]	N/A	N/A

N/A: not applicable

(1) Cryptographic Support

FCS_COP.1/TDES, FCS_COP.1/DES and FCS_COP.1/AES Cryptographic operation require a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard. The dependencies are discusses in Section 6.3.

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

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

(a) [Triple-DES operation]

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

FCS_COP.1/TDES Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1/TDES The TSF shall perform *encryption and decryption*[16] in accordance with a specified cryptographic algorithm *Triple Data Encryption Standard (TDES) in Electronic Codebook (ECB) Mode*[17] and *cryptographic key sizes of 112 bits*[18] that meet the following *standards*[19]:

- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication 800-67, Version 1.1, Revised 19 May 2008*
- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, NIST Special Publication 800-38A, 2001 Edition*

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.

(b) [AES operation]

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

FCS_COP.1/AES Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1/AES The TSF shall perform *encryption and decryption* [20] in

[16] [assignment : list of cryptographic operations]

[17] [assignment : cryptographic algorithm]

[18] [assignment : cryptographic key sizes]

[19] [assignment : list of standards]

[20] [assignment : list of cryptographic operations]

accordance with a specified cryptographic algorithm *Advanced Encryption Standard (AES) in Electronic Code Book (ECB) Mode or Cipher Block Chaining (CBC) Mode or Output Feedback (OFB) Mode or Cipher-based Message Authentication Code (CMAC) Mode* [21] and *cryptographic key sizes of 128 bits*[22] that meet the following *standards*[23]:

- *U.S. Department of Commerce / National Institute of Standards and Technology, Advanced Encryption Standard (AES), FIPS PUB 197, 2001 November 26*
- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, NIST Special Publication 800-38A, 2001 Edition*
- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, The CMAC Mode for Authentication, NIST Special Publication 800-38B, 2005 Edition*

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.

(c) [DES operation]

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

FCS_COP.1/DES Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1/DES The TSF shall perform *encryption and decryption* [24] in accordance with a specified cryptographic algorithm *Data Encryption Standard (DES) in Electronic Codebook (ECB) Mode or Cipher Block Chaining (CBC) mode* [25] and *cryptographic key sizes of 56 bits*[26] that meet the following *standards*[27]:

[21] [assignment : cryptographic algorithm]

[22] [assignment : cryptographic key sizes]

[23] [assignment : list of standards]

[24] [assignment : list of cryptographic operations]

[25] [assignment : cryptographic algorithm]

[26] [assignment : cryptographic key sizes]

- *U.S. Department of Commerce / National Institute of Standards and Technology, DATA ENCRYPTION STANDARD (DES), Federal Information Processing Standards Publication 46-3, Reaffirmed 1999 October 25, withdrawn 2005 May 19*
- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, NIST Special Publication 800-38A, 2001 Edition*

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.

Note: DES does not achieve 100 bits of security in general encryption/ decryption scenarios. For this TOE only the side-channel resistance of DES (i.e. information leakage resistance and fault injection resistance) will be evaluated, not its cryptographic strength. If DES should be used in the security functionality of the embedded software, the corresponding strength/suitability has to be rated in the composite evaluation, in context of the attack potential claimed for the composite product.

(2) Memory Access Control

The security service being provided is described in the Security Function Policy (SFP) Memory Access Control Policy. The security functional requirement "Subset access control (FDP_ACC.1)" requires that this policy is in place and defines the scope were it applies. The security functional requirement "Security attribute based access control (FDP_ACF.1)" addresses security attribute usage and characteristics of policies. It describes the rules for the function that implements the Security Function Policy (SFP) as identified in FDP_ACC.1. The decision whether an access is permitted or not is taken based upon attributes allocated to the software. Examples for such attributes are "the memory area where the software is executed from, the memory area where the access is performed to, special information or properties tied to the software, and/or the operation to be performed" (refer to below). The corresponding permission control information is evaluated so that access is granted/effective or denied/inoperable.

The security functional requirement "Static attribute initialization (FMT_MSA.3)" ensures that the default values of security attributes are

[27] [assignment : list of standards]

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

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

Memory Access Control Policy

The TOE shall control read, write, delete, and execute accesses of software residing in memory areas on data including code stored in memory areas.

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

FDP_ACC.1	**Subset access control**
Hierarchical to:	No other components.
FDP_ACC.1.1	The TSF shall enforce *the Memory Access Control Policy*[28] on *all subjects (software in memories), all objects (data including code stored in memories) and all the operations defined in the Memory Access Control Policy*[29].
Dependencies:	FDP_ACF.1 Security attribute based access control.

