NXP Secure Smart Card Controller E201382

Security Target Lite

Rev. 1.3 – 25 September 2017 **Evaluation documentation**

Final **Public**

Document Information

Info	Content
Keywords	CC, Security Target Lite, E201382
Abstract	Security Target Lite of the NXP Secure Smart Card Controller E201382, which is developed and provided by NXP Semiconductors, Business Unit Security & Connectivity according to the Common Criteria for Information Technology Security Evaluation Version 3.1 at EAL5 augmented

Rev	Date	Description
0.1	07-August-2014	Draft Version
0.2	04-September-2015	Updated based on Certification Body requirements.
0.3	28-September-2015	Updated based on ITSEF review.
0.4	02-October-2015	Update of Variable Definitions for Commercial Type Names.
0.5	19-October-2015	Update based on ITSEF review.
1.0	26-January-2016	Update of Date and Release information for the Components of the TOE.
1.1	03-February-2016	Another Update of Date and Release information for the Components of the TOE.
1.2	12-February-2016	Update of TOE Reference.
1.3	25-September-2017	Update of Date and Release information for the Components of the TOE.

1 ST Introduction

This chapter is divided into the following sections: ”ST Reference”, ”TOE Reference”, ”TOE Overview” and ”TOE Description”.

1.1 ST Reference

NXP Secure Smart Card Controller E201382 Security Target, 1.3, NXP Semiconductors, 25 September 2017.

1.2 TOE Reference

NXP Secure Smart Card Controller E201382, IC Hardware Versions VA and VB, IC Dedicated Support Software Version 1.0.

1.3 TOE Overview

1.3.1 Usage and Major Security Functionality of the TOE

The TOE is the IC hardware platform NXP Secure Smart Card Controller E201382 with IC Dedicated Software and documentation describing instruction set and usage of the TOE. The TOE does not include a customer-specific Security IC Embedded Software.

The IC hardware is a microcontroller incorporating a central processing unit (CPU), memories accessible via a Memory Management Unit (MMU), cryptographic coprocessors, other security components and several electrical communication interfaces. The central processing unit supports a 32-/16-bit instruction set optimized for smart card applications. The first and in some cases the second byte of an instruction are used for operation encoding. On-chip memories are ROM, NVM and RAM. The Non-Volatile Memory (NVM) can be used as data or program memory. It consists of high reliable memory cells, which guarantee data integrity. NVM is optimized for applications that require reliable non-volatile data storage for data and program code. Dedicated security functionality protects the contents of all memories.

The IC Dedicated Software comprises IC Dedicated Test Software for test purposes and IC Dedicated Support Software. The IC Dedicated Support Software consists of the Boot Software, which controls the boot process of the hardware platform. Furthermore the TOE provides a Hardware Abstraction Layer (HAL) (the HAL Software) and a Crypto Library (the Crypto Library) simplifying the access to the hardware for the Security IC Embedded Software. The Application Management Software supports download of code and data to NVM by the Composite Product Manufacturer before Operational Usage (for example, during development).

The documentation includes a Data Sheet with several addenda, such as System Interface Specification or Crypto Library user manuals, description of the Instruction Set or guidance documentation. This documentation comprises a description of the architecture, the secure configuration and usage of the IC hardware platform and the IC Dedicated Support Software by the Security IC Embedded Software.

The security functionality of the TOE is designed to act as an integral part of a complete security system in order to strengthen the design as a whole. Several security mechanisms are completely implemented in and controlled by the TOE. Other security mechanisms allow for configuration by or even require support of the Security IC Embedded Software.
E201382 is dedicated to application specific operating systems in the fields of public transport, access control, loyalty programs and retail, which allow a reduction of available resources such as memory sizes or coprocessors. Hence, E201382 shall maintain:

- The integrity and the confidentiality of code and data stored in its memories,
- The different TOE modes with the related capabilities for configuration and memory access and
- The integrity, the correct operation and the confidentiality of security functionality provided by the TOE.

This is ensured by the construction of the TOE and its security functionality.
NXP Secure Smart Card Controller E201382 basically provides a hardware platform for an implementation of a smart card application with:

- Functionality to calculate Data Encryption Standard (Triple-DES) with up to three keys.
- Hardware to calculate Advanced Encryption Standard (AES)
- A True Random Number Generator.
- A Deterministic Random Number Generator.
- Memory management control.
- Cyclic redundancy check (CRC) calculation, and
- ISO/IEC14443A contactless interface.

In addition, several security mechanisms are implemented to ensure proper operation as well as integrity and confidentiality of stored data. For example, this includes security mechanisms for memory protection and security exceptions as well as sensors, which allow operation under specified conditions only. Memory encryption is used for memory protection and chip shielding is added to the chip.

1.3.2 TOE Type

The TOE NXP Secure Smart Card Controller E201382 is provided as an IC hardware platform with IC Dedicated Software for various operating systems and applications with high security requirements.

1.3.3 Required non-TOE Hardware/Software/Firmware

None

1.4 TOE Description

1.4.1 Physical Scope of TOE

E201382 is manufactured in an advanced $140nm$ CMOS technology. A block diagram of the IC hardware is depicted in Figure 1.1.

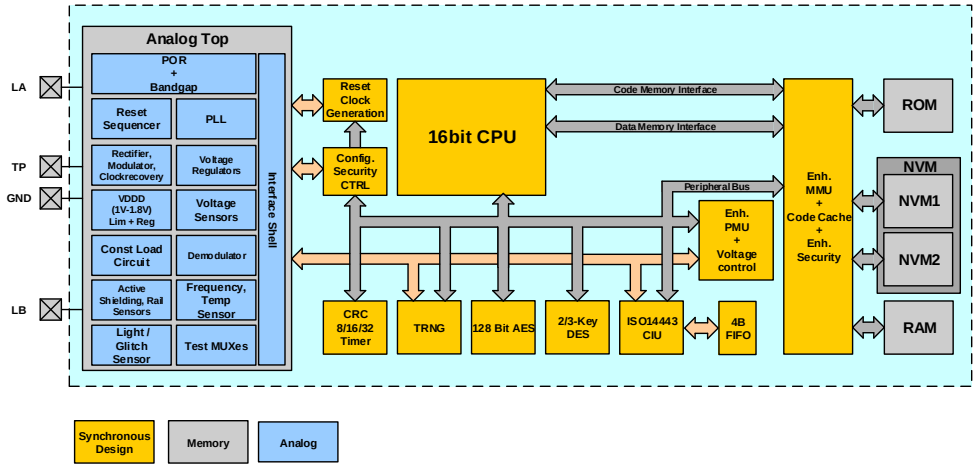

Fig. 1.1: Block Diagram

E201382 consists of the IC hardware and IC Dedicated Software. The IC Dedicated Software is composed of IC Dedicated Test Software for test purposes and IC Dedicated Support Software. The IC Dedicated Test Software contains the Test Software, the IC Dedicated Support Software is composed of the Boot Software, the HAL Software, the HAL Library, the Crypto Library and the Application Management Software . All other software is called Security IC Embedded Software. The Security IC Embedded Software is not part of the TOE. All components of the TOE are listed in section 1.4.1.1.

1.4.1.1 TOE components

Type	Name	Release	Date	Form of Delivery
IC Hardware	E201382	VA	11.06.2015	Wafer, modules and package
		VB	11.06.2015	Wafer, modules and package
IC Dedicated Test Software	Test Software	1.0	18.06.2015	part of System_Mode_combined_rom.dat
IC Dedicated Support Software	Boot Software	1.0	18.06.2015	part of System_Mode_combined_rom.dat

Type	Name	Release	Date	Form of Delivery
	HAL Software	1.0	18.06.2015	part of System_Mode_combined_rom.dat
	HAL Library	1.0	18.06.2015	part of System_Mode_combined_rom.dat
	Application Management Software	1.0	18.06.2015	part of System_Mode_combined_rom.dat
	Crypto Library	1.0	18.06.2015	part of System_Mode_combined_rom.dat
Document	E201382 EMBRACE, Secure smart card controller, Product Data Sheet [20]	254231	2017-01-20	Electronic Document
Document	E201382 EMBRACE, Instruction Set Manual, Product Data Sheet addendum [18]	277311	2015-06-25	Electronic Document
Document	E201382 EMBRACE, System Interface Manual, Product Data Sheet addendum [19]	279815	2017-01-17	Electronic Document
Document	E201382 EMBRACE, Application Management, Product Data Sheet addendum [16]	279913	2017-01-17	Electronic Document
Document	E201382 EMBRACE, Crypto Library, Product Data Sheet addendum [17]	280015	2016-12-15	Electronic Document
Document	E201382 VA and VB, Wafer and delivery specification, Product Data Sheet addendum [7]	340611	2016-02-02	Electronic Document
Document	NXP Secure Smart Card Controller E201382, Information on Guidance and Operation, Guidance and Operation Manual [14]	281117	2017-09-20	Electronic Document

Tab. 1.1: Components of the TOE

The IC Hardware is identified by its nameplate, that is located in the layout of the chip (see [20] how to inspect the nameplate). Note that IC Hardware versions VA and VB only differ with respect to the Antenna Configuration, where VA is associated with 17pF and VB with 70 pF. The IC Dedicated Software is identified by 'IC Dedicated Software version', which can be read out by the Security IC Embedded Software via a GetVersion command as described in [19]. Notice that the IC Dedicated Software version is reset to 00h for each new IC Hardware version. Notice, that System_Mode_combined_rom.dat will be delivered to the customer together with the toolchain which allows efficient development. However, the libraries will finally only be used for linking purposes and are not

mixed with customer code as they are permanently located in ROM whereas the customer code is submitted via the order entry procedure to NVM.

1.4.2 Evaluated Configurations

The customer selects logical and physical configuration options of E201382 without modification of its physical scope described in section 1.4.1. Logical configuration options are structured in major configuration options according to section 1.4.2.1 and minor configuration options according to section 1.4.2.2. Physical configuration options are the package types as detailed in section 1.4.2.3.

1.4.2.1 Major configuration options

Two major configuration are present, which are denoted by the names E201382 VA and E201382 VB. A major configuration is provided with several minor configuration options, which are introduced in Section 1.4.2.2.
The Order Entry Form [15] is individual to each type name. The first seven characters in the name of a major configuration give the type name and therewith the Order Entry Form [15] belonging to.
The major configuration is provided with several minor configuration options, which are defined in section 1.4.2.2.

1.4.2.2 Minor configuration options

Minor configurations are chosen by the customer via Order Entry Form [15] as detailed in Table 1.2. The Order Entry Form [15] identifies the minor configuration options, which are supported by a major configuration.

Product option	Choices	Description
Default application	1 or 2	This option determines which application shall be run by default. Default value is 1.
MIFARE options	No MIFARE or MIFARE Classic 1K or MIFARE Classic 4K	This option determines whether and which MIFARE Classic configuration shall be available. Only available if number of user applications is 1. Default value is No MIFARE.
MIFARE's Data Memory Location	EEPROM or FLASH	This option determines the default data memory location for MIFARE Classic. Default value is EEPROM.
Number of anti-tearing pages	Value between 0 and 128	This value determines the number of anti-tearing pages. Default value is 4.
Product life cycle at delivery	*Release*, *Pre-Release* or *Application Management*	This value determines in which life cycle the TOE is delivered. Default value is *Release*.
Antenna Configuration	17pF or 70pF	This value determines the antenna configuration. Default value is 17pF.
UID options	Double (7-byte) or Single (4-byte using revolving xFh range) or Random (4-byte)	This value determines the UID setting. Default value is Double (7-byte).
Contactless Communication Protocol	ISO/IEC 14443-4 (T=CL) or ISO/IEC 14443-3 (proprietary) or EMVCo 2.1 (modified T=CL)	This value determines the contactless communication protocol. Default value is ISO/IEC 14443-4 (T=CL).

Product option	Choices	Description
ISO/IEC 14443 communication data rates enable	106, 212, 424, 848 kbit/s or 106, 212, 424 kbit/s or 106, 212 kbit/s or 106 kbit/s	This value determines the enabled communication data rates. Default value is 106, 212, 424, 848 kbit/s.

Tab. 1.2: Evaluated minor configuration options

1.4.2.3 Evaluated package types

The commercial types are named according to the following format.

- E2*nxxfepp(p)/mvrrffyy*

Italic characters in the above format are replaced as described in Table 1.3 and Table 1.4 to retrieve a commercial type name. The commercial type name is composed of fixed symbols, which are detailed in Table 1.3, and variable entries, which are filled in according to the rules in Table 1.4.

Variable	Description	Values	Evaluated Options
n	Number of E2 generation	numeric	'0' for evolution 0
xx	Interface and Feature Configuration	alpha numeric	'13'
f	FLASH Memory Size	alphanumeric in 8K steps	'8' for 64KB
e	EEPROM Memory Size	alphanumeric in 5K steps	'2' for 10KB
pp(p)	Package delivery type	alpha numeric	see table 1.4
/	separator (mandatory)		
m	Manufacturer identifier	alpha numeric	't' for SSMC
v	Version of mask set	alphabetic	'A' for HW version VA
			'B' for HW version VB
rr	FLASH code number, which identifies the FLASH mask	alpha numeric	customer individual
ff	FabKey number, which identifies the EEPROM content at TOE delivery	alpha numeric	customer individual
yy	MIFARE Emulations	alpha numeric	'00' No MIFARE
			'M0' MIFARE Classic
			'M1' MIFARE Plus EV1
			'D2' MIFARE DESFire EV2

Tab. 1.3: Variable Definitions for Commercial Type Names

E201382	Description
Ux	Wafer not thinner than 50μm (The letter "x" in "Ux" stands for a capital letter or a number, which identifies the wafer type)
A4	MOA4

E201382	Description
A6	MOB6

Tab. 1.4: Supported Package Types

For example, commercial type name E201382U13/tA010100 means: E20 optimized secure contactless smart card controller, 10 KBytes EEPROM, 64 KBytes FLASH, 8 inch wafer with bumps (sawn; 120 μm thickness, on file, frame carrier, electronic fail die marking according to SECS-II format), SSMC, maskset version A, FLASH code Nr 01, FabKey Nr 01, No MIFARE. The characters 'rr' and 'ff' are individual for each customer product. The package types do not influence the security functionality of the TOE. They only define which pads are connected in the package and for what purpose and in which environment the chip can be used. Note that the security of the TOE is not dependent on which pad is connected or not – the connections just define how the product can be used. If the TOE is delivered as wafer the customer can choose the connections on his own.
Security during development and production is ensured for all package types listed above, for details refer to section 1.4.4.
The commercial type name identifies major configuration and package type of the TOE as well as the Security IC Embedded Software. However, the commercial type name does not itemize the minor configuration options of the TOE, which are introduced in section 1.4.2.2. Instead, minor configuration options are identified in the Order Entry Form, which is assigned to the FLASH code number and the FabKey number of the commercial type name.

Remark 1. Please note that MIFARE Classic is not part of the TOE.