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

FDP_ACF.1	**Security attribute based access control**
Hierarchical to:	No other components.
FDP_ACF.1.1	The TSF shall enforce the *Memory Access Control Policy*[30] to objects based on the following[31]:

[28] [assignment: access control SFP]

[29] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

[30] [assignment: access control SFP]

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

Subjects: software in memories,
Objects: data in memories,
Attributes: *memory address of program counter*
memory address of accessed data
permission control information

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *evaluate the corresponding permission control information before, during or after the access so that accesses to be denied can not be utilised by the subject attempting to perform the operation*[32].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *none*[33].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*[34].

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

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

FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the *Memory Access Control Policy*[35] to provide *restrictive*[36] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow *no subject (provided that the Memory Access Control Policy is enforced and the necessary access is therefore allowed)*[37] to specify alternative initial values to override the default values when an object or information is

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

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

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

[35] [assignment: access control SFP, information flow control SFP]

[36] [selection: restrictive, permissive, other property]

[37] [assignment: the authorised identified roles]

created.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

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

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the *Memory Access Control Policy*[38] to restrict the ability to *modify*[39] the security attributes *permission control information*[40] to *software running in API Mode*[41].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

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

FMT_SMF.1 Specification of management functions

Hierarchical to: No other components

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: *configuration of the permission control information* [42].

Dependencies: No dependencies

(3) Random Numbers

The TOE generates random numbers. To define the IT security functional requirements of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined in section 5.1. This family FCS_RNG Generation of random numbers describes

[38] [assignment: access control SFP, information flow control SFP]

[39] [selection: change_default, query, modify, delete, [assignment: other operations]]

[40] [assignment: list of security attributes]

[41] [assignment: the authorised identified roles]

[42] [assignment: list of management functions to be provided by the TSF]

the functional requirements for random number generation used for cryptographic purposes.

The TOE shall meet the requirement "Random number generation (FCS_RNG.1)" as specified below (Common Criteria Part 2 extended).

FCS_RNG.1 Random number generation (Class PTG.2)

Hierarchical to: No other components.

FCS_RNG.1.1 The TSF shall provide a *physical*[43] 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*[44]*.*

(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 applied upon specified internal events*[45]*. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.*[46]

FCS_RNG.1.2 The TSF shall provide *1 byte* that meet:

(PTG.2.6) *Test procedure A*[47] *does not distinguish the internal random*

[43] [selection: physical, non-physical true, deterministic, hybrid physical, hybrid deterministic]

[44] [selection: prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy]

[45] [selection: externally, at regular intervals, continuously, applied upon specified internal events]

[46] [assignment: list of security capabilities]

[47] [assignment: additional standard test suites]

	numbers from output sequences of an ideal RNG.
(PTG.2.7)	*The average Shannon entropy per internal random bit exceeds 0.997.*

Dependencies: No dependencies.

Note: This functional requirement taken from [KS2011] is seen as a refinement of the one stated in [PP].

6.2 Security Assurance Requirements for the TOE

The assurance level for this Security Target is **EAL6** augmented with the following components:

- ASE_TSS.2

The assurance requirements are given in the following Table 7.
Augmentations compared to [PP] are marked in bold face.

Table 7: Assurance Requirements

Assurance class	ID	Family name
Development (Class ADV)	ADV_ARC.1	Architectural Design
	ADV_FSP.5	**Functional Specification**
	ADV_IMP.2	**Implementation Representation**
	ADV_INT.3	**TSF Internals**
	ADV_SPM.1	**Security policy modelling**
	ADV_TDS.5	**TOE Design**
Guidance documents (Class AGD)	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life-cycle support (Class ALC)	**ALC_CMC.5**	**CM Capabilities**
	ALC_CMS.5	**CM Scope**
	ALC_DEL.1	Delivery
	ALC_DVS.2	Development Security
	ALC_LCD.1	Life-Cycle Definition
	ALC_TAT.3	**Tools and Techniques**
Security Target evaluation (Class ASE)	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.2	Security Objectives
	ASE_REQ.2	Derived Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.2	**TOE Summary Specification**
Tests (Class ATE)	**ATE_COV.3**	**Coverage**
	ATE_DPT.3	**Depth**
	ATE_FUN.2	**Functional Tests**
	ATE_IND.2	Independent Testing
Vulnerability assessment (Class AVA)	AVA_VAN.5	Vulnerability Analysis

6.2.1 Refinements of the TOE Assurance Requirements

Refinements list of the assurance requirements taken from [PP, 6.2.1] is shown in Table 8. For details of the refinements refer to [PP].