1.4.3 Logical Scope of TOE

1.4.3.1 Hardware Description

The TOE distinguishes three TOE modes:

1. **Super System Mode (SSM)**
2. **System Mode (SM)**
3. **User Mode (UM)**

The Super System Mode is not available to the Security IC Embedded Software. In Super System Mode the TOE executes the Boot Software and the IC Dedicated Test Software. Notice that parts of the HAL Software execute also in Super System Mode and other parts are executed in System Mode and can be accessed via so-called system calls either from User Mode or System Mode. HAL Library and Crypto Library are in a shared (System Mode and User Mode access) code space. The Application Management Software runs in System Mode. The Security IC Embedded Software executes in User Mode. Note also that the CPU itself only distinguishes between the User Mode and the System Mode. From CPU's perspective there is no difference between the System Mode and the Super System Mode. The difference from system perspective is only that the Super System Mode can extend its access rights to Special Function Registers compared to what is visible in System Mode (it can grant

access to test features). However, this is enforced by the Memory Management Unit where the Super System Mode is modelled as an own mode (in that context sometimes referred to as 'Test Mode') that has extended access rights compared to System Mode.
During phase 3 IC Manufacturing according to the Security IC product life-cycle in the PP [21], start-up and reset of E201382 always complete with Test Mode and execution of the Test-ROM Software. Test Mode and Test-ROM Software are permanently disabled before TOE Delivery according to the Security IC product life-cycle [21]. Then start-up and reset of E201382 always end up in System Mode.
User Mode is available to the developer of the Security IC Embedded Software. System Mode has unlimited access to the hardware components available to the Security IC Embedded Software. User Mode has restricted access to the CPU, specific Special Function Registers and the memories depending on the access rights granted by software running in System Mode. The hardware components are controlled by the Security IC Embedded Software via Special Function Registers or the hardware abstraction software. Both are interrelated to the activities of the CPU, the Memory Management Unit, interrupt control, I/O configuration, NVM, timers and the coprocessors.
The E201382 provides interrupts. Interrupts force a jump to a specific fixed vector address in the ROM. Any interrupt can therefore be controlled and guided by a specific part of the Security IC Embedded Software. In addition, E201382 provides user calls and system calls. These calls have to be explicitly done by the Security IC Embedded Software via dedicated CPU instructions. A user call starts the execution of related code dedicated to User Mode, a system call starts the execution of related code dedicated to System Mode except SYS0 which executes test functionality run in Super System Mode.
The Watchdog timer is intended to abort irregular program executions by a time-out mechanism and is enabled and configured by the Security IC Embedded Software.
The E201382 incorporates 48 kBytes of ROM, 1280 Bytes of RAM, 10 kBytes of EEPROM and 64 kBytes of FLASH. The memories EEPROM and FLASH are in the following combined under the term *NVM*. Access control to all four memory types is enforced by a Memory Management Unit (MMU). The MMU partitions each memory into several parts, defined as objects in the Hardware Access Control Policy (see section 6.1.6).
The Triple-DES coprocessor supports single DES and Triple-DES operations. Only Triple-DES is in the scope of this evaluation, in 2- key or 3-key operation with two/three 56-bit keys (112-/168-bit). The AES coprocessor supports AES operation with 128-bit key length. The random number generator provides true random numbers without pseudo random calculation. The deterministic random number generator provides pseudo-random calculation seeded by the true random number generator. The CRC coprocessor provides CRC generation polynomial CRC-8, CRC-16 and CRC-32.
The TOE protects secret data, which are stored to and operated by the TOE, against physical tampering. A memory encryption is added to the memories RAM, ROMand NVM. Chip shielding is added in form of active and passive shield over logic and memories. Sensors in form of light, voltage, temperature and frequency sensors are distributed over the chip area. The security functionality of the IC hardware platform is mainly provided by the TOE, and completed by the Security IC Embedded Software. This causes dependencies between the security functionality of the TOE and the security functionality provided by the Security IC Embedded Software.

1.4.3.2 Software Description

Figure 1.2 illustrates the different pieces of software. Operating system and applications of a Security IC are developed by the customers and included under the heading Security IC Embedded Software. The Security IC Embedded Software depends on the usage of the IC hardware platform. It is stored in the UM_Code_Seg for customers developing code for User Mode (see Figure 1.2; "Customer Code"). It is not part of the TOE. There are two use cases relevant for Customer Code:

One Application in NVM: the Application can store data in EEPROM and FLASH respectively.

Two Applications in NVM: Application 1 can store data only in EEPROM, Application 2 can store data only in FLASH.

Notice, that if two applications are available they cannot be executed in parallel. There needs to be a switch operation triggered which shuts down the currently running applications, cleans up the temporary data and boots the other application.

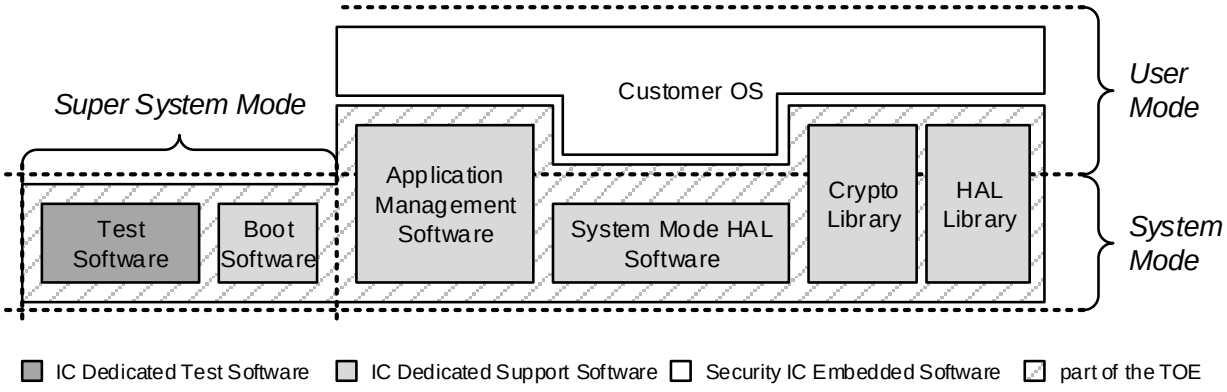

Fig. 1.2: Software Components of the TOE

The IC Dedicated Software comprises the IC Dedicated Test Software and the IC Dedicated Support Software described in the following.
The IC Dedicated Test Software is developed by NXP and embedded in the Test Software. The IC Dedicated Test Software includes the test operating system, test routines for the various blocks of the circuitry, control flags for the status of the EEPROM's manufacturer area and shut down functions to ensure that security relevant test routines cannot be executed illegally after phase 3. This is stored in the NXP_ConfigData_Seg. Moreover, the IC Dedicated Test Software is used to download patch code related to System Mode (stored in SM_PatchCode_Seg)
.
The IC Dedicated Support Software comprises the following five parts:

1. The Boot Software is executed after each reset of the TOE, i.e. every time when the TOE starts. It sets up the TOE and does some basic configuration of the hardware based on the settings stored in NXP_ConfigData_Seg . The Boot Software is stored in the BootTestCode_Seg.

2. The HAL Software is partly stored in the BootTestCode_Seg and partly stored in the SM_Code_Seg and accessed by the Security IC Embedded Software via system calls. It provides basic NVM access and basic System functionality like self-testing, error-counter handling and reset functionality. Notice, that Boot Software and IC Dedicated Test Software also access HAL Software. Some of the functionality is exclusively available to the latter two.

3. The Application Management Software is stored in the ROM and cannot be directly accessed by the Security IC Embedded Software. It provides functionality by which the Security IC Embedded Software developer can manage the application during its development, such as download, verify and lock of an application. Depending on the life-cycle of the TOE at delivery, this functionality is either locked by NXP (life-cycle *Release*), or must be locked by customer (life-cycles *Pre-Release* and *Application Management*) before the Operational Usage.

4. The HAL Library is stored in the SharedCode_Seg and provides several non-crypto related functions to the Security IC Embedded Software.

5. The Crypto Library is stored in the SharedCode_Seg and provides functions to access AES, DES and RNG functionality to the Security IC Embedded Software.

1.4.3.3 Documentation

The following documents contain a functional description and guidelines for the use of the security functionality, as needed to develop Security IC Embedded Software:

- ”E201382 EMBRACE, Secure smart card controller, Product Data Sheet, NXP Semiconductors” [20] contains a basic functional description of the hardware
- ”E201382 EMBRACE, System Interface Manual, Product Data Sheet addendum, NXP Semiconductors” [19] contains the interface description of HAL Software and HAL Library together with a collection of Special Function Registers accessible in User Mode
- ”E201382 EMBRACE, Application Management, Product Data Sheet addendum, NXP Semiconductors” [16] contains the description of Application Management Software.
- ”E201382 EMBRACE, Instruction Set Manual, Product Data Sheet addendum, NXP Semiconductors” [18] contains a detailed specification of the CPU instructions
- ”E201382 EMBRACE, Crypto Library, Product Data Sheet addendum, NXP Semiconductors” [17] describes AES, DES and the Random Number Generator of the interface description of Crypto Library.

- ”E201382 VA and VB, Wafer and delivery specification, Product Data Sheet addendum, NXP Semiconductors” [7] describes physical identification of the TOE and the secure delivery process.
- ”NXP Secure Smart Card Controller E201382, Information on Guidance and Operation, Guidance and Operation Manual” [14] describes aspects of the program interface and the use of programming techniques to improve the security.

The whole documentation shall be used by the developer to develop the Security IC Embedded Software.

1.4.4 Security during Development and Production

During the design and the layout process only people involved in the specific development project for an IC have access to sensitive data. Different people are responsible for the design data and for customer related data. The production of the wafers includes two different steps regarding the production flow. In the first step the wafers are produced with the fixed masks independent of the customer. After that step the wafers are completed with the customer specific mask, including the FLASH Code, and the remaining mask set. The test process of every die is performed by a test center of NXP. Delivery processes between the involved sites provide accountability and traceability of the TOE. NXP embeds the dice into modules, inlays or packages based on customer demand.
Information about non-functional items is stored on magnetic/optical media enclosed with the delivery or the non-functional items are physically marked. In summary, the TOE can be delivered in Wafer and modules. The availability of major configuration options of the TOE in package types is detailed in section 1.4.2.1.

1.4.5 Life-Cycle and Delivery of the TOE

The Security IC product life-cycle is scheduled in phases as introduced in the Security IC Platform Protection Profile with Augmentation Packages [21], section 1.2.4. IC Development as well as IC Manufacturing and Testing, which are phases 2 and 3 of the life-cycle, are part of the evaluation. Phase 4 IC Packaging is also part of the evaluation. The Security IC is delivered at the end of phase 3 or phase 4 in the life-cycle. Therefore the TOE evaluation perimeter, comprising the development and production environment of the TOE, consists of life-cycle phases 2 - 4. With respect to Application Note 3 in [21] the TOE supports the authentic delivery using the FabKey feature. For further details please refer to the data sheet [20] and the guidance and operation manual [14].

1.4.6 TOE Intended Usage

The final product as a combination of the hardware platform and the compiled Security IC Embedded Software comprising the operating system and application are used by the end-user (phase 7). The method of use of the product in this phase depends on the application. The TOE is intended to be used in an insecure environment that does not protect against threats.
The device is developed for most high-end safeguarded applications, and is designed for embedding into contactless smart cards according to ISO 14443, [23]. Usually the smart card is assigned to a single individual only although the smart card may be expected to be used for multiple applications. So the TOE must meet security

requirements to be applied to security modules. Secret data stored on the smart card shall be used as input for the calculation of authentication data, the calculation of signatures and the encryption of data and keys.
The TOE user environment is the environment from TOE Delivery to phase 7. At the phases up to 6, the TOE user environment must be a controlled environment.
In the end-user environment (phase 7) Security ICs are used in a wide range of applications to assure authorized conditional access. Examples of such are Pay-TV, Banking Cards, Portable communication SIM cards, Health cards, Transportation cards. The end-user environment therefore covers a wide spectrum of very different functions, thus making it difficult to avoid and monitor any abuse of the TOE.

1.4.7 Interface of the TOE

The electrical interfaces of the TOE are the pads (called LA and LB) for the antenna of the contactless interface unit. There are additional test pads which are exclusively accessible during the production testing before the delivery.
The software interface of the TOE depends on the TOE mode:

- The IC Dedicated Test Software is not available after delivery.
- The parts of the Application Management Software which are relevant for downloading, verifying and locking the device are not available after life-cycle Phase 5.
- The Boot ROM Software is executed in Test Mode. This software initializes and verifies the configuration of the TOE before the Security IC Embedded Software (after TOE delivery) starts in User Mode.
- After TOE Delivery the logical interface is defined by the Security IC Embedded Software. The Security IC Embedded Software is executed in User Mode. This is the only available Mode to the user. The Security IC Embedded Software is stored in the NVM.

Note: Note that the logical interface provided by the TOE for the Security IC Embedded Software is the set of functions defined in the HAL Software, HAL Library and Crypto Library, the address map of the CPU including memories, Special Function Registers and the instruction set of the CPU. This interface of the hardware platform is not an external interface of a composite product including the TOE.

The chip surface can be seen as an interface of the TOE, too. This interface is taken into account regarding environmental stress e.g. like temperature and in the case of an attack where the attacker tries to manipulate the chip surface or the chip.

Note: An external energy and timing supply as well as a data interface are necessary for the operation of the TOE. Beyond the physical behaviour the interface is defined by the application environment.

2 Conformance Claims

This Security Target claims to be conformant to the Common Criteria version 3.1:

- Common Criteria for Information Technology Security Evaluation, Part 1 – Introduction and general model - Version 3.1 CCMB-2012-09-001, Revision 4, September 2012, [3]
- Common Criteria for Information Technology Security Evaluation, Part 2 – Security functional components, Version 3.1 CCMB-2012-09-002, Revision 4, September 2012, [4]
- Common Criteria for Information Technology Security Evaluation, Part 3 – Security Assurance Components, Version 3.1 CCMB-2012-09-003, Revision 4, September 2012, [5]

For the evaluation the following methodology will be used:

- Common Methodology for Information Technology Security Evaluation CEM-99/045 Part 2 – Evaluation Methodology, Version 3.1 CCMB-2012-09-004, Revision 4, September 2012, [6]

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in chapter 6.

2.1 Package Claim

This Security Target claims conformance to the assurance package **EAL5 augmented**. The augmentations to EAL5 are ALC_DVS.2 and AVA_VAN.5. In addition, the Security Target is augmented using the component ASE_TSS.2, which is chosen to include architectural information on the security functionality of the TOE.

Note: The Protection Profile (PP) ”Security IC Platform Protection Profile with Augmentation Packages” [21] to which this Security Target claims conformance (refer to section 2.2) requires assurance level EAL4 augmented. The changes, which are needed for EAL5, are described in the relevant sections of this Security Target.

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

2.2 PP Claim

This Security Target claims strict conformance to the Security IC Platform Protection Profile with Augmentation Packages, [21]. Thus, the concepts are used in the same sense. For the definition of terms refer to [21]. This chapter does not need any supplement in the Security Target.
The Protection Profile (PP) ”Security IC Platform Protection Profile with Augmentation Packages” [21] defines ”Augmentation Packages”. This Security Target includes the augmentation package ”Loader Package 1” defined in chapter 7.3.1 of the PP [21].
The TOE provides additional functionality, which is not covered in [21]. In accordance with Application Note 4 of [21] this additional functionality is added using the policy P.Add-Components (see section 3.3).

2.3 Conformance Claim Rationale

According to section 2.2 this ST claims strict conformance to the Security IC Platform Protection Profile with Augmentation Packages [21].
The TOE type defined in section 1.3.2 of this Security Target is a smart card controller with IC Dedicated Software. This is consistent with the TOE definition for a Security IC in section 1.2.2 of [21].
The sections within this document where security problem definitions, security objectives and security requirements are defined, clearly state which of these items are taken from the Protection Profile and which are added in this Security Target. Therefore the content of the Protection Profile is not repeated here. Moreover, all additionally stated items in this Security Target do not contradict the items included from the PP (see the respective sections in this document). The operations done for the SFRs taken from the PP are also clearly indicated.
The evaluation assurance level claimed for this TOE is shown in section 6.2 to include respectively exceed the requirements claimed by the PP (EAL4+).
These considerations show that the Security Target correctly claims conformance to the Security IC Platform Protection Profile with Augmentation Packages, [21].

3 Security Problem Definition

This chapter lists the assets, threats, assumptions and organizational security policies from the PP [21] and describes extensions to these elements in detail.

3.1 Description of Assets

All assets, which are defined in section 3.1 of the PP [21], are related to standard functionality. They are applied in this Security Target. These assets are:

- Integrity and confidentiality of User Data stored and in operation,
- Integrity and confidentiality of Security IC Embedded Software, stored and in operation,
- Correct operation of the Security Services provided by the TOE for the Security IC Embedded Software,
- Deficiency of random numbers.

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

- Logical design data, physical design data, IC Dedicated Software, configuration data,
- Initialization data and pre-personalization data, specific development aids, data related to test and characterization, material for software development support, photo masks.

Note that the keys for cryptographic calculations using security services of the TOE are treated as User Data.

3.2 Threats

All threats, which are defined in section 3.2 of the PP [21], are valid for this Security Target. These threats are listed in Table 3.1. In addition the threat T.Masquerade_TOE is applicable for this TOE as listed below.

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

Name	Title
T.Leak-Inherent	Inherent Information Leakage
T.Phys-Probing	Physical Probing
T.Malfunction	Malfunction due to Environmental Stress
T.Phys-Manipulation	Physical Manipulation
T.Leak-Forced	Forced Information Leakage
T.Abuse-Func	Abuse of Functionality

Name	Title
T.RND	Deficiency of Random Numbers
T.Masquerade_TOE	Masquerade the TOE

Tab. 3.1: Threats defined in the Security IC Platform Protection Profile with Augmentation Packages

In compliance with Application Note 4 in the PP [21] the TOE provides additional functionality to protect against threats that appear when the TOE is used for multiple applications.
The TOE provides the Security IC Embedded Software running in System Mode with control of access to memories and hardware components by different applications running in User Mode. In this context, User Data of different applications is stored to such memory and processed by such hardware components. The Security IC Embedded Software controls all this User Data. Any access to User Data assigned to one application by another application contradicts separation between different applications and is considered as a threat.
The TOE shall avert threat T.Unauthorised-Access as specified below.

T.Unauthorised-Acce ss **Unauthorized Memory or Hardware Access**

Adverse action: An attacker may try to read, modify or execute code or data stored in restricted memory areas. An attacker may try to access or operate hardware resources that are restricted by executing code that accidentally or deliberately accesses these restricted hardware resources. Any code executed or data used in Boot Mode, System Mode or User Mode may accidentally or deliberately access code or User Data of other applications. Any code executed or data used in Boot Mode, System Mode or User Mode may accidentally or deliberately access hardware resources that are restricted to other applications.

Threat agent: Attacker having high attack potential and access to the TOE.

Asset: Code executed by and data belonging to the IC Dedicated Support Software running in Super System Mode as well as code executed by and data belonging to the Security IC Embedded Software.

Restrictions of access to memories and hardware resources, which are available to the Security IC Embedded Software, must be defined and implemented by the security policy of the Security IC Embedded Software based on the specific application context.
The threats defined in this Security Target are summarized in Table 3.2.

Name	Title
T.Unauthorised-Access	Unauthorized Memory or Hardware Access

Tab. 3.2: Additional Threats defined in this ST

3.3 Organizational Security Policies

All security policies, which are defined in section 3.3 of the PP [21], are valid for this Security Target. These security policies are listed in Table 3.3.

Name	Title
P.Process-TOE	Identification during TOE Development and Production
P.Lim_Block_Loader	Limiting and Blocking the Loader Functionality

Tab. 3.3: Policies defined in the Security IC Platform Protection Profile with Augmentation Packages

In compliance with Application Note 5 in the PP [21], this Security Target defines one additional security policyas detailed below.
The TOE provides specific security functionality, which can be used by the Security IC Embedded Software. This specific security functionality is not derived from threats identified for the TOE. Instead, the Security IC Embedded Software decides how to use this security functionality to protect from threats for the composite product. Thus, security policy P.Add-Components is defined as follows.

P.Add-Components **Additional Specific Security Components**
The TOE shall provide the following additional security functionality to the Security IC Embedded Software:

- Triple DES encryption and decryption
- AES encryption and decryption
- Deterministic Random Number Generation
- Mechanism to provide protection of residual information
- Self Testing
- A function to reset the device
- Integrity support of data stored to NVM

The security policies defined in this Security Target are summarized in Table 3.4.

Name	Title
P.Add-Components	Additional Specific Security Components

Tab. 3.4: Additional Security Policies defined in this ST

3.4 Assumptions

All assumptions, which are defined in section 3.4 of the PP [21], are valid for this Security Target. These assumptions are listed in Table 3.5.

Name	Title
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation
A.Resp-Appl	Treatment of user data of the Composite TOE

Tab. 3.5: Assumptions defined in the Security IC Platform Protection Profile with Augmentation Packages

In compliance with Application Notes 6 and 7 in PP [21], this Security Target defines two additional assumptions as follows.

A.Check-Init **Check of initialization data by the Security IC Embedded Software**
The Security IC Embedded Software must provide a function to check initialization data. Such data is defined by the Composite Product Manufacturer and injected by the TOE Manufacturer into the non-volatile memory to provide the ability to identify and trace the TOE.

The following additional assumption considers specialized encryption hardware of the TOE.
The developer of the Security IC Embedded Software must ensure appropriate usage of key-dependent functions as defined below during phase 1 of the Security IC product life-cycle [21].

A.Key-Function **Usage of Key-dependent Functions**
Key-dependent functions (if any) shall be implemented in the Security IC Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced).
Note that here the routines which may compromise keys when being executed are part of the Security IC Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines which are part of the TOE and (ii) the processing of User Data including cryptographic keys.

The assumptions defined in this Security Target are summarized in Table 3.6.

Name	Title
A.Check-Init	Check of initialization data by the Security IC Embedded Software
A.Key-Function	Usage of Key-dependent Functions

Tab. 3.6: Additional Assumptions defined in this ST

4 Security Objectives

This chapter defines the security objectives that shall be met by the TOE, the Security IC Embedded Software Development Environment and the Operational Environment.

4.1 Security Objectives for the TOE

All security objectives for the TOE, which are defined in the PP [21], are applied to this Security Target. These security objectives are listed in Table 4.1.

Name	Title
O.Leak-Inherent	Protection against Inherent Information Leakage
O.Phys-Probing	Protection against Physical Probing
O.Malfunction	Protection against Malfunctions
O.Phys-Manipulation	Protection against Physical Manipulation
O.Leak-Forced	Protection against Forced Information Leakage
O.Abuse-Func	Protection against Abuse of Functionality
O.Identification	TOE Identification
O.RND	Random Numbers
O.Cap_Avail_Loader	Capability and availability of the Loader

Tab. 4.1: Security Objectives of the TOE defined in the Security IC Platform Protection Profile with Augmentation Packages

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

In compliance with Application Notes 8 and 9 in the PP [21], additional security objectives for the TOE are defined below based on additional functionality provided by the TOE.

O.DES **Data Encryption Standard**
The TOE shall provide the cryptographic functionality to calculate Triple DES encryption and decryption over one up to several blocks in the following modes of operation: ECB, CBC, CBC-MAC and CMAC.

O.AES **Advanced Encryption Standard**
The TOE shall provide the cryptographic functionality to calculate AES encryption and decryption over one up to several blocks in the following modes of operation: ECB, CBC, CBC-MAC and CMAC.

O.INTEGRITY_CHK **Integrity Control of Transferred Data**
The TOE shall provide integrity protection of User Data and TSF data transferred between different parts of the TOE. This comprises data transfer between memories or between a memory and a hardware resource of the TOE.

O.NVM_INTEGRITY **Integrity Support of data stored to NVM**
The TOE shall provide detection and correction of failures in NVM memories to support integrity of contents stored there.

O.MEM_ACCESS **Area based Memory Access Control**
The TOE shall control access of CPU instructions to memory areas depending on memory partitioning and based on TOE modes Super System Mode, System Mode and User Mode. In Super System Mode, System Mode and User Mode the TOE shall control access also based on configuration. In User Mode, the TOE shall control access also based on memory segments, which are configured in System Mode when implementing a memory management scheme. This control shall be individual to each memory segment and consider different access rights.

O.SFR_ACCESS **Special Function Register Access Control**
The TOE shall control access of CPU instructions to Special Function Registers depending on the purpose of the register and based on TOE modes. In Super System Mode and System Mode, the TOE shall have full access to the Special Function Registers. In User Mode, the TOE shall deny access to any Special Function Register.

O.HW_REUSE **Application reuse of Memory**
The TOE shall include measures to ensure that the memory resources being used by an application of the TOE cannot be disclosed to subsequent users of the same memory resource of another application.

O.Self-Test **Self Test**
The TOE shall include functionality to perform a self-test to detect physical manipulation.

O.Reset **Reset function**
The TOE shall provide the Security IC Embedded Software with a function to reset the device.

O.REUSE **Crypto reuse of Memory**
The TOE shall provide the measures to ensure that the memory resources being used by the TOE for the CryptoLib cannot be disclosed to subsequent users of the same memory resource.

The objectives of the TOE defined in this Security Target are summarized in Table 4.2.

Name	Title
O.INTEGRITY_CHK	Integrity Control of Transferred Data
O.NVM_INTEGRITY	Integrity Support of data stored to NVM
O.MEM_ACCESS	Area based Memory Access Control
O.SFR_ACCESS	Special Function Register Access Control
O.HW_REUSE	Application reuse of Memory
O.Self-Test	Self Test
O.Reset	Reset function

Tab. 4.2: Security Objectives of the TOE defined in this ST (Part 1)

Name	Title
O.AES	Advanced Encryption Standard
O.DES	Data Encryption Standard
O.REUSE	Crypto reuse of Memory

Tab. 4.3: Security Objectives of the TOE defined in this ST (Part 2)

4.2 Security Objectives for the Security IC Embedded Software Development Environment

All security objectives for the Security IC Embedded Software development Environment, which are defined in the PP [21], are applied to this Security Target. These security objectives are listed in Table 4.4.

Name	Title
OE.Resp-Appl	Treatment of User Data

Tab. 4.4: Security Objectives of the Development Environment defined in the Security IC Platform Protection Profile with Augmentation Packages

Clarification related to "Treatment of User Data (OE.Resp-Appl)"
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.
In case the Security IC Embedded Software operates multiple applications on the TOE, OE.Resp-Appl must also be met. The Security IC Embedded Software must not disclose security relevant User Data of one application to another application when processed in or stored to the TOE.

4.3 Security Objectives for the Operational Environment

In addition to the security objective for the operational environment as required by CC Part 1 [3] all security objectives for the operational environment, which are defined in the PP [21], are applied to this Security Target. These security objectives are listed in Table 4.5.

Name	Title
OE.Process-Sec-IC	Protection during composite product manufacturing
OE.Lim_Block_Loader	Limitation of capability and blocking the Loader

Tab. 4.5: Security Objectives of the Operational Environment defined in the Security IC Platform Protection Profile with Augmentation Packages

The following additional security objectives for the operational environment are defined in this Security Target.
The following security objective for the operational environment derives from assumption A.Check-Init. The TOE provides specific functionality that requires the TOE Manufacturer to implement measures for unique identification

of the TOE. Security objective OE.Check-Init is defined to allow for such a TOE specific implementation.

OE.Check-Init **Check of initialization data by the Security IC Embedded Software**
To ensure the receipt of the correct TOE, the Security IC Embedded Software shall check a sufficient part of the pre-personalization data. This shall include at least the FabKey Data that is agreed between the customer and the TOE Manufacturer.

The objectives for the operational environment defined in this Security Target are summarized in Table 4.6.

Name	Title
OE.Check-Init	Check of initialization data by the Security IC Embedded Software

Tab. 4.6: Security Objectives of the Operational Environment defined in this ST

4.4 Security Objectives Rationale

Section 4.4 in the PP [21] provides a rationale how the threats, organisational security policies and assumptions are addressed by the security objectives defined in the PP [21]. Table 4.7 summarizes how threats, organisational security policies and assumptions of the PP are addressed by security objectives defined in the PP and ST, respectively. All these items are in line with those in the PP [21].

Security Problem Definition	Security Objective	Notes
T.Leak-Inherent	O.Leak-Inherent	
T.Phys-Probing	O.Phys-Probing	
T.Malfunction	O.Malfunction O.Self-Test O.INTEGRITY_CHK	
T.Phys-Manipulation	O.Phys-Manipulation O.Self-Test	
T.Leak-Forced	O.Leak-Forced	
T.Abuse-Func	O.Abuse-Func	
T.RND	O.RND	
P.Process-TOE	O.Identification	Phases 2–3
A.Process-Sec-IC	OE.Process-Sec-IC	Phases 4–6
A.Resp-Appl	OE.Resp-Appl	Phase 1
T.Masquerade_TOE	OE.Process-Sec-IC	
P.Lim_Block_Loader	O.Cap_Avail_Loader OE.Lim_Block_Loader	

Tab. 4.7: Security Objectives (PP and ST) vs. Security Problem Definition (PP)

Table 4.8 summarizes how threats, organisational security policies and assumptions of this ST are addressed by security objectives defined in the PP and ST, respectively.

Security Problem Definition	Security Objective	Notes
T.Unauthorised-Access	O.MEM_ACCESS O.SFR_ACCESS	
P.Add-Components	O.AES O.DES O.REUSE O.RND O.HW_REUSE O.Self-Test O.Reset O.NVM_INTEGRITY	
A.Check-Init	OE.Check-Init	Phases 1 and 4–6
A.Key-Function	OE.Resp-Appl	Phase 1

Tab. 4.8: Security Objectives (PP and ST) vs. Security Problem Definition (ST)

The rationale for the threat T.Masquerade_TOE is given below:
Justification related to T.Masquerade_TOE:

Objective	Rationale
OE.Process-Sec-IC	The Security Objective for the Operational Environment requires that the confidentiality and integrity of the TOE is maintained. Thus the threat is covered.

The rationale for all items defined in the Security Target is given below.
Justification related to T.Unauthorised-Access:

Objective	Rationale
O.MEM_ACCESS	TOE must enforce memory partitioning with address mapping and control of access to memories in System Mode and User Mode. Access rights in User Mode must be explicitly granted by Security IC Embedded Software running in System Mode. Thus, security violations caused by accidental or deliberate access to restricted data, code and shared hardware resources can be prevented.

Objective	Rationale
O.SFR_ACCESS	The TOE must enforce control of access to Special Function Registers in System Mode and User Mode. Access rights in User Mode must be explicitly granted by code running in System Mode. Thus, security violations caused by accidental or deliberate access to restricted data, code and shared hardware resources can be prevented.

Justification related to P.Add-Components:

Objective	Rationale
O.AES	This objective covers the security policy because it requires the TOE to implement the functionality AES as required by the security policy.
O.DES	This objective covers the security policy because it requires the TOE to implement the functionality DES as required by the security policy.
O.REUSE	This objective covers the security policy because it requires the TOE to partly implement functionality to provide protection of residual information as required by the security policy.
O.RND	This objective covers the security policy because it requires the TOE to implement functionality to provide deterministic random number generation as required by the security policy.
O.HW_REUSE	This objective covers the security policy because it requires the TOE to partly implement functionality to provide protection against residual information as required by the security policy.
O.Self-Test	This objective covers the security policy because it requires the TOE to implement the functionality Self Testing as required by the security policy.
O.Reset	This objective covers the security policy because it requires the TOE to implement the functionality to reset the device as required by the security policy.
O.NVM_INTEGRITY	This objective covers the security policy because it requires the TOE to implement functionality of integrity support of data stored to NVM as required by the security policy.