Table 8: Refinements list of Assurance Requirements

Refinements of the assurance requirements	Family name	Abbreviated name
Refinements regarding Delivery procedure	Delivery	ALC_DEL
Refinements regarding Development Security	Development Security	ALC_DVS
Refinement regarding CM scope	CM Scope	ALC_CMS
Refinement regarding CM capabilities	CM Capabilities	ALC_CMC
Refinements regarding Security Architecture	Architectural Design	ADV_ARC
Refinements regarding Functional Specification	Functional Specification	ADV_FSP
Refinements regarding Implementation Representation	Implementation Representation	ADV_IMP
Refinement regarding Test Coverage	Coverage	ATE_COV
Refinement regarding User Guidance	Operational User Guidance	AGD_OPE
Refinement regarding Preparative User Guidance	Preparative User Guidance	AGD_PRE
Refinement regarding Vulnerability Analysis	Vulnerability Analysis	AVA_VAN

Five refinements from the [PP] have to be discussed since the assurance level of the corresponding component is increased in the Security Target

CM Scope (ALC_CMS)

The refinement from the [PP] can be applied even to the chosen assurance component ALC_CMS.5. The assurance component ALC_CMS.4 is extended to ALC_CMS.5 with regard to the scope of the configuration list. The refinement is not touched in terms of this matter.

CM Capabilities (ALC_CMC)

The refinement from the [PP] can be applied even to the chosen assurance component ALC_CMC.5. The assurance component ALC_CMC.4 is extended to ALC_CMC.5 with aspects regarding the following:

(i) justification of the acceptance procedures, and

(ii) advanced CM system (automated means to identify all other configuration items that are affected by the change of a given configuration item, identification of the version of the implementation representation, and so on)

The refinement provides the detailed explanation about production control system and administration procedures, and is not touched in terms of those matters.

Functional Specification (ADV_FSP)

The refinement from the [PP] can be applied even to the chosen assurance component

ADV_FSP.5. The assurance component ADV_FSP.4 is extended to ADV_FSP.5 with aspects regarding (i) the description of TSFI using a semi-formal style, and (ii). error messages that do not result from an invocation of a TSFI and the rationale for them.
The refinement provides the detailed explanation about IC Dedicated Test Software, and description content of functional specification, and is not touched in terms of those matters.

Implementation Representation (ADV_IMP)
The refinement from the [PP] can be applied even to the chosen assurance component ADV_IMP.2. The assurance component ADV_IMP.1 is extended to ADV_IMP.2 with aspects regarding the amount of implementation that is mapped to the TOE design description. The refinement is not touched in terms of this matter.

Coverage (ATE_COV)
The refinement from the [PP] can be applied even to the chosen assurance component ATE_COV.3. The assurance component ATE_COV.2 is extended to ATE_COV.3 with aspects regarding the degree of analysis. The refinement provides the detailed explanation about test operating conditions and evidence of existence and effectiveness of mechanisms against physical attacks, and is not touched in terms of those matters.

6.2.2 Definition of ADV_SPM

The developer shall provide a formal security policy model for the Memory Access Control Policy. The Memory Access Control Policy comprises the following Security Functional Requirements: FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3, FMT_SMF.1 with the associated dependencies. Moreover, parts of the Data Processing Policy (FDP_IFC.1, FDP_ITT.1, FPT_ITT.1), Limited Availability and Limited Capability Policy (FMT_LIM.1, FMT_LIM.2), Malfunctions (FPT_FLS.1, FRU_FLT.2), Audit Storage (FAU_SAS.1) and Resistance to physical attack (FPT_PHP.3) are part of the model.

6.3 Security Requirements Rationale

6.3.1 Rationale for the security functional requirements

Table 9 below gives an overview, how the security functional requirements are combined to meet the security objectives.
The rationale justified in [PP, 6.3] is not changed. Hereinafter, only the additional aspects added, and aspects added to the TOE (identified by the use of **bold type**) are justified in detail.

Table 9: Security Requirements versus Security Objectives

Objective	TOE Security Functional and Assurance Requirements
O.Leak-Inherent	- FDP_ITT.1 - FPT_ITT.1 - FDP_IFC.1
O.Phys-Probing	- FPT_PHP.3
O.Malfunction	- FRU_FLT.2 - FPT_FLS.1
O.Phys-Manipulation	- FPT_PHP.3
O.Leak-Forced	All requirements listed for O.Leak-Inherent - FDP_ITT.1, - FPT_ITT.1, - FDP_IFC.1 plus those listed for O.Malfunction and O.Phys-Manipulation - FRU_FLT.2, - FPT_FLS.1, - FPT_PHP.3
O.Abuse-Func	- FMT_LIM.1 - FMT_LIM.2 plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced - FDP_ITT.1, - FPT_ITT.1, - FDP_IFC.1, - FPT_PHP.3, - FRU_FLT.2, - FPT_FLS.1,
O.Identification	- FAU_SAS.1
O.RND	- FCS_RNG.1 plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced“ - FDP_ITT.1, - FPT_ITT.1, - FDP_IFC.1, - FPT_PHP.3, - FRU_FLT.2, - FPT_FLS.1,
O.Add-Functions	- **FCS_COP.1/TDES** - **FCS_COP.1/AES** - **FCS_COP.1/DES**
OE.Plat-Appl	not applicable
OE.Resp-Appl	not applicable
OE.Interpreter	not applicable
OE.Process-Sec-IC	not applicable
O.Mem-Access	- **FDP_ACC.1 “Subset access control”** - **FDP_ACF.1 “Security attribute based access control”** - **FMT_MSA.3 “Static attribute initialisation”** - **FMT_MSA.1 “Management of security attributes”** - **FMT_SMF.1 “Specification of management functions”**