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-Components. These security objectives are also valid for the additional specific security functionality since they must avert the related threats also for the components added related to the policy.

Justification related to A.Check-Init:

Objective	Rationale
OE.Check-Init	This objective requires the Security IC Embedded Software developer to implement a function as stated in this assumption.

Justification related to A.Key-Function:

Objective	Rationale
OE.Resp-Appl	The definition of this objective of the PP [21] is further clarified in this Security Target: 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. 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 implemented in the environment. In addition, the treatment of User Data comprises the implementation of a multi-application operating system that does not disclose security relevant User Data of one application to another one. These measures make sure that the assumption A.Key-Function is still covered by this objective.

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

5 Extended Components Definitions

This Security Target does not define extended components.

Note that the Security IC Platform Protection Profile with Augmentation Packages [21] defines extended security functional requirements FCS_RNG.1, FMT_LIM.1, FMT_LIM.2, FAU_SAS.1 and FDP_SDC.1 in chapter 5, which are included in this Security Target.

6 Security Requirements

This chapter defines the security requirements that shall be met by the TOE. These security requirements are composed of the security functional requirements and the security assurance requirements that the TOE must meet in order to achieve its security objectives. CC allows several operations to be performed on security requirements (on the component level); refinement, selection, assignment, and iteration are defined in section 8.1 of CC Part 1 [3]. These operations are used in the PP [21] and in this Security Target, respectively.

The **refinement** operation is used to add details to requirements, and thus, further intensifies a requirement.

The **selection** operation is used to select one or more options provided by the PP [21] or CC in stating a requirement. Selections having been made are denoted as italic text.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made are denoted as italic text.

The **iteration** operation is used when a component is repeated with varying operations. It is denoted by showing brackets ”[iteration indicator]” and the iteration indicator within the brackets.

For the sake of a better readability, the iteration operation may also be applied to some single components (being not repeated) in order to indicate belonging of such SFRs to same functional cluster. In such a case, the iteration operation is applied to only one single component.

Whenever an element in the PP [21] contains an operation that is left uncompleted, the Security Target has to complete that operation.

6.1 Security Functional Requirements

All Security Functional Requirements (SFRs) of the TOE are presented in the following sections to support a better understanding of the combination of the PP [21] and this Security Target. Tables 6.1 and 6.2 summarize the SFRs defined in the PP and ST, respectively.

Name	Title
FAU_SAS.1[HW]	Audit Storage
FCS_RNG.1[HW]	Random Number Generation (Class PTG.2)
FDP_ITT.1[HW]	Basic Internal Transfer Protection
FDP_IFC.1	Subset Information Flow Control
FDP_SDC.1[HW]	Stored data confidentiality
FDP_SDI.2[HW]	Stored data integrity monitoring and action
FMT_LIM.1[HW]	Limited Capabilities
FMT_LIM.1[Loader]	Limited Capabilities
FMT_LIM.2[HW]	Limited Availability
FMT_LIM.2[Loader]	Limited Availability
FPT_FLS.1	Failure with Preservation of Secure State
FPT_ITT.1[HW]	Basic Internal TSF Data Transfer Protection
FPT_PHP.3	Resistance to Physical Attack

Name	Title
FRU_FLT.2	Limited Fault Tolerance

Tab. 6.1: Security Functional Requirements defined in the Security IC Platform Protection Profile with Augmentation Packages

Name	Title
FCS_COP.1[HW_DES]	Cryptographic Operation (DES)
FCS_COP.1[HW_AES]	Cryptographic Operation (AES)
FDP_ACC.1[MEM]	Subset Access Control (Memories)
FDP_ACC.1[SFR]	Subset Access Control (Special Function Registers)
FDP_ACF.1[MEM]	Security Attribute Based Access Control (Memories)
FDP_ACF.1[SFR]	Security Attribute Based Access Control (Special Function Registers)
FDP_RIP.1[HW]	Subset Residual Information Protection
FMT_MSA.1[MEM]	Management of Security Attributes (Memories)
FMT_MSA.1[SFR]	Management of Security Attributes (Special Function Registers)
FMT_MSA.3[MEM]	Static Attribute Initialization (Memories)
FMT_MSA.3[SFR]	Static Attribute Initialization (Special Function Registers)
FMT_SMF.1[HW]	Specification of Management Functions (Hardware)
FMT_SMF.1[SW]	Specification of Management Functions (Software)
FPT_TST.1	TSF Testing

Tab. 6.2: Security Functional Requirements defined in this ST (Part 1)

Name	Title
FCS_RNG.1[DET]	Random Number Generation (Deterministic)
FCS_COP.1[SW_DES]	Cryptographic Operation (DES & TDES)
FCS_COP.1[SW_AES]	Cryptographic Operation (AES)
FDP_RIP.1[SW]	Subset Residual Information Protection

Tab. 6.3: Security Functional Requirements defined in this ST (Part 2)

6.1.1 SFRs of the Protection Profile

All SFRs, which are defined in the PP [21], are summarized in Table 6.1. Some of these SFRs are defined in CC Part 2 [4] and eventually subject to refinement, selection, assignment and/or iteration operation in the PP [21]. Others are newly defined in the PP [21].

SFRs FDP_ITT.1 and FPT_ITT.1 are defined in CC Part 2 [4] and are subject to refinement, selection and assignment operations in the PP [21]. The selection operations are further extended in this Security Target, which

results in the following SFRs. Iteration [HW] is done here to prepare for other iterations that address any future major configurations of the TOE. The TOE shall meet requirement FDP_ITT.1 as specified below.

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

The TOE shall meet requirement FPT_ITT.1 as specified below.

FPT_ITT.1[HW]	**Basic Internal TSF Data Transfer Protection**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FPT_ITT.1.1[HW]	The TSF shall protect TSF data from *disclosure and modification* when it is transmitted between separate parts of the TOE.
Refinement:	The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE.

SFR FAU_SAS.1 is defined in the PP [21] and there is subject to two assignment operations. A third assignment operation is left open in the PP [21]. This operation assigns the type of persistent memory to which audit information is stored, and is filled in by this Security Target. In addition, the operation, which assigns the list of audit information, is further extended in this Security Target. Iteration [HW] is done here to prepare for other iterations that address any future major configurations of the TOE. This results in the following SFR:

FAU_SAS.1[HW]	**Audit Storage**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FAU_SAS.1.1[HW]	The TSF shall provide *the test process before TOE Delivery* with the capability to store *the Initialisation Data and/or Pre-personalisation Data* in the *NVM*.

SFR FDP_SDC.1 is defined in the PP [21] and is subject to one assignment operation. Iteration [HW] is done here to prepare for other iterations that address any future major configurations of the TOE. This results in the following SFR:

FDP_SDC.1[HW]	**Stored data confidentiality**
Hierarchical-To	No other components.
Dependencies	No dependencies.

FDP_SDC.1.1[HW]	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the *RAM and Non-Volatile Memory*.

SFR FDP_SDI.2 is defined in the PP [21] and is subject to two assignment operations. Iteration [HW] is done here to prepare for other iterations that address any future major configurations of the TOE. This results in the following SFR:

FDP_SDI.2[HW]	**Stored data integrity monitoring and action**
Hierarchical-To	FDP_SDI.1 Stored data integrity monitoring
Dependencies	No dependencies.
FDP_SDI.2.1[HW]	The TSF shall monitor user data stored in containers controlled by the TSF for *modification, deletion, repetition or loss of data* on all objects, based on the following attributes: *integrity check information associated with the data stored in memories*.
FDP_SDI.2.2[HW]	Upon detection of a data integrity error, the TSF shall *perform an error correction if possible and a Security Reset if not*.

For FCS_RNG.1.1 the PP [21] partially fills in the assignment for the security capabilities of the RNG by requiring a total failure test of the random source and adds an assignment operation for additional security capabilities of the RNG.
In addition, for FCS_RNG.1.2 the PP [21] partially fills in the assignment operation for the defined quality metric for the random numbers by replacing it by a selection and assignment operation.
For the above operations the original operations defined in chapter 5 of the PP [21] have been replaced by operations defined in chapter 3 of [1] and the open operations of the partially filled in operations in the statement of the security requirements in section 4.4 of [1] for better readability. Note that the selection operation for the RNG type has already been filled in by the PP [21]. Iteration [HW] is done here to prepare for other iterations that address any future major configurations of the TOE. This results in the following SFR:

FCS_RNG.1[HW]	**Random Number Generation (Class PTG.2)**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1[HW]	The TSF shall provide a *physical* random number generator that implements:
	(PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.
	(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG *prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.*
	(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.

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

(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered *at regular intervals or continuously*. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.

FCS_RNG.1.2[HW] The TSF shall provide *octets of bits* that meet:

(PTG.2.6) Test procedure A [1] does not distinguish the internal random numbers from output sequences of an ideal RNG.

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

Note: The definition of the Security Functional Requirement FCS_RNG.1 has been taken from [1].

Note: The functional requirement FCS_RNG.1[HW] is a refinement of FCS_RNG.1 defined in PP [21] according to [1].

Note: The Shannon entropy 0.997 per internal random bit compares to 7.976 per octet.

Note: Application Note 20 in [21] requires that the Security Target specifies for the security capabilities in FCS_RNG.1.1[HW] how the results of the total failure test of the random source are provided to the Security IC Embedded Software. The results of the internal test sequence are provided to the Security IC Embedded Software as a pass or fail criterion. The entropy of the random number is measured by the Shannon-Entropy as follows: $E = -\sum_{i=0}^{255} p_i \cdot \log_2 p_i$ where p_i is the probability that the byte $(b_7, b_6, \ldots, b_0)$ is equal to i as binary number. Here the term "bit" means measure of the Shannon-Entropy. The value "7.976" is assigned due to the requirements of "AIS31", [2].

By this, all assignment/selection operations are performed. This Security Target does not perform any other/further operations than stated in [1].
In addition to FCS_RNG.1[HW] the Crypto Library provides a deterministic random number generator:

FCS_RNG.1[DET]	**Random Number Generation (Deterministic)**
Hierarchical-To	No other components.
Dependencies	No dependencies

FCS_RNG.1.1[DET] The TSF shall provide a *deterministic* random number generator that implements:

(DRG.3.1) *If initialized with a random seed using a PTRNG of class PTG.2 (as defined in [2]) as random source, the internal state of the RNG shall have at least 230 bits (TDES) respectively 254 bits (AES) of entropy.*

(DRG.3.2) *The RNG provides forward secrecy (as defined in [2]).*

(DRG.3.3) *The RNG provides backward secrecy even if the current internal state is known (as defined in [2]).*

[1] Note: according par.295 in [2] the assignment may be empty.

FCS_RNG.1.2[DET] The TSF shall provide random numbers that meet:

(DRG.3.4) *The RNG, initialized with a random seed using a PTRNG of class PTG.2 (as defined in [2]) as random source, generates output for which in AES mode* 2^{48} *and in 3DES mode* 2^{35} *strings of bit length 128 are mutually different with probability at least* $1 - 2^{-24}$ *in AES mode and* $1 - 2^{-17}$ *in 3DES mode.*

(DRG.3.5) *Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A* [2] *(as defined in [2]).*

Note: The CryptoLib ROM Software provides the Security IC Embedded Software with seperate functionality to initialise the deterministic random number generator (which includes the chi-square test) and to generate pseudo-random data. It is the responsibility of the user to initialise the DRNG before generating random data. If it is tried to request pseudo-random numbers without having seeded the DRNG a security reset is triggered.

Note: Only if the chi-square test succeeds the hardware random number generator seeds the deterministic random number generator implemented as part of the CryptoLib ROM Software.

In addition to the requirements in the PP [21] this Security Target performs iterations of FMT_LIM.1 and FMT_LIM.2 to ensure that deploying Loader functionality after TOE Delivery does not weaken the TOE:

FMT_LIM.1[Loader]	**Limited Capabilities**
Hierarchical-To	No other components.
Dependencies	FMT_LIM.2 Limited availability.
FMT_LIM.1.1[Loader]	The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Deploying Loader functionality after *TOE Delivery* does not allow stored user data to be disclosed or manipulated by unauthorized user.

FMT_LIM.2[Loader]	**Limited Availability**
Hierarchical-To	No other components.
Dependencies	FMT_LIM.1 Limited capabilities.
FMT_LIM.2.1[Loader]	The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: The TSF prevents deploying the Loader functionality after *TOE Delivery*.

In compliance with Application Note 12 in the PP [21] the following section defines additional SFRs related to cryptographic functionality and access control functionality, which are required by this Security Target, but not by the PP [21].

[2] Note: according par.295 in [2] the assignment may be empty.

As required by Application Note 14 in the PP [21] the secure state is described in Section 7.2.1 in the rationale for SF.OPC.
Regarding Application Note 15 in the PP [21] generation of additional audit data is not defined for requirements FRU_FLT.2 and FPT_FLS.1.
As required by Application Note 19 in the PP [21] the automatic response of the TOE is described in Section 7.2.1 in the rationale for SF.PHY.

6.1.2 Additional SFRs regarding Cryptographic Support

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

FCS_COP.1[HW_DES]	**Cryptographic Operation (DES)**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.
FCS_COP.1.1[HW_DES]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Triple Data Encryption Algorithm (TDEA)* and cryptographic key sizes *of 112 or 168 bit* that meet the following *standards*: • *FIPS PUB 46-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION DATA ENCRYPTION STANDARD (DES) Reaffirmed 1999 October 25, keying options 1 and 2 [9]*
Note:	The cryptographic functionality FCS_COP.1[HW_DES] provided by the TOE achieves a security level of maximum 80 Bits, if keying option 2 is used.
Note:	The security functionality is resistant against side channel analysis and similar techniques. To fend off attackers with high attack potential a security level of at least 80 Bits must be used.

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

FCS_COP.1[HW_AES]	**Cryptographic Operation (AES)**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.
FCS_COP.1.1[HW_AES]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Advanced Encryption Standard (AES) algorithm* and a cryptographic key sizes *of 128 bit* that meet the following *standards*:

- *FIPS Publication 197, Advanced Encryption Standard (AES), NIST Special Publication 800-38A, 2001 [8]*

The (DES software as part of the Crypto Library of the) TOE shall meet the requirement ”Cryptographic operation (FCS_COP.1)” as specified below.

FCS_COP.1[SW_DES]	**Cryptographic Operation (DES & TDES)**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.
FCS_COP.1.1[SW_DES]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *DES and Triple-DES in one of the following modes of operation: ECB, CBC, CBC-MAC or CMAC* and cryptographic key sizes *1-key DES (56 bit), 2-key TDES (112 bit) or 3-key (168 bit)* that meet the following *standards*: • *FIPS Publication 46-3 (DES and TDES)* • *NIST Special Publication 800-38A Recommendation for Block Cipher Modes of Operation* • *NIST Special Publication 800-38C Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality – CBCMAC* • *NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*
Application Note:	The security functionality is resistant against side channel analysis and other attacks described in [10]. To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards). In particular this means that Single-DES shall not be used.

The (AES software as part of the Crypto Library of the) TOE shall meet the requirement ”Cryptographic operation (FCS_COP.1)” as specified below.

FCS_COP.1[SW_AES]	**Cryptographic Operation (AES)**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.
FCS_COP.1.1[SW_AES]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *AES in one of the following modes of operation: ECB, CBC, CBC-MAC or CMAC* and cryptographic key sizes *128 bit* that meet the following *standards*: • *NIST Special Publication 800-38A Recommendation for Block Cipher Modes of Operation* • *NIST Special Publication 800-38C Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality – CBCMAC*

- *NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*

Application Note: The security functionality is resistant against side channel analysis and other attacks described in [10].

6.1.3 Additional SFRs regarding Protection of TSF

The (HAL Software of the) TOE shall meet the requirement ”TSF Testing (FPT_TST.1)” as specified below.

FPT_TST.1	**TSF Testing**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of self tests *at the request of the authorised user to demonstrate the correct operation of:* • *the active shielding* • *the sensors*
FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of *Special Function Registers holding static values loaded during start-up*.
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

6.1.4 Additional SFRs regarding Security Management

The (HAL Software of the) TOE shall meet the requirement ”Specification of Management Functions (FMT_SMF.1)” as specified below.

FMT_SMF.1[SW]	**Specification of Management Functions (Software)**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FMT_SMF.1.1[SW]	The TSF shall be capable of performing the following management functions: • *Performing a System Reset* • *Terminating the IC* • *Getting the state of the Error Counter* • *Getting the state of the Delay Latch* • *Enabling the visibility of User Mode Special Function Registers in User Mode context.* • *Reading out the FabKey area*

Refinement: The System Reset and the Security Reset re-boot the IC. Once the error counter has reached a pre-defined value the IC is locked and cannot be reactivated. Terminating the IC means that the error counter is directly set to its termination value where the IC is locked.

6.1.5 Additional SFRs regarding User Data Protection

The (HAL Software of the) TOE shall meet the requirement "Subset Residual Information Protection (FSP_RIP.1)" as specified below.

FDP_RIP.1[HW]	**Subset Residual Information Protection**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FDP_RIP.1.1[HW]	The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from* the following objects: *all application dedicated temporary data when switching between applications*.
Refinement:	Whenever a switch occurs between Application 1 and Application 2, all application-dedicated temporary data such as user data and TSF data shall be deleted since these resources are shared. This includes RAM data as well as register contents.

In addition the following requirements are relevant for the CryptoLib:

FDP_RIP.1[SW]	**Subset Residual Information Protection**
Hierarchical-To	No other components.
Dependencies	No dependencies
FDP_RIP.1.1[SW]	The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation of the resource from the following objects: *all objects (variables)* used by the *Crypto Library* as specified in the user guidance documentation.

6.1.6 Additional SFRs regarding Access Control

The hardware shall provide different TOE modes to the Security IC Dedicated Support Software and Security IC Embedded Software. The TOE shall separate Security IC Dedicated Support Software and Security IC Embedded Software from each other by both, partitioning of memory and different TOE modes. The management of access to code and data as well as the configuration of the hardware shall be performed each in a dedicated TOE mode. The hardware shall enforce a separation between different applications (i.e. parts of the Security IC Embedded Software) running on the TOE. An application shall not be able to access hardware components without explicitly granted permission.

The Security Function Policy (SFP) **Hardware Access Control Policy** uses the definitions defined in the following sections. Thereby, subjects, objects and attributes are defined in a semi-formal tabular way. Each of them is equipped with a unique label shown in the second column of each table's header. Subjects and object are provided with a title and a descriptive block in addition. Operations can belong to objects (in that case contained in the first column) or to attributes (in that case contained in the second column).

6.1.6.1 Subjects

Subject	SSM_Code	Code run in Super System Mode
Info	Parts of the HAL Software and the Boot Software as part of the IC Dedicated Support Software and the Test Software as the IC Dedicated Test Software, executed as instructions by the CPU.	

Subject	SM_Code	Code run in System Mode
Info	Parts of the HAL Software, parts of the Crypto Library Software and parts of the Application Management Software as parts of the IC Dedicated Support Software, executed as instructions by the CPU.	

Subject	UM_Code	Code run in User Mode
Info	Parts of the HAL Software and parts of the Crypto Library Software as parts of the IC Dedicated Support Software and the Security IC Embedded Software, executed as instructions by the CPU.	

Subject	UM_SglApp_Code	Single Application Code run in User Mode
Info	The Security IC Embedded Software (Single Application Code), executed as instructions by the CPU.	

Subject	UM_App1_Code	Application 1 Code run in User Mode
Info	The Security IC Embedded Software (Application 1 Code), executed as instructions by the CPU.	

Subject	UM_App2_Code	Application 2 Code run in User Mode
Info	The Security IC Embedded Software (Application 2 Code), executed as instructions by the CPU.	

Notice, that UM_Code covers *all* user mode code in a generic sense. If one application is stored in NVM, then UM_SglApp_Code relates to all code in NVM. If two applications are stored in NVM then UM_App1_Code relates to Application 1 and UM_App2_Code to Application 2. [20] does not distinguish between 'Single Application' and 'Application 1'. This is only used here to make the definition of Subjects more clear.

6.1.6.2 Objects/Operations/Security Attributes related to Data in Memories

Object	SM_RAM_Seg	SM RAM Segment
Info	Located in the RAM memory, used exclusively for Super System Mode and System Mode stack and data.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	baseaddress	Configuration of base address via SFR_MemSegCfg.

Object	UM_RAM_Seg	UM RAM Segment
Info	Located in the RAM memory, used exclusivley for UM stack and data.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	DM9_RAM_Seg	DM9 RAM Segment
Info	Located in the RAM memory, used for efficiently accessing frequently used volatile data in User Mode. Must be a subset of the UM_RAM_Seg Segment. Only the intersection between DM9_RAM_Seg and UM_RAM_Seg will be accessible.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	baseaddress	Can be relocated physically via SFR_MemSegCfg.

Object	Key_RAM_Seg	Key RAM Segment
Info	Located in the RAM memory, used for key management.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	EE_UserData_Seg	EEPROM User Data Segment
Info	Located in the EEPROM memory, intended for User Mode non-volatile data storage.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.
Attribute	baseaddress	Configuration of base address via SFR_MemSegCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	FL_UserData_Seg	FLASH User Data Segment
Info	Located in the FLASH memory, intended for User Mode non-volatile data storage.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.
Attribute	baseaddress	Configuration of base address via SFR_MemSegCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	NXP_ConfigData_Seg	NXP Configuration Data Segment
Info	Located in the EEPROM memory and has a fixed size. Stores low level configuration.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.

Object	ROM_Mirror_Seg	ROM Mirror Segment
Info	Located in the ROM memory and its size depends on the physical module size. Can be used for signature generation.	
Operation	read	Read data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.

Object	EE_Mirror_Seg	EEPROM Mirror Segment
Info	Located in the EEPROM memory and its size depends on the physical module size. Can be used for signature generation and is for test purposes only.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.

Object	FL_Mirror_Seg	FLASH Mirror Segment
Info	Located in the FLASH memory and its size depends on the physical module size. Can be used for signature generation and is for test purposes only.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.

Object	SM_ROMConst_Seg	SM ROM Constant Segment
Info	Located in the ROM memory, stores constants for System Mode and Super System Mode.	
Operation	read	Read data.

Object	SM_ROMConst_Seg	SM ROM Constant Segment
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	UM_FLConst_Seg	UM FLASH Constant Segment
Info	Located in the FLASH memory, stores constants for User Mode.	
Operation	read	Read data.
Operation	write	Write data.
Attribute	size	Configuration of size via SFR_MemSegCfg.
Attribute	enable	Enable r/w access via SFR_MemSegCfg.

Object	UM_FLApp1Const_Seg	UM FLASH Application 1 Constant Segment
Info	Located in the FLASH memory, stores constants for User Mode (Application 1).	
Operation	read	Read data.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	UM_FLApp2Const_Seg	UM FLASH Application 2 Constant Segment
Info	Located in the FLASH memory, stores constants for User Mode (Application 2).	
Operation	read	Read data.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	SharedConst_Seg	Shared Constant Segment
Info	Located in the ROM memory, stores constants for code that shall be executed in all TOE modes.	
Operation	read	Read data.
Attribute	size	Configuration of size via SFR_MemSegCfg.

6.1.6.3 Objects/Operations/Security Attributes related to Code in Memories

Object	XCall_Table_Seg	Sys Call/User Call/ISR Table Segment
Info	Located in the ROM memory and has a fixed size. Contains the entry points for system call, user calls and interrupt service handler.	
Info	Located in the FLASH memory and has a fixed size. Contains the entry points for system call, user calls and interrupt service handler.	
Operation	execute	Execute code.

Object	BootTestCode_Seg	Boot/Test Code Segment
Info	Located in the ROM memory and has a fixed size (Mask Coded Bits). Contains the Boot ROM Software and the Test ROM Software of the TOE.	
Operation	execute	Execute code.
Operation	read	Read data.
Attribute	enable	Enable read and execute access via SFR_AccCtrlCfg.

Object	SM_Code_Seg	SM Code Segment
Info	Located in the ROM memory. Contains the code of the TOE that runs with System Mode priviledge.	
Operation	read	Read code.
Operation	execute	Execute code.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	UM_Code_Seg	UM Code Segment
Info	Located in the FLASH memory. Contains the code of the TOE that runs with User Mode priviledge.	
Operation	execute	Execute code.
Operation	read	Read data.
Attribute	baseaddress	Configuration of base address via SFR_MemSegCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	UM_App1Code_Seg	UM Application 1 Code Segment
Info	Located in the FLASH memory. Contains the Application 1 code of the TOE that runs with User Mode priviledge.	
Operation	execute	Execute code.
Operation	read	Read data.
Attribute	baseaddress	Configuration of base address via SFR_MemSegCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	UM_App2Code_Seg	UM Application 2 Code Segment
Info	Located in the FLASH memory. Contains the Application 2 code of the TOE that runs with User Mode priviledge.	
Operation	execute	Execute code.
Operation	read	Read data.
Attribute	baseaddress	Configuration of base address via SFR_MemSegCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	SharedCode_Seg	Shared Code Segment
Info	Located in the ROM memory. Contains the code of the TOE that shall be visible in all TOE modes.	
Operation	execute	Execute code.
Operation	read	Read data.
Attribute	baseaddress	Configuration of base address via SFR_MemSegCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	SM_PatchCode_Seg	SM Patch Code Segment
Info	Located in the EEPROM memory. Contains the patch code of the TOE that is intended to replace or extend any Super System Mode or System Mode code in the ROM memory.	
Operation	execute	Execute code.
Operation	read	Read data.
Operation	write	Write data.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.
Attribute	size	Configuration of size via SFR_MemSegCfg.

6.1.6.4 Objects/Operations/Security Attributes related to Special Function Registers

Object	SFR_SysMgmt	Special Function Registers related to System Management
Info	Group of Special Function Registers related to system management.	
Operation	access	General access to this Special Function Register Group.
Operation	read	Read a configuration setting.
Operation	write	Write a configuration setting.

Object	SFR_MemSegCfg	Special Function Registers related to Memory Segment Configuration
Info	Group of Special Function Registers to configure the base address and size of data and code segments located in ROM, RAM, EEPROM and FLASH.	
Operation	access	General access to this Special Function Register Group.
Operation	read	Read base address or size.
Operation	write	Write a base address or size.

Object	SFR_AccCtrlCfg	Special Function Registers related to its Access Control
Info	Group of Special Function Registers to configure the access to data and code segments located in ROM, RAM, EEPROM and FLASH as well as access Special Function Registers.	
Operation	access	General access to this Special Function Register Group.

Object	SFR_AccCtrlCfg	Special Function Registers related to its Access Control
Operation	read	Read a configuration setting / value.
Operation	write	Write a configuration setting / value.

Object	SFR_Testing	Special Function Registers related to Testing
Info	Group of Special Function Registers reserved for testing purposes.	
Operation	access	General access to this Special Function Register Group.
Operation	read	Read a configuration setting / value.
Operation	write	Write a configuration setting / value.

Object	SFR_HWComp	Special Function Registers related to Hardware Components
Info	Group of Special Function Registers used to utilize the following hardware components: • AES Coprocessor • DES Coprocessor • CRC Coprocessor • Physical Random Number Generator	
Operation	access	General access to this Special Function Register Group.
Operation	read	Read a configuration setting / value.
Operation	write	Write a configuration setting / value.

6.1.6.5 Access Rules

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

FDP_ACC.1[MEM]	**Subset Access Control (Memories)**
Hierarchical-To	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* on *all code running on the TOE, all memories and all memory operations*.
Application Note:	The Access Control Policy shall be enforced by implementing a Memory Management Unit, which maps virtual addresses to physical addresses. The CPU always uses virtual addresses, which are mapped to physical addresses by the Memory Management Unit. Prior to accessing the respective memory address, the Memory Management Unit checks if the access is allowed. A denied read or write access or read/write to a non-existing memory address is treated as a security violation and will trigger a Security Reset.

FDP_ACC.1[SFR]	**Subset Access Control (Special Function Registers)**
Hierarchical-To	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1[SFR]	The TSF shall enforce the *Hardware Access Control Policy* on *all code running on the TOE, all Special Function Registers and all Special Function Register operations.*
Application Note:	The Hardware Access Control Policy shall be enforced by implementing hardware access control to each Special Function Register. For every access the TOE mode is used to determine if the access shall be granted or denied. A denied read or write access or read/write to a non-existing Special Function Registers is treated as a security violation and will trigger a Security Reset.

The following access control rules are defined in a semi-formal way, i.e. each rule is provided with a unique label and each rule exactly identifies the subject (via its label defined in section 6.1.6.1), object (via its label defined in the sections 6.1.6.2, 6.1.6.3 and 6.1.6.4, respectively) and operation (added to the associated operation via "."). For operations with explicit authorized access, the related attribute is referenced (an shown via a hyperlink to the unique label of the attribute associated to the operation via ".").