The following rationale is performed for the augmented parts.

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/TDES, FCS_COP.1/DES and FCS_COP.1/AES)” exactly require those functions to be implemented which are demanded by O.Add-Functions. Therefore, FCS_COP.1/TDES, FCS_COP.1/DES and FCS_COP.1/AES are suitable to meet the security objective.

Nevertheless, the developer of the Security IC Embedded Software must ensure that the additional functions are used as specified and that the User Data processed by these functions are protected as defined for the application context. These issues are addressed by the specific security functional requirements:

- [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.

All these requirements have to be fulfilled to support OE.Resp-Appl for FCS_COP.1/TDES, FCS_COP.1/DES and for FCS_COP.1/AES.

The usage of cryptographic algorithms requires the use of appropriate keys. Otherwise these cryptographic functions do not provide security. The keys have to be unique with a very high probability, and must have a certain cryptographic strength etc. In case of a key import into the TOE (which is usually after TOE delivery) it has to be ensured that quality and confidentiality are maintained. Keys for TDES, DES and AES are provided by the environment
In this ST the objectives for the environment OE.Plat-Appl and OE.Resp-Appl have been clarified. The Security IC Embedded Software defines the use of the cryptographic functions FCS_COP.1 provided by the TOE. The requirements for the environment FDP_ITC.1, FDP_ITC.2, FCS_CKM.1 and FCS_CKM.4 support an appropriate key management. These security requirements are suitable to meet OE.Resp-Appl.

The justification of the security objective and the additional requirements (both for the TOE and its environment) show that they do not contradict to the rationale already given in the Protection Profile for the assumptions, policy and threats defined there.

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

The security functional requirement “Subset access control (FDP_ACC.1)” with the related Security Function Policy (SFP) “Memory Access Control Policy” exactly requires to implement an area based memory access control as demanded by O.Mem-Access. Therefore, FDP_ACC.1, FDP_ACF.1, FMT_MSA.3, FMT_MSA.1 and FMT_SMF.1 with

its SFP are suitable to meet the security objective.

The security functional requirement "Static attribute initialization (FMT_MSA.3)" requires that the TOE provides default values for security attributes. These default values can not be overwritten by any subject (software) provided that the necessary access is not allowed what is further detailed in the security functional requirement "Management of security attributes (FMT_MSA.1)": The ability to update the security attributes is restricted to privileged subject(s). These management functions ensure that the required access control can be realized using the functions provided by the TOE.

Note that there is a detailed explanation for each security functional requirement in Section 6.1.

6.3.2 Rationale for definition of the extended component FCS_RNG.1

The CC part 2 defines the component FIA_SOS.2, which is similar to FCS_RNG.1.
The CC part 2, annex G.3 [CCV31_2], states: "This family defines requirements for mechanisms that enforce defined quality metrics on provided secrets, and generate secrets to satisfy the defined metric". Even the operation in the element FIA_SOS.2.2 allows listing the TSF functions using the generated secrets. Because all applications discussed in annex G.3 are related to authentication, the component FIA_SOS.2 is also intended for authentication purposes while the term "secret" is not limited to authentication data (cf. CC part 2, paragraphs 39-42).
Paragraph 685 in the CC part 2 [CCV31_2] recommends use of the component FCS_CKM.1 to address random number generation. However, this may hide the nature of the secrets used for key generation and does not allow describing random number generation for other cryptographic methods (e.g., challenges, padding), authentication (e.g., password seeds), or other purposes (e.g., blinding as a countermeasure against side channel attacks).
The component FCS_RNG addresses general RNG, the use of which includes but is not limited to cryptographic mechanisms. FCS_RNG allows to specify requirements for the generation of random numbers including necessary information for the intended use. These details describe the quality of the generated data where other security services rely on. Thus by using FCS_RNG a ST author is able to express a coherent set of SFRs that include or use the generation of random numbers as a security service.