FDP_ACF.1[MEM]	**Security Attribute Based Access Control (Memories)**
Hierarchical-To	No other components.
Dependencies	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
FDP_ACF.1.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* to objects based on the following: *all subjects and objects and the attributes themselves defined as the objects SFR_SysMgmt and SFR_MemSegCfg.*
FDP_ACF.1.2[MEM]	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

ACF12.MEM:SSM:ROM2	*The SSM_Code is allowed to perform SM_ROMConst_Seg.read.*
ACF12.MEM:SSM:ROM4	*The SSM_Code is allowed to perform SharedConst_Seg.read.*
ACF12.MEM:SSM:ROM5	*The SSM_Code is allowed to perform XCall_Table_Seg.execute.*
ACF12.MEM:SSM:ROM7	*The SSM_Code is allowed to perform SM_Code_Seg.execute.*
ACF12.MEM:SSM:ROM9	*The SSM_Code is allowed to perform SharedCode_Seg.execute.*
ACF12.MEM:SSM:FL1	*The SSM_Code is allowed to perform UM_FLConst_Seg.read.*
ACF12.MEM:SSM:RAM1	*The SSM_Code is allowed to perform SM_RAM_Seg.read and SM_RAM_Seg.write.*
ACF12.MEM:SSM:RAM2	*The SSM_Code is allowed to perform UM_RAM_Seg.read and UM_RAM_Seg.write.*
ACF12.MEM:SM:ROM2	*The SM_Code is allowed to perform SM_ROMConst_Seg.read.*
ACF12.MEM:SM:ROM4	*The SM_Code is allowed to perform SharedConst_Seg.read.*
ACF12.MEM:SM:ROM5	*The SM_Code is allowed to perform XCall_Table_Seg.execute.*
ACF12.MEM:SM:ROM7	*The SM_Code is allowed to perform SM_Code_Seg.execute.*

ACF12.MEM:SM:ROM9 *The SM_Code is allowed to perform SharedCode_Seg.execute.*

ACF12.MEM:SM:FL1 *The SM_Code is allowed to perform UM_FLConst_Seg.read.*

ACF12.MEM:SM:RAM1 *The SM_Code is allowed to perform SM_RAM_Seg.read and SM_RAM_Seg.write.*

ACF12.MEM:SM:RAM2 *The SM_Code is allowed to perform UM_RAM_Seg.read and UM_RAM_Seg.write.*

ACF12.MEM:UM:ROM4 *The UM_Code is allowed to perform SharedConst_Seg.read.*

ACF12.MEM:UM:ROM9 *The UM_Code is allowed to perform SharedCode_Seg.execute.*

ACF12.MEM:UM:FL1 *The UM_SglApp_Code is allowed to perform UM_FLConst_Seg.read.*

ACF12.MEM:UM:FL1a *The UM_App1_Code is allowed to perform UM_FLApp1Const_Seg.read.*

ACF12.MEM:UM:FL1b *The UM_App2_Code is allowed to perform UM_FLApp2Const_Seg.read.*

ACF12.MEM:UM:FL3 *The UM_SglApp_Code is allowed to perform UM_Code_Seg.execute.*

ACF12.MEM:UM:FL3a *The UM_App1_Code is allowed to perform UM_App1Code_Seg.execute.*

ACF12.MEM:UM:FL3b *The UM_App2_Code is allowed to perform UM_App2Code_Seg.execute.*

ACF12.MEM:UM:RAM2 *The UM_Code is allowed to perform UM_RAM_Seg.read and UM_RAM_Seg.write.*

ACF12.MEM:UM:RAM3 *The UM_Code is allowed to perform DM9_RAM_Seg.read and DM9_RAM_Seg.write.*

FDP_ACF.1.3[MEM] The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

ACF13.MEM:SSM:ROM1 *The SSM_Code is allowed to perform ROM_Mirror_Seg.read if the attribute ROM_Mirror_Seg.enable grants this right.*

ACF13.MEM:SSM:ROM1a *The SSM_Code is allowed to perform BootTestCode_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*

ACF13.MEM:SSM:ROM1b *The SSM_Code is allowed to perform SM_Code_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*

ACF13.MEM:SSM:ROM1d *The SSM_Code is allowed to perform SharedCode_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*

ACF13.MEM:SSM:ROM6 *The SSM_Code is allowed to perform BootTestCode_Seg.execute if the attribute BootTestCode_Seg.enable grants this right.*

ACF13.MEM:SSM:EE1 *The SSM_Code is allowed to perform EE_UserData_Seg.read and EE_UserData_Seg.write if the attribute EE_UserData_Seg.enable grants this right.*

ACF13.MEM:SSM:EE2 *The SSM_Code is allowed to perform NXP_ConfigData_Seg.read and NXP_ConfigData_Seg.write if the attribute NXP_ConfigData_Seg.enable grants this right.*

ACF13.MEM:SSM:EE4 *The SSM_Code is allowed to perform SM_PatchCode_Seg.read and SM_PatchCode_Seg.write if the attribute SM_PatchCode_Seg.enable grants this right.*

ACF13.MEM:SSM:EE4 *The SSM_Code is allowed to perform SM_PatchCode_Seg.execute if the attribute SM_PatchCode_Seg.enable grants this right.*

ACF13.MEM:SSM:EE6 *The SSM_Code is allowed to perform EE_Mirror_Seg.read and EE_Mirror_Seg.write if the attribute EE_Mirror_Seg.enable grants this right.*

ACF13.MEM:SSM:FL1 *The SSM_Code is allowed to perform UM_FLConst_Seg.write if the attribute UM_FLConst_Seg.enable grants this right.*

ACF13.MEM:SSM:FL2 *The SSM_Code is allowed to perform FL_UserData_Seg.read and FL_UserData_Seg.write if the attribute FL_UserData_Seg.enable grants this right.*

ACF13.MEM:SSM:FL4 *The SSM_Code is allowed to perform FL_Mirror_Seg.read and FL_Mirror_Seg.write if the attribute FL_Mirror_Seg.enable grants this right.*

ACF13.MEM:SSM:RAM4 *The SSM_Code is allowed to perform Key_RAM_Seg.read and Key_RAM_Seg.write if the attribute Key_RAM_Seg.enable grants this right.*

ACF13.MEM:SM:ROM1 *The SM_Code is allowed to perform ROM_Mirror_Seg.read if the attribute ROM_Mirror_Seg.enable grants this right.*

ACF13.MEM:SM:ROM1b *The SM_Code is allowed to perform SM_Code_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*

ACF13.MEM:SM:ROM1d *The SM_Code is allowed to perform SharedCode_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*

ACF13.MEM:SM:EE1 *The SM_Code is allowed to perform EE_UserData_Seg.read and EE_UserData_Seg.write if the attribute EE_UserData_Seg.enable grants this right.*

ACF13.MEM:SM:EE2 *The SM_Code is allowed to perform NXP_ConfigData_Seg.read and NXP_ConfigData_Seg.write if the attribute NXP_ConfigData_Seg.enable grants this right.*

ACF13.MEM:SM:EE4 *The SM_Code is allowed to perform SM_PatchCode_Seg.read and SM_PatchCode_Seg.write if the attribute SM_PatchCode_Seg.enable grants this right.*

ACF13.MEM:SM:EE4 *The SM_Code is allowed to perform SM_PatchCode_Seg.execute if the attribute SM_PatchCode_Seg.enable grants this right.*

ACF13.MEM:SM:FL1 *The SM_Code is allowed to perform UM_FLConst_Seg.write if the attribute UM_FLConst_Seg.enable grants this right.*

ACF13.MEM:SM:FL2 *The SM_Code is allowed to perform FL_UserData_Seg.read and FL_UserData_Seg.write if the attribute FL_UserData_Seg.enable grants this right.*

ACF13.MEM:SM:FL4 *The SM_Code is allowed to perform FL_Mirror_Seg.read and FL_Mirror_Seg.write if the attribute FL_Mirror_Seg.enable grants this right.*

ACF13.MEM:SM:RAM4 *The SM_Code is allowed to perform Key_RAM_Seg.read and Key_RAM_Seg.write if the attribute Key_RAM_Seg.enable grants this right.*

ACF13.MEM:UM:EE4 *The UM_Code is allowed to perform SM_PatchCode_Seg.execute if the attribute SM_PatchCode_Seg.enable grants this right.*

ACF13.MEM:UM.SglApp:EE1 *The UM_SglApp_Code is allowed to perform EE_UserData_Seg.read if the attribute EE_UserData_Seg.enable grants this right.*

ACF13.MEM:UM.App1:EE1 *The UM_App1_Code is allowed to perform EE_UserData_Seg.read if the attribute EE_UserData_Seg.enable grants this right.*

ACF13.MEM:UM.SglApp:FL2 *The UM_SglApp_Code is allowed to perform FL_UserData_Seg.read if the attribute FL_UserData_Seg.enable grants this right.*

ACF13.MEM:UM.App2:FL2 *The UM_App2_Code is allowed to perform FL_UserData_Seg.read if the attribute FL_UserData_Seg.enable grants this right.*

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

FDP_ACF.1[SFR] Security Attribute Based Access Control (Special Function Registers)

Hierarchical-To No other components.

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

FDP_ACF.1.1[SFR] The TSF shall enforce the *Hardware Access Control Policy* to objects based on the following: *all subjects and objects and the attributes itself defined as the object SFR_AccCtrlCfg*.

FDP_ACF.1.2[SFR] The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

ACF12.SFR:SSM:GRP1 *The SSM_Code is allowed to perform SFR_SysMgmt.access.*

ACF12.SFR:SSM:GRP2 *The SSM_Code is allowed to perform SFR_MemSegCfg.access.*

ACF12.SFR:SSM:GRP3 *The SSM_Code is allowed to perform SFR_Testing.access.*

ACF12.SFR:SSM:GRP4 *The SSM_Code is allowed to perform SFR_HWComp.access.*

ACF12.SFR:SM:GRP1 *The SM_Code is allowed to perform SFR_SysMgmt.access.*

ACF12.SFR:SM:GRP2 *The SM_Code is allowed to perform SFR_MemSegCfg.access.*

ACF12.SFR:SM:GRP4 *The SM_Code is allowed to perform SFR_HWComp.access.*

ACF12.SFR:UM:GRP4 *The UM_Code is allowed to perform SFR_HWComp.access.*

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

ACF13.SFR:UM:GRP1 *The UM_Code is allowed to perform SFR_SysMgmt.read and SFR_SysMgmt.write to SFR.MMU_UM_EVAL, SFR.MMU_WD_CNTR and SFR.MMU_DM9BASE.*

FDP_ACF.1.4[SFR] The TSF shall explicitly deny access of subjects to objects based on the rules: *Access to Special Function Registers in User Mode.*

ACF14.SFR:SSM:GRP4.1 *The SSM_Code is not allowed to perform SFR_HWComp.read for Special Function Registers used as AES/DES key registers.*

ACF14.SFR:SM:GRP4.1 *The SM_Code is not allowed to perform SFR_HWComp.read for Special Function Registers used as AES/DES key registers.*

ACF14.SFR:UM:GRP4.1 *The UM_Code is not allowed to access any Special Function Register related to User Mode if SFR_AccCtrlCfg denies this right.*

ACF14.SFR:UM:GRP4.2 *The UM_Code is not allowed to perform SFR_HWComp.read for Special Function Registers SFR.CRC_DATAH, SFR.CRC_DATAL.*

ACF14.SFR:UM:GRP4.3 *The UM_Code is not allowed to perform SFR_HWComp.read for Special Function Registers used as AES/DES key registers.*

6.1.6.6 Implications of the Hardware Access Control Policy

The Access Control Policy has some implications, that can be drawn from the policy and that are essential parts of the TOE security functionality.

- Code executed in Super System Mode is quite powerful and used to configure and test the TOE.
- Code executed in the System Mode can administrate the configuration of Memory Management Unit, because it has access to the respective Special Function Registers.
- Code executed in the User Mode hardly administrate the configuration of the TOE, because it has very limited access to the related Special Function Registers.

FMT_MSA.3[MEM]	**Static Attribute Initialization (Memories)**
Hierarchical-To	No other components.
Dependencies	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[MEM]	The TSF shall allow *no subject* to specify alternative initial values to override the default values when an object or information is created.
Application Note:	Restrictive means here that the reset values of the Special Function Registers related to SFR_MemSegCfg are set to zero, which effectively disables all related MMU rules. The TOE does not provide objects or information that can be created, since it provides access to memory areas. The definition of objects that are stored in the TOE's memory is subject to the Security IC Embedded Software.

FMT_MSA.3[SFR]	**Static Attribute Initialization (Special Function Registers)**
Hierarchical-To	No other components.
Dependencies	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1[SFR]	The TSF shall enforce the *Hardware Access Control Policy* to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[SFR]	The TSF shall allow *no subject* to specify alternative initial values to override the default values when an object or information is created.

Application Note:	The TOE does not provide objects or information that can be created, since no further security attributes can be derived (i.e. the set of Special Function Registers that contain security attributes is fixed). The definition of objects that are stored in the TOE's memory is subject to the Security IC Embedded Software.
FMT_MSA.1[MEM]	**Management of Security Attributes (Memories)**
Hierarchical-To	No other components.
Dependencies	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* to restrict the ability to *modify* the security attributes *defined as the object SFR_MemSegCfg* to *code executed in System Mode or Super System Mode.*
Application Note:	This component does not include any management functionality for the configuration of the memory partition. This is because the memory partition is fixed and cannot be changed after TOE delivery.
FMT_MSA.1[SFR]	**Management of Security Attributes (Special Function Registers)**
Hierarchical-To	No other components.
Dependencies	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1[SFR]	The TSF shall enforce the *Hardware Access Control Policy* to restrict the ability to *modify* the security attributes *defined in the Special Function Registers* to *code executed in a TOE mode which has write access to the respective Special Function Registers.*
FMT_SMF.1[HW]	**Specification of Management Functions (Hardware)**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FMT_SMF.1.1[HW]	The TSF shall be capable of performing the following management functions: • *Change of TOE mode to User Mode by calling one of the following instructions: USR or EUSR* • *Change of TOE mode to System Mode by calling one of the following instructions: SYS or ESYS* • *Change of TOE mode to Super System Mode by calling SYS0* • *Change of TOE mode by invoking an interrupt* • *Change of TOE mode by finishing an interrupt (with instruction RETI)*
Application Note:	The iteration of FMT_MSA.1 with the dependency to FMT_SMF.1 may imply a separation of the Specification of Management Functions. However, iteration of FMT_SMF.1 is not needed for hardware access control (FMT_MSA.1[MEM] and FMT_MSA.1[SFR]) because all management functions rely on the same features implemented in the hardware.

6.2 Security Assurance Requirements

Table 6.38 below lists all security assurance components that are valid for this Security Target. With one exception these security assurance components are required by EAL5 (see section 2.2) or by the Protection Profile.
The exception is the component ASE_TSS.2 which is chosen as an augmentation in this ST to give architectural information on the security functionality of the TOE.
The refinements of the PP [21], that must be adapted for EAL5, are described in section 6.2.1.

Name	Title
ADV_ARC.1	Security architecture description
ADV_FSP.5	Complete semi-formal functional specification with additional error information
ADV_IMP.1	Implementation representation of the TSF
ADV_INT.2	Well-structured internals
ADV_TDS.4	Semiformal modular design
AGD_OPE.1	Operational user guidance
AGD_PRE.1	Preparative procedures
ALC_CMC.4	Production support, acceptance procedures and automation
ALC_CMS.5	Development tools CM coverage
ALC_DEL.1	Delivery procedures
ALC_DVS.2	Sufficiency of security measures
ALC_LCD.1	Developer defined life-cycle model
ALC_TAT.2	Compliance with implementation standards
ASE_INT.1	ST introduction
ASE_CCL.1	Conformance claims
ASE_SPD.1	Security problem definition
ASE_OBJ.2	Security objectives
ASE_ECD.1	Extended components definition
ASE_REQ.2	Derived security requirements
ASE_TSS.2	TOE summary specification with architectural design summary
ATE_COV.2	Analysis of coverage
ATE_DPT.3	Testing: modular design
ATE_FUN.1	Functional testing
ATE_IND.2	Independent testing - sample
AVA_VAN.5	Advanced methodical vulnerability analysis

Tab. 6.38: Security Assurance Requirements

6.2.1 Refinements of the TOE Security Assurance Requirements

The Security Target claims conformance to the PP [21] and therefore it has to be conform to the refinements of the TOE security assurance requirements (see Application Note 22 in [21]). Because the refinements in the PP [21] are defined for the security assurance components of EAL4, some refinements have to be applied to assurance components of the higher level EAL5 stated in the Security Target.
Table 6.39 lists the influences of the refinements of the PP [21] on the Security Target. Most of the refined security assurance components have the same level in both documents (PP [21] and Security Target). The following two subsections apply the refinements to ALC_CMS.5 and ADV_FSP.5, which are different between the PP [21] and the Security Target.

SAR	Note
ALC_DEL	Same as in PP, refinement valid without change
ALC_DVS	Same as in PP, refinement valid without change
ALC_CMS	ALC_CMS.5, refinements valid without change
ALC_CMC	Same as in PP, refinement valid without change
ADV_ARC	Same as in PP, refinement valid without change
ADV_FSP	ADV_FSP.5, refinements have to be adapted
ADV_IMP	Same as in PP, refinement valid without change
ATE_COV	Same as in PP, refinement valid without change
AGD_OPE	Same as in PP, refinement valid without change
AGD_PRE	Same as in PP, refinement valid without change
AVA_VAN	Same as in PP, refinement valid without change[3]

Tab. 6.39: Security Assurance Requirements

6.2.1.1 Refinements regarding CM scope (ALC_CMS)

This Security Target requires a higher evaluation level for the CC family ALC_CMS, namely ALC_CMS.5 instead of ALC_CMS.4. The refinement of the PP [21] regarding ALC_CMS.4 is a clarification of the configuration item ”TOE implementation representation”. Since in ALC_CMS.5, the content and presentation of evidence element ALC_CMS.5.1C only adds a further configuration item to the list of items to be tracked by the CM system, the refinement can be applied without changes.
The refinement of the configuration item ”TOE implementation representation” of ALC_CMS.4 can be found in section 6.2.1.3 of [21] and is not cited here.