6.3.3 Dependencies of security functional requirements

Table 10 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.

This rationale is adopted from [PP, 6.3.2], with additional aspects (identified by the use of **bold type**).

Table 10: Dependencies of the Security Functional Requirements

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 in [PP, 6.3.2]
FPT_ITT.1	None	No dependency
FCS_RNG.1	None	No dependency
FCS_COP.1/TDES	**FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4**	**Yes (by the environment)**
FCS_COP.1/AES	**FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4**	**Yes (by the environment)**
FCS_COP.1/DES	**FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FCS_CKM.4**	**Yes (by the environment)**
FDP_ACC.1	**FDP_ACF.1**	**Yes**
FDP_ACF.1	**FDP_ACC.1 FMT_MSA.3**	**Yes Yes**
FMT_MSA.3	**FMT_MSA.1 FMT_SMR.1**	**Yes See discussion below**
FMT_MSA.1	**FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1**	**Yes See discussion below Yes**

The dependencies defined for FCS_COP.1/TDES, FCS_COP.1/DES and FCS_COP.1/AES are addressed in the environment through the presence of OE.Plat-Appl and OE.Resp-Appl.
These dependencies all address the appropriate management of cryptographic keys used by the specified cryptographic function and are not part of the [PP]. The requirements concerning key management shall be fulfilled by the environment since the Security IC Embedded Software is designed for a specific application context and uses the cryptographic functions provided by the TOE.

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

6.3.4 Rationale for the Assurance Requirements

The assurance level EAL6 and the augmentation with the requirements ASE_TSS.2 were chosen in order to meet assurance expectations explained in the following paragraphs.

An assurance level of EAL6 with the augmentations ASE_TSS.2 is 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 defense against such attacks, the evaluators should have access to the low level design and source code.
Additionally the mandatory technical document "Application of Attack Potential to Smartcards" [JHAS] shall be taken as a basis for the vulnerability analysis of the TOE.

ASE_TSS.2 TOE summary specification with architectural design summary
For sophisticated attacks, it is necessary for evaluators and potential consumers to gain a general understanding that there isn't a specific concern regarding the TOE security architecture. Therefore ASE_TSS.2 is selected.
By ASE_TSS.2, evaluators and potential consumers can gain a general understanding of how the TOE protects itself against interference, logical tampering and bypass.
ASE_TSS.2 has dependencies to ASE_INT.1 "ST introduction", ASE_REQ.1 "Stated security requirements", and ADV_ARC.1 "Security architecture description".
All these dependencies are satisfied by EAL6.

6.3.5 Security Requirements are Internally Consistent

In addition to the discussion in [PP, 6.3.4], the security functional requirement FCS_COP.1/TDES, FCS_COP.1/DES and FCS_COP.1/AES are newly added to this Security Target. The additional rationale to deal with FCS_COP.1/TDES, FCS_COP.1/DES and FCS_COP.1/AES is as follows.

The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced also protect the cryptographic algorithms implemented according to the security functional requirement FCS_COP.1/TDES, FCS_COP.1/DES and FCS_COP.1/AES. Therefore, these security functional requirements support the secure implementation and operation of FCS_COP.1/TDES, FCS_COP.1/DES and FCS_COP.1/AES.

The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced also protect the area based memory access control function implemented according to the security functional requirement described in the security functional requirement FDP_ACC.1 with reference to the Memory Access Control Policy and

details given in FDP_ACF.1. Therefore, those security functional requirements support the secure implementation and operation of FDP_ACF.1 with its dependent security functional requirements.

7 TOE Summary Specification

7.1 TOE security functionality

7.1.1 TOE Security Features

(1) SF.RNG: Random Number Generator

The TOE generates true random numbers, and meets the class PTG.2 of [AIS31]. The TOE implements this security function by means of a physical hardware random number generator working stable within the limits guaranteed by the security function SF.FAS.
The generated random numbers are used internally and can be used by the Security IC Embedded Software for e.g. the generation of cryptographic keys.

(2) SF.FAS: Filters and Sensors

The TOE prevents any malfunction and ensures its correct operation.

The TOE incorporates effective filters on the essential signal lines so as to eliminate the cause for possible faults such as glitches. Moreover, the TOE has sensors to detect a variety of operating conditions that could lead to malfunctions, including frequency, voltages and temperatures. These filters and sensors functions are listed in the following Table 11.