6.2.1.2 Refinements regarding CM scope (ADV_FSP)

This Security Target requires a higher evaluation level for the CC family ADV_FSP, namely ADV_FSP.5 instead of ADV_FSP.4. The refinement of the PP [21] regarding ADV_FSP.4 is concerned with the complete representation of

[3] According to the Application Note 30 in [21] the Security Target should indicate the version of the document Supporting Document Mandatory Technical Document Application of Attack Potential to Smart Cards [22] used for the vulnerability analysis.

the TSF, the purpose and method of use of all TSFI, and the accuracy and completeness of the SFR instantiations. The refinement is not a change in the wording of the action elements, but a more detailed definition of the above items.
The higher level ADV_FSP.5 requires a Functional Specification in a ”semi-formal style” (ADV_FSP.5.2C).
The component ADV_FSP.5 enlarges the scope of the error messages to be described from those resulting from an invocation of a TSFI (ADV_FSP.5.6C) to also those not resulting from an invocation of a TSFI (ADV_FSP.5.7C). For the latter a rationale shall be provided (ADV_FSP.5.8C).
Since the higher level ADV_FSP.5 only affects the style of description and the scope of and rationale for error messages, the refinements can be applied without changes and are valid for ADV_FSP.5. The refinement of the original component ADV_FSP.4 can be found in section 6.2.1.6 of the Protection Profile [21] and is not cited here.

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

Section 6.3.1 in [21] provides a rationale for the mapping between security functional requirements and security objectives defined in the PP [21]. The mapping is reproduced in the following table. Notice, that only TOE objectives are listed since no SFRs are mapped to objectives related to operational respectively development environment.

SO	SFR
O.Leak-Inherent	FDP_IFC.1 FDP_ITT.1[HW] FPT_ITT.1[HW]
O.Phys-Probing	FDP_SDC.1[HW] FPT_PHP.3
O.Malfunction	FPT_FLS.1 FRU_FLT.2
O.Phys-Manipulation	FDP_SDI.2[HW] FPT_PHP.3
O.Leak-Forced	FDP_IFC.1 FDP_ITT.1[HW] FPT_FLS.1 FPT_ITT.1[HW] FPT_PHP.3 FRU_FLT.2
O.Abuse-Func	FDP_IFC.1 FDP_ITT.1[HW] FMT_LIM.1[HW] FMT_LIM.2[HW] FPT_FLS.1

SO	SFR
	FPT_ITT.1[HW] FPT_PHP.3 FRU_FLT.2
O.Identification	FAU_SAS.1[HW]
O.RND	FCS_RNG.1[DET] FCS_RNG.1[HW] FDP_IFC.1 FDP_ITT.1[HW] FPT_FLS.1 FPT_ITT.1[HW] FPT_PHP.3 FRU_FLT.2
O.Cap_Avail_Loader	FMT_LIM.1[Loader] FMT_LIM.2[Loader]

Tab. 6.40: Security Functional Requirements vs. Security Objectives (PP)

The Security Target additionally defines the SFRs for the TOE that are listed in Tables 6.41 and 6.42. In addition Security Requirements for the Environment are defined. The following table gives an overview, how the requirements are combined to meet the security objectives.

SO	SFR
O.INTEGRITY_CHK	FDP_ITT.1[HW] FPT_ITT.1[HW]
O.NVM_INTEGRITY	FDP_SDI.2[HW]
O.MEM_ACCESS	FDP_ACC.1[MEM] FDP_ACF.1[MEM] FMT_MSA.1[MEM] FMT_MSA.3[MEM] FMT_SMF.1[HW]
O.SFR_ACCESS	FDP_ACC.1[SFR] FDP_ACF.1[SFR] FMT_MSA.1[SFR] FMT_MSA.3[SFR] FMT_SMF.1[HW]
O.HW_REUSE	FDP_RIP.1[HW]
O.Self-Test	FPT_TST.1
O.Reset	FMT_SMF.1[SW]

Tab. 6.41: Security Functional Requirements vs. Security Objectives (ST Part1)

SO	SFR
O.AES	FCS_COP.1[HW_AES] FCS_COP.1[SW_AES]
O.DES	FCS_COP.1[HW_DES] FCS_COP.1[SW_DES]
O.REUSE	FDP_RIP.1[SW]

Tab. 6.42: Security Functional Requirements vs. Security Objectives (ST Part2)

The rationale for all items defined in the Security Target is given below.

Justification related to O.INTEGRITY_CHK:

SFR	Rationale
FDP_ITT.1[HW]	This SFR requires the TOE to check the integrity of User Data and TSF data when transferred between different parts of the TOE as required by the objective.
FPT_ITT.1[HW]	This SFR requires the TOE to check the integrity of User Data and TSF data when transferred between different parts of the TOE as required by the objective.

Justification related to O.NVM_INTEGRITY:

SFR	Rationale
FDP_SDI.2[HW]	This SFR requires a control function, that adjusts the conditions of a NVM block so that integrity of the data read from it can be ensured by the TOE.

Justification related to O.MEM_ACCESS:

SFR	Rationale
FDP_ACC.1[MEM]	This SFR with the related SFP ”Hardware Access Control Policy” exactly requires to implement a memory partition as demanded by the objective.
FDP_ACF.1[MEM]	This SFR with the related SFP ”Hardware Access Control Policy” defines the rules to implement the memory partition as demanded by the objective.
FMT_MSA.3[MEM]	This SFR requires that the TOE provides default values for the security attributes. Since the TOE is a hardware platform these default values are generated by the reset procedure for the related Special Function Register. They are needed by the TOE to provide a default configuration after reset. Therefore this SFR meets the objective.

SFR	Rationale
FMT_MSA.1[MEM]	This SFR requires that 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. Therefore this SFR meets the objective.
FMT_SMF.1[HW]	This SFR is used for the specification of the management functions to be provided by the TOE as demanded by the objective.

Justification related to O.SFR_ACCESS:

SFR	Rationale
FDP_ACC.1[SFR]	This SFR with the related SFP ”Hardware Access Control Policy” requires to implement access control for Special Function Register as demanded by this objective.
FDP_ACF.1[SFR]	This SFR with the related SFP ”Hardware Access Control Policy” exactly require certain security attributes to implement the access control to Special Function Register as demanded by this objective.
FMT_MSA.3[SFR]	This SFR requires that the TOE provides default values for the Special Function Register (values as well as access control). The default values are needed to ensure a defined setup for the operation of the TOE. There this SFR meets the objective.
FMT_MSA.1[SFR]	This SFR is realized in a way that – besides the definition of access rights to Special Function Registers related to hardware components in User Mode – no management of the security attributes is possible because the attributes are implemented in the hardware and cannot be changed. Thefore this SFR meets the objective.
FMT_SMF.1[HW]	This SFR is used for the specification of the management functions to be provided by the TOE as demanded by this objective.

Justification related to O.HW_REUSE:

SFR	Rationale
FDP_RIP.1[HW]	This SFR requires the TOE to provide procedural measures to prevent disclosure of memory contents that was used by the TOE. This applies to the code segments run in System Mode which requires the IC Dedicated Support Software to make unavailable all memory contents that has been used by it. Therefore this SFR meets the objective.

Justification related to O.Self-Test:

SFR	Rationale
FPT_TST.1	This SFR requires self-testing of the TOE to authorized users as required by the objective.

Justification related to O.Reset:

SFR	Rationale
FMT_SMF.1[SW]	This SFR requires to provide management functions allowing to reset the TOE as required by the objective.

Justification related to O.AES:

SFR	Rationale
FCS_COP.1[SW_AES]	This SFR requires the TOE to support AES encryption and decryption as demanded by the objective. FCS_COP.1[SW_AES] requires the modes of operation on top of FCS_COP.1[HW_AES].
FCS_COP.1[HW_AES]	This objective requires the TOE to support AES encryption and decryption. FCS_COP.1[HW_AES] requires the AES according to the standard.

Justification related to O.DES:

SFR	Rationale
FCS_COP.1[SW_DES]	This SFR requires the TOE to support Triple-DES encryption and decryption as demanded by the objective. FCS_COP.1[SW_DES] requires the modes of operation on top of FCS_COP.1[HW_DES].
FCS_COP.1[HW_DES]	This objective requires the TOE to support Tripe-DES encryption and decryption. FCS_COP.1[HW_DES] requires the Triple-DES according to the standard.

Justification related to O.REUSE:

SFR	Rationale
FDP_RIP.1[SW]	O.REUSE requires the TOE to provide procedural measures to prevent disclosure of memory contents that was used by the TOE. This applies to the code segments run in System Mode and is met by the SFR FDP_RIP.1[SW], which requires the CryptoLib to make unavailable all memory contents that has been used by it.

6.3.2 Dependencies of Security Functional Requirements

The dependencies listed in the PP [21] are independent of the additional dependencies listed in the table below. The dependencies of the PP [21] are fulfilled within the PP [21] and at least one dependency is considered to be satisfied.

The following discussion demonstrates how the dependencies defined by Part 2 of the Common Criteria for the requirements specified in sections 6.1 and 6.2 are satisfied.

The dependencies defined in the Common Criteria are listed in the table below:

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FAU_SAS.1[HW]	No dependencies.	No dependency
FCS_RNG.1[HW]	No dependencies.	No dependency
FDP_ITT.1[HW]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes
FDP_IFC.1	FDP_IFF.1 Simple security attributes	See discussion in the PP
FDP_SDC.1[HW]	No dependencies.	No dependency
FDP_SDI.2[HW]	No dependencies.	No dependency
FMT_LIM.1[HW]	FMT_LIM.2 Limited availability.	Yes
FMT_LIM.1[Loader]	FMT_LIM.2 Limited availability.	Yes
FMT_LIM.2[HW]	FMT_LIM.1 Limited capabilities.	Yes
FMT_LIM.2[Loader]	FMT_LIM.1 Limited capabilities.	Yes
FPT_FLS.1	No dependencies.	No dependency
FPT_ITT.1[HW]	No dependencies.	No dependency
FPT_PHP.3	No dependencies.	No dependency
FRU_FLT.2	FPT_FLS.1 Failure with preservation of secure state.	Yes

Tab. 6.53: Dependencies of Security Functional Requirements (PP)

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FCS_COP.1[HW_DES]	[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.	See discussion below.
FCS_COP.1[HW_AES]	[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.	See discussion below.
FDP_ACC.1[MEM]	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1[MEM].
FDP_ACC.1[SFR]	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1[SFR].
FDP_ACF.1[MEM]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1[MEM], FMT_MSA.3[MEM].
FDP_ACF.1[SFR]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1[SFR], FMT_MSA.3[SFR].
FDP_RIP.1[HW]	No dependencies.	
FMT_MSA.1[MEM]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1[MEM], FMT_SMF.1[HW]. For FMT_SMR.1, see discussion below.
FMT_MSA.1[SFR]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1[SFR], FMT_SMF.1[HW]. For FMT_SMR.1, see discussion below.
FMT_MSA.3[MEM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[MEM]. For FMT_SMR.1, see discussion below.

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FMT_MSA.3[SFR]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SFR]. For FMT_SMR.1, see discussion below.
FMT_SMF.1[HW]	No dependencies.	
FMT_SMF.1[SW]	No dependencies.	
FPT_TST.1	No dependencies.	

Tab. 6.54: Dependencies of Security Functional Requirements (ST Part 1)

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FCS_RNG.1[DET]	No dependencies	
FCS_COP.1[SW_DES]	[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.	See note below.
FCS_COP.1[SW_AES]	[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.	See note below.
FDP_RIP.1[SW]	No dependencies	

Tab. 6.55: Dependencies of Security Functional Requirements (ST Part 2)

The developer of the Security IC Embedded Software must ensure that the additional security functional requirements FCS_COP.1[HW_DES] and FCS_COP.1[HW_AES] are used as specified and that the User Data processed by the related security functionality is protected as defined for the application context.
The dependent requirements of FCS_COP.1[HW_DES] and FCS_COP.1[HW_AES] completely address the appropriate management of cryptographic keys used by the specified cryptographic function and the management of access control rights as specified for the memory access control function. All requirements concerning these management functions shall be fulfilled by the environment (Security IC Embedded Software).
The functional requirements [FDP_ITC.1, or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4 are not included in

this Security Target since the TOE only provides a pure engine for encryption and decryption without additional features for the handling of cryptographic keys. Therefore the Security IC Embedded Software must fulfill these requirements related to the needs of the realized application.
The dependency FMT_SMR.1 introduced by the two components FMT_MSA.1[MEM] respectively FMT_MSA.1[SFR] and FMT_MSA.3[MEM] respectively FMT_MSA.3[SFR] must be fulfilled by the Security IC Embedded Software. The definition and maintenance of the roles that act on behalf of the functions provided by the hardware must be subject of the Security IC Embedded Software.

6.3.3 Rationale for the Assurance Requirements

The selection of assurance components is based on the underlying PP [21]. The Security Target uses the same augmentations as the PP, but chooses a higher assurance level. The level EAL5 is chosen in order to meet assurance expectations of digital signature applications and electronic payment systems. Additionally, the requirement of the PP [21] to choose at least EAL4 is fulfilled.
The rationale for the augmentations is the same as in the PP. The assurance level EAL5 is an elaborated predefined level of the CC, part 3 [5]. The assurance components in an EAL level are chosen in a way that they build a mutually supportive and complete set of components. The requirements chosen for augmentation do not add any dependencies, which are not already fulfilled for the corresponding requirements contained in EAL5. Therefore, these components add additional assurance to EAL5, but the mutual support of the requirements is still guaranteed.
As stated in the section 6.3.3 of [21], it has to be assumed that attackers with high attack potential try to attack smart cards used for digital signature applications or payment systems. Therefore specifically AVA_VAN.5 was chosen by the PP [21] in order to assure that even these attackers cannot successfully attack the TOE.

6.3.4 Security Requirements are Internally Consistent

The discussion of security functional requirements and assurance components in the preceding sections has shown that mutual support and consistency are given for both groups of requirements. The arguments given for the fact that the assurance components are adequate for the functionality of the TOE also show that the security functional and assurance requirements support each other and that there are no inconsistencies between these groups.
The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation O.Leak-Forced and O.Cap_Avail_Loader also protect the cryptographic algorithms and the memory access/separation control function as well as the access control to Special Function Register implemented according to the security functional requirement FCS_COP.1[HW_DES], FCS_COP.1[HW_AES] , FCS_COP.1[SW_DES], FCS_COP.1[SW_AES], FCS_RNG.1[DET] and FDP_ACC.1[MEM], FDP_ACC.1[SFR] with reference to the Access Control Policies defined in FDP_ACF.1[MEM] and FDP_ACF.1[SFR]. Therefore, these security functional requirements support the secure implementation and operation of FCS_COP.1[HW_DES], FCS_COP.1[HW_AES] , FCS_COP.1[SW_DES],

FCS_COP.1[SW_AES] and of FDP_ACC.1[MEM] respectively FDP_ACC.1[SFR] with FDP_ACF.1[MEM] respectively FDP_ACF.1[SFR] as well as the dependent security functional requirements.