Table 11: Filter and Sensor functions

Filter / Sensor	Functions
VDDA Voltage Sensor	Analog power-supply Voltage anomaly detection (high voltage / low voltage)
VDD Voltage Sensor	Digital power-supply Voltage anomaly detection (high voltage / low voltage)
VPP Voltage Sensor	FeRAM pressor power-supply Voltage anomaly detection (high voltage / low voltage)
VDD Voltage Glitch Sensor	Glitch on VDD Voltage detection
Low Frequency Sensor	Low frequency Clock detection
Clock Filter	- High frequency Clock detection - Glitch on Clock removal
Reset Filter	Glitch on Reset Signal removal
Light Sensor	Light detection
Temperature Sensor	Temperature detection (high temperature / low temperature)

If any abnormality is detected on sensors, CPU and all registers are initialized.

In addition, the TOE starts the self-test upon power-up at all times. If any abnormality is detected on the filters or sensors, CPU and all registers are

initialized. It is therefore ensured that these filters and sensors properly operate.

All the instructions that are executed in CPU are being monitored. When an illegal instruction is referenced in the CPU, it indicates a corruption due to an attack. In this case the TOE enters the reset state and CPU and all registers are initialized.

Parameter that is set up to IC Dedicated Software is checked. If it is an unauthorized value, CPU and all registers are initialized.

(3) SF.PHY: Tamper Resistance

The TOE comprises various physical measures that make tamper attacks more difficult and to protect thereby data stored in the ROM, SRAM, FeRAM such as User Data, Security IC Embedded Software and other critical operating information (TSF data in particular) from being modified by FIB etc. or disclosed using the physical probing.

One of the countermeasures is memory scramble.

Furthermore, sensing shield is embedded. If any abnormal physical operation is detected, CPU and all registers are initialized.

The critical data as mentioned above is protected using such secured mechanism.

(4) SF.DPR: Data Protection

The TOE may be susceptible to physical attacks: therefore it has potential risk of internal data leakage. For example, if an attacker collects measurements on the signals being used in processing User data and/or TSF data, and performs complex computation processes on them, he or she may obtain their confidential data in the TOE thereby or possibly directly from FeRAM or ROM.
To avoid such unwanted leakage, particularly to protect against SPA, DPA, DFA and timing attack, the TOE comprises the security measures.

(5) SF.MCT: Mode Control

For chip, there are Test mode and Normal mode. Factory setting is the Normal mode.

After the execution of all tests at Phase 3, test mode entry becomes impossible and the transition from Normal mode to Test mode falls into disuse.

Under the mode control as described above, abuse of test functions is prevented after TOE delivery.

(6) SF.CRPT: Cryptography

The TOE realizes the TDES encryption/decryption, DES encryption/decryption and AES encryption/decryption as specified by the following standards:

- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication 800-67, Version 1.1, Revised 19 May 2008*
- *U.S. Department of Commerce / National Institute of Standards and Technology, DATA ENCRYPTION STANDARD (DES), Federal Information Processing Standards Publication 46-3, Reaffirmed 1999 October 25, withdrawn 2005 May 19*
- *U.S. Department of Commerce / National Institute of Standards and Technology, Advanced Encryption Standard (AES), FIPS PUB 197, 2001 November 26*
- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, NIST Special Publication 800-38A, 2001 Edition*
- *U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, The CMAC Mode for Authentication, NIST Special Publication 800-38B, 2005 Edition*

(7) SF.ACU: Access Control Unit

All addresses are being monitored by this security function.
For this function, there are settable 2 modes; User mode and API mode. Accessible/inaccessible area is controlled in accordance with a selected mode.

When an address is specified to point to an access-inhibited area, it indicates a corruption due to an attack. In this case the TOE enters the reset state and CPU and all registers are initialized

(8) SF.ID: ID Injection

In the last function testing at Phase 3, some data to uniquely identify the TOE are injected into the write lock area of FeRAM. This sort of information can't be rewritten. Therefore the data like ID written down in the TOE isn't changed.

7.2 TOE Summary Specification Rationale

Table 12 below gives an overview, how the security functional requirements are fulfilled by TOE security functions. The text following after the table justifies in detail.
This security target (ST-Lite) can't provide the rationale for the specification of TOE summary.

Table 12: mapping of SFR to TOE Security Function

SFR \ TSF	SF.RNG	SF.FAS	SF.PHY	SF.DPR	SF.MCT	SF.CRPT	SF.ACU	SF.ID
FRU_FLT.2		✓						
FPT_FLS.1		✓						
FDP_ITT.1			✓	✓				
FPT_ITT.1			✓	✓				
FDP_IFC.1			✓	✓				
FPT_PHP.3			✓					
FMT_LIM.1					✓			
FMT_LIM.2					✓			
FAU_SAS.1								✓
FCS_RNG.1	✓							
FCS_COP.1/TDES						✓		
FCS_COP.1/AES						✓		
FCS_COP.1/DES						✓		
FDP_ACC.1							✓	
FDP_ACF.1							✓	
FMT_MSA.3							✓	
FMT_MSA.1							✓	
FMT_SMF.1							✓	

7.3 TOE architectural design summary

This section describes how the TOE protects itself against interference, logical tampering, and bypass.

7.3.1 Physical Attacks

Microelectronic tools enable to either access or modify an IC by removing or adding material (etching, FIB, etc).
The security feature SF.PHY described in section 7.1.1 (3) counters such attacks.

7.3.2 Overcoming sensors and filters

This attack covers ways of deactivating or avoiding the different types of sensor that an IC may use to monitor the environmental conditions and to protect itself from conditions that would threaten correct operation of the TOE.
The security feature SF.FAS described in section 7.1.1 (2) counters such attacks.

7.3.3 Perturbation Attacks

Perturbation attacks change the normal behaviour of an IC in order to create an exploitable error in the operation of a TOE.
The security features SF.DPR described in section 7.1.1 (4) counter the attack by modifying a value read from memory during the read operation and the program flow.
The security features SF.RNG described in section 7.1.1 (1) counter the attack by changing the characteristics of random numbers generated.

7.3.4 DFA Attacks

With DFA an attacker tries to obtain a secret by comparing a calculation without an error and calculations that do have an error.
The security features SF.FAS and SF.DPR described in sections 7.1.1 (2) and 7.1.1 (4) counter such attacks.

7.3.5 Side-channel Attacks

SPA and DPA aim at exploiting the information leaked through characteristic variations in the power consumption of electronic components – yet without damaging the TOE in any way what-so-ever.
The security features SF.DPR described in section 7.1.1 (4) counter such attacks.

The attack path aims to correlate more than once per TOE computation using hypotheses on intermediate states that depend on secret key parts.
The countermeasure is the same as described above.

The attack path aims to recover secret data such as keys, plaintext, and so on by observing electromagnetic radiation emitted from TOE.
The countermeasure is the same as described above.

7.3.6 Exploitation of Test features

The attack path aims to enter the IC test mode to provide a basis for further attacks.
The security features SF.MCT described in section 7.1.1 (5) counter such attacks.

7.3.7 Attacks on RNG

The attack path aims to predict the output of the RNG (e.g. of reducing the output entropy).
The security feature SF.RNG described in section 7.1.1 (1) counters such attacks

7.3.8 Software Attacks

First an information gathering attack step, which may be relevant to a number of different types of attack, is considered. And then, the following specific attack techniques that may exploit software vulnerabilities are introduced: Editing commands, Direct protocol attacks, Man-in-the-middle attacks, and Replay attacks.
For the countermeasures for these attacks, see the section 7.3.10 to 7.3.13.

7.3.9 Information gathering

An attacker may try to observe message sequence to obtaining information on an unknown protocol.
It's really difficult for an attacker to obtain valid information because sensitive commands and responses are encrypted.
The security features SF.RNG described in section 7.1.1 (1) reinforce the authentication.

In addition, the security features SF.PHY described in section 7.1.1 (3) and SF.DPR described in section 7.1.1 (4) protect against tamper attack to the authentication keys.

Moreover the security feature SF.DPR described in section 7.1.1 (4) protects against SPA/DPA or DFA to the authentication keys.

7.3.10 Editing commands

An attacker may try to modify commands during the communication sequence to see if the card gives an unexpected reply.
It's really difficult for an attacker to modify commands because sensitive commands are protected.

7.3.11 Direct protocol attacks

An attacker may try to receive uninitialized communication buffer.

It's really difficult for an attacker to receive secret data because the security feature SF.DPR described in section 7.1.1 (4) counters such attacks.

7.3.12 Man-in-the-middle attacks

An attacker may try to hide in the communication path between two entities that are executing a valid communication.
The countermeasure is same as section 7.3.10.

7.3.13 Replay attacks

An attacker may try to use a protocol analyzer to monitor and copy packets as they flow between smartcard and Reader/Writer.
The countermeasure is same as section 7.3.10.

7.3.14 Bypass authentication or access control

An attacker may try to get unauthorized access to data residing on the smartcard respectively at performing operations which do not match the current life cycle state of processed data objects or of the Operating System.
It's really difficult for an attacker to try such attacks because command editing, man-in-the-middle attack or replay attack is extremely difficult as described above.