A Security IC hardware platform requires Security IC Embedded Software to build a secure product. Thereby the Security IC Embedded Software must support the security functionality of the hardware and implement a sufficient management of the security services implemented in the hardware. The realization of the Security Functional Requirements within the TOE provides a good balance between flexible configuration and restrictions to ensure a secure behaviour of the TOE.

7 TOE Summary Specification

7.1 Portions of the TOE Security Functionality

The TOE Security Functionality (TSF) directly corresponds to the TOE security functional requirements defined in Section 6. The Security Functionality provided by the TOE is split into Security Services (SS) and Security Features (SF). Both are active and applicable to phases 4 to 7 of the Security IC product life-cycle.
The TOE also comprises security mechanisms, which are not listed as security functionality in the following. Such mechanisms do not implement a complete Security Services or Security Features. They can be used to implement further Security Services and/or Security Features based on Security IC Embedded Software using these security mechanisms.

7.1.1 Security Services

Tables 7.1 (for PP) and 7.2 and 7.3 (for ST) list the Security Services defined for the TOE.

Name	Title
SS.RNG	Random Number Generation

Tab. 7.1: Security Services defined in the scope of the Protection Profile

Name	Title
SS.SELF_TEST	Self Test
SS.RESET	Reset Functionality

Tab. 7.2: Security Services defined in the extended scope of this Security Target (Part 1)

Name	Title
SS.SW_DES	Triple DES
SS.SW_AES	AES
SS.SW_RNG	Deterministic Random Number Generator

Tab. 7.3: Security Services defined in the extended scope of this Security Target (Part 2)

SS.RNG **Random Number Generation**

The Random Number Generator continously produces random numbers with a length of one byte. The TOE implements SS.RNG by means of a physical hardware random number generator working stable within the valid ranges of operating conditions, which are guaranteed by SF.OPC.

The TOE fulfills AIS31 class PTG.2 [2]. The behaviour of the Random Number Generator is independent of the Security IC Embedded Software. The entropy of the random numbers as claimed by the security functional requirement are ensured by the requirements of AIS31. Therefore SS.RNG obviously meets FCS_RNG.1[HW]. The Crypto Library provides a χ^2 test functionality, which can be used by the Security IC Embedded Software to detect hardware defects and bad quality of the random numbers. The Random Number Generator is suitable for generation of signature key pairs, generation of session keys for symmetric encryption mechanisms, random padding bits, zero-knowledge proofs, generation of seeds for DRNGs and fulfils the online test requirements defined in AIS31 [2].

SS.SW_RNG **Deterministic Random Number Generator**

The CryptoLib implements a software (pseudo) RNG that can be used as a general purpose random source. This software RNG has to be seeded by random numbers taken from the hardware RNG provided via SS.RNG. Then implementation of the software RNG is based on the standard NIST-SP-800-90 CTR-DBRG.

Notice, that Crypto Library is based on AES respectively DES and fulfills AIS31 DRG.3 [2].

SS.SW_DES **Triple DES**

SS.SW_DES supports four modes of operation:

- *ECB according to [11]*
- *CBC according to [11]*
- *CBC-MAC according to [13]*
- *CMAC according to [12]*

A keystore concept is used for secure handling of DES, 2K3DES and 3K3DES keys.

SS.SW_AES **AES**

SS.SW_AES supports four modes of operation:

- *ECB according to [11]*
- *CBC according to [11]*
- *CBC-MAC according to [13]*
- *CMAC according to [12]*

A keystore concept is used for secure handling of AES keys.

SS.SELF_TEST **Self Test**

SS.SELF_TEST provides a function to check whether the TOE has been manipulated physically. This includes an active shielding check, sensor check, verifying the signature of code and performing a consistency check of Special Function Registers with static configuration.

SS.RESET **Reset Functionality**

SS.RESET provides the Security IC Embedded Software with a function to reset the device. This enables the Security IC Embedded Software preserving a secure state in case it detects abnormal operations or attacks. The reset functionality provides an ordinary System Reset (that is, "Power-On Reset") and a security relevant reset (Security Reset) which can be executed only a limited time before the device is disabled permanently.

7.1.2 Security Features

Tables 7.4 (for PP) and 7.5 and 7.6 (for ST) list the Security Features defined for the TOE.

Name	Title
SF.OPC	Control of Operating Conditions
SF.PHY	Protection against Physical Manipulation
SF.LOG	Logical Protection
SF.COMP	Protection of Mode Control

Tab. 7.4: Security Features defined in the scope of the Protection Profile

Name	Title
SF.MEM_ACC	Memory Access Control
SF.SFR_ACC	Special Function Register Access Control
SF.REUSE	Reuse of Memory

Tab. 7.5: Security Features defined in the extended scope of this Security Target (Part 1)

Name	Title
SF.Object_Reuse	Object Reuse

Tab. 7.6: Security Features defined in the extended scope of this Security Target (Part 2)

SF.OPC **Control of Operating Conditions**

SF.OPC ensures the correct operation of the TOE (functions offered by the micro-controller including the standard CPU as well as the unified AES/Triple-DES co-processor, the memories, registers, I/O interfaces and the other system peripherals) during the execution of the IC Dedicated Support Software and Security IC Embedded Software. This includes all specific security features of the TOE which are able to provide an active response.

The TOE ensures its correct operation and prevents any malfunction by means of three kinds of features:

Environmental Control: Set of security mechanisms that detect if the TOE runs out of the specified operation conditions. It needs to be assured that in operation mode all ambient conditions are within their specified limits. Sensors take over the role of measuring the ambient conditions and reacting in case of specification violation of one of the ambient parameters. If a sensors monitors a violation of the specified ambient conditions a reset is triggered. Depending on the type of sensor the reset might be a security reset that decrements the error counter.

Execution Integrity Set of security mechanisms that detect if an execution of an operation has been manipulated. It needs to be assured that manipulations on operations are detected and trigger a reset that affects the error counter. Manipulating operations means the operation itself is attacked. On an abstract view this could mean that some kind of memory (e.g. register) has been attacked. On a more detailed view it can also mean that entire wires or gates are attacked. Executing integrity is achieved by means such as the following ones:

- validity checking of in- and output of security critical operations
- integrity protection of data, code and address path
- integrity protection of memories, data registers, key registers and control registers
- monitoring state machines
- integrity protection of sensor signals
- double calculations and checks

Integrity protection is achieved by various techniques, such as parity protection, redundant encoding and execution, monitoring, CRCs.

Availability Set of security mechanisms that take care that the availability of the TOEs functionality is limited if attacks occur. It needs to be assured that the detection of an attack results in secure state. This is achieved by the fact that any kind of attack or operation outside the operation conditions results in a reset where the TOE boots in the initial configuration. Depending on the kind of reset source the reset might also have an effect on the error counter. This is especially the case for integrity violations that cannot be unintended ones.

SF.PHY **Protection against Physical Manipulation**

The feature SF.PHY protects the TOE against manipulation of

(i) the hardware,

(ii) the IC Dedicated Software in the ROM,

(iii) the Security IC Embedded Software in the NVM and

(iv) the application data in the RAM and NVM including the configuration data stored in NXP_ConfigData_Seg.

It also protects all data stored in the memories including User Data and TSF data against disclosure by physical probing when stored or while being processed by the TOE.

The TOE ensures its correct operation and prevents any malfunction by means of several kinds of features:

- **Layout Protection:** Set of security mechanisms that hamper reverse engineering of the IC, such as layout randomization, active and passive shielding, techniques to hide shielding, multilayer interconnection, wide bus widths and dummy routing.
- **Code- & Datapath Integrity Protection:** Set of security mechanisms that ensure that manipulations on data or code stored and transmitted from memories to the CPU are detected with high probability. This includes integrity protection of the whole code and data path including CPU internals. Integrity verification is always done before the according data is processed, for example, by an ALU operation.
- **Memory Integrity Protection:** Set of security mechanisms that ensure that manipulations on memory content are detected with high probability. This includes integrity protection of memories and registers. EEPROM and FLASH are additionally equipped with error correction codes, double read technology and anti-tearing.
- **Address Path Integrity Protection:** Set of security mechanisms that ensure that manipulations on the address path are detected with high probability.
- **Startup Integrity Protection:** Set of security mechanisms that detect integrity errors during startup (e.g. with respect to configuration data).
- **Redundant Encoding:** Set of security mechanisms that ensure that security critical flags and the according checks are kept with an according level of redundancy.
- **Code Integrity Protection:** Set of security mechanisms that detect if code has been manipulated. This is especially checked by SS.SELF_TEST.
- **Code- & Datapath Encryption:** Set of security mechanisms that ensure that code or data processed by the CPU is stored and transmitted in encrypted form. All data transmitted over the code or datapath is encrypted with an address-dependent non-linear encryption scheme. En- and decryptions are performed in the CPU core.
- **Address Scrambling:** Set of security mechanisms that ensure that physical addresses are scrambled before writing data to the memory.
- **Code- & Datapath Key Management:** Set of security mechanisms that ensure that keys used for the secure data path are derived correctly and securely. This includes address dependent key derivation functionality with an according strength of diffusion and confusion to achieve a good avalanche effect.

SF.LOG **Logical Protection**

SF.LOG implements measures to limit or eliminate the information that might be contained in the shape and amplitude of signals or in the time between events found by measuring such signals. This comprises the power consumption and signals on the other pads that are not intended by the terminal or the Security IC Embedded Software. Thereby SF.LOG prevents the disclosure of User Data or TSF data stored and/or processed in the security IC through the measurement of the power consumption or emanation and subsequent complex signal processing. The protection of the TOE comprises different features within the design that support the other portions of security functionality.

The cryptographic coprocessor includes special features to hamper SPA/DPA analysis of shape and amplitude of the power consumption and ensures that the calculation time is independent from any key and plain/cipher text. These include blinding and randomization techniques.
Specific features as described for SF.PHY (for example, the encryption features) and for SF.OPC (e.g. the filter feature) support the logical protection. For instance, the encryption of the whole data and code path including memory and register contents.

SF.COMP **Protection of Mode Control**

SF.COMP provides a control of the TOE modes. This includes the protection of electronic fuses stored in a protected memory area, and the possibility to store initialisation or pre-personalisation data in the so-called FabKey Area.

The control of the TOE modes prevents the abuse of test functions after TOE delivery. Additionally it also ensures that features used during the boot sequence to configure the TOE cannot be abused. Hardware circuitry and the Boot Software determine whether the test functionality is available or not. If it is available, the TOE starts the IC Dedicated Test Software in the System Mode. Otherwise, the TOE switches to the User Mode or System Mode and starts execution of the Security IC Embedded Software.
The switch to the IC Dedicated Test Software is prevented after TOE delivery because specific electronic fuses guarantee that the IC Dedicated Test Software cannot be selected. The System Mode is the more privileged TOE mode, the User Mode is the less privileged TOE mode. The Boot Software is executed in Super System Mode. HAL Software is executed in Super System Mode (the parts acting as helper function for IC Dedicated Test Software and Boot Software) and System Mode. Application Management Software is also executed in System Mode, but its features for application download, verification and locking must be disabled before the Operational Usage of the TOE. HAL Library and Crypto Library are located in shared code space, where ”shared” means visible in System Mode and User Mode. Security IC Embedded Software has User Mode privileges, but can call functions of HAL Software, HAL Library and Crypto Library and Special Function Registers which are made visible.
The protection of the electronic fuses especially ensures that configuration options with regard to the security functionality cannot be changed, abused or influenced in any way in User Mode. SF.COMP ensures that activation

or deactivation of security features cannot be influenced by the Security IC Embedded Software. SF.COMP limits the capabilities of the test functions and provides test personnel during phase 3 with the capability to store the identification and/or pre-personalization data in the EEPROM.

SF.MEM_ACC **Memory Access Control**

SF.MEM_ACC controls access of any subject (program code comprising processor instructions) to the memories of the TOE.

Code in memories is split into several segments (Subjects) dedicated to TOE modes (see section 6.1.6.1). The memories are split into several segments (Objects) for which Operations and Attributes are defined (see sections 6.1.6.2 and 6.1.6.3). SF.MEM_ACC enforces access rules defined over Subjects, Objects and the associated Operations. Access can be full or conditional. Conditional means that the access depends on a configuration setting of the MMU. In the boot-phase of the TOE these settings are per default switched to a highest level of restriction. Each functionality that is executed in System Mode and needs to access memory segments with restricted accessibility first change the settings of the MMU, then access the according memory segment and afterwards the settings are again disabled. If during execution an error occurs the settings are automatically set to the default state, thus preserving a secure state. Functionality provided by the HAL Software, HAL Library and Crypto Library does the above described setting of segment visibility automatically.
In addition to basic access rules, the MMU checks firewall settings which are also configurable.

SF.SFR_ACC **Special Function Register Access Control**

SF.SFR_ACC implements the access control to the Special Function Registers as specified in the Access Control Policy and the Security Functional Requirements FDP_ACC.1[SFR] and FDP_ACF.1[SFR].

Based on the function of the register and on the TOE mode, the read and/or write access for a specific Special Function Register is allowed or not allowed. SF.SFR_ACC will ignore any read operation on the Special Function Registers that are not allowed or not implemented and will trigger a security reset if happening.
The combination of SF.SFR_ACC and SF.COMP ensures that access to Special Function Registers is restricted to User Mode. Only a restricted subset of Special Function Registers is accessible to the Security IC Embedded Software.
In addition, SF.MEM_ACC permanently checks whether the selected addresses are within the boundaries of the physically implemented memory ranges. Access to outside the boundary of the physically implemented memory ranges forces a reset. Also, SF.MEM_ACC permanently checks for the consistency of addresses.

SF.REUSE **Reuse of Memory**

SF.REUSE ensures that when a HAL session is closed, all security relevant data, such as temporary variables or cryptographic keys, is erased securely.

SF.Object_Reuse **Object Reuse**

The TOE provides internal security measures which clear memory areas used by the CryptoLib after usage.

Additionally all registers holding application specific settings are set to their default values or are erased.

7.2 TOE Summary Specification Rationale

7.2.1 Mapping of Security Functional Requirements and TOE Security Functionality

The following table provides a mapping of portions of the TSF to SFR. The mapping is described in detail in the text following the table.

TSF	SFR	Title
SS.RNG	FCS_RNG.1[HW]	Random Number Generation (Class PTG.2)
SF.OPC	FPT_FLS.1	Failure with Preservation of Secure State
	FRU_FLT.2	Limited Fault Tolerance
SF.PHY	FDP_ITT.1[HW]	Basic Internal Transfer Protection
	FDP_SDI.2[HW]	Stored data integrity monitoring and action
	FPT_ITT.1[HW]	Basic Internal TSF Data Transfer Protection
	FPT_PHP.3	Resistance to Physical Attack
	FMT_SMF.1[HW]	Specification of Management Functions (Hardware)
SF.LOG	FDP_ITT.1[HW]	Basic Internal Transfer Protection
	FDP_IFC.1	Subset Information Flow Control
	FDP_SDC.1[HW]	Stored data confidentiality
	FPT_ITT.1[HW]	Basic Internal TSF Data Transfer Protection
SF.COMP	FAU_SAS.1[HW]	Audit Storage
	FMT_LIM.1[HW]	Limited Capabilities
	FMT_LIM.1[Loader]	Limited Capabilities
	FMT_LIM.2[HW]	Limited Availability
	FMT_LIM.2[Loader]	Limited Availability

Tab. 7.7: TOE Security Functionality vs. Security Functional Requirements (PP0084)

TSF	SFR	Title
SS.SELF_TEST	FPT_TST.1	TSF Testing
SS.RESET	FMT_SMF.1[SW]	Specification of Management Functions (Software)
SF.MEM_ACC	FMT_LIM.2[HW]	Limited Availability

TSF	SFR	Title
	FDP_ACC.1[MEM