8 Annex

8.1 Glossary of Vocabulary

Terms	Definitions
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 Manufacturer	The Composite Product Manufacturer has the following roles (i) the Security IC Embedded Software Developer (Phase 1), (ii) the Composite Product Integrator (Phase 5) and (iii) the Personaliser (Phase 6). If the TOE is delivered after Phase 3 in form of wafers or sawn wafers (dice) he has the role of the IC Packaging Manufacturer (Phase 4) in addition. 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 (refer to section 1.4.2 and [PP, 7.1.1]).
End-consumer	User of the Composite Product in Phase 7.
IC Dedicated Software	IC proprietary software embedded in a Security IC (also known as IC firmware) and developed by the IC Developer. Such software is required for testing purpose (IC Dedicated Test Software) but may provide additional services to facilitate usage of the hardware and/or to provide additional services (IC Dedicated Support Software).
IC Dedicated Test Software	That part of the IC Dedicated Software (refer to above) which is used to test the TOE before TOE Delivery but which does not provide any functionality thereafter.
IC Dedicated Support Software	That part of the IC Dedicated Software (refer to above) which provides functions after TOE Delivery. The usage of parts of the IC Dedicated Software might be restricted to certain phases.
Initialization Data	Initialization 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-personalization Data	Any data supplied by the Card Manufacturer that is injected into the non-volatile memory by the Integrated Circuits manufacturer (Phase 3). These data are for instance used for traceability and/or to secure shipment between phases.
Security IC	(as used in this Protection Profile) Composition of the TOE, the Security IC Embedded Software, User Data and the package (the Security IC carrier).

Terms	Definitions
Security IC Embedded Software	Software embedded in a smart card 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.
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 Security IC Embedded Software in the application context. User data comprise all data in the final smart card IC except the TSF data.
PICC bus	Bus between CPU and each device (Test Circuit, FeRAM, Cryptographic Hardware, Communication Hardware, Access Control, and Security Circuit).

8.2 List of Abbreviations

Abbreviations	Meanings
AES	Advanced Encryption Standard
ASK	Amplitude Shift Keying
CBC	Cipher Block Chaining
CC	Common Criteria Version 3.1
DES	Data Encryption Standard
DFA	Differential Fault Analysis
DPA	Differential Power Analysis
EAL	Evaluation Assurance Level
EB	Electron Beam
ECB	Electronic Code Book
FIB	Focused Ion Beam
IC	Integrated Circuit
IT	Information Technology
NMI	Non-Maskable Interrupt
PP	Protection Profile
RF	Radio Frequency
RNG	Random Number Generator
SPA	Simple Power Analysis
ST	Security Target
TDES	Triple Data Encryption Standard
TOE	Target of Evaluation
Triple-DES	Triple Data Encryption Standard
TSF	TOE Security functionality

8.3 Related Documents

Abbreviated name	References
[AIS31]	Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, AIS31, Version 3.0, 15.05.2013
[CC]	Common Criteria for Information Technology Security Evaluation; Version 3.1
[CC-1]	Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1 Revision 4
[CC-2]	Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; Version 3.1 Revision 4
[CC-3]	Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components; Version 3.1 Revision 4
[FIPS46-3]	U.S. Department of Commerce / National Institute of Standards and Technology, DATA ENCRYPTION STANDARD (DES), Federal Information Processing Standards Publication 46-3, Reaffirmed 1999 October 25, withdrawn 2005 May 19
[FIPS197]	U.S. Department of Commerce / National Institute of Standards and Technology, Advanced Encryption Standard (AES), FIPS PUB 197, 2001 November 26
[AGD-SES]	MN67S150 Smart Card IC Operational User Guidance, - for Security IC Embedded Software Developer -
[AGD-CM]	MN67S150 Smart Card IC Preparative User Guidance, - for Card Manufacturer -
[JHAS]	Joint Interpretation Library, Application of Attack Potential to Smartcards, Version 2.9, January 2013
[KS2011]	A proposal for: Functionality classes for random number generators, Version 2.0, 18 September 2011
[PP]	Smartcard IC Platform Protection Profile, BSI-PP-0035, Version 1.0, June 2007.
[ISO/IEC 14443]	ISO/IEC 14443: Identification cards -- Contactless integrated circuit(s) cards -- Proximity cards --
[JISX6319-4]	JAPANESE INDUSTRIAL STANDARD, Specification of implementation for integrated circuit(s) cards – Part 4: High speed proximity cards JISX6319-4: 2010
[SP-800-38A]	U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, NIST Special Publication 800-38A, 2001 Edition
[SP-800-38B]	U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for Block Cipher Modes of Operation, The CMAC Mode for Authentication, NIST Special Publication 800-38B, 2005 Edition
[SP-800-67]	U.S. Department of Commerce / National Institute of Standards and Technology, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, NIST Special Publication 800-67, Version 1.1, Revised 19 May 2008
[JIL]	Joint Interpretation Library, ST-Lite, Version1.1, December 2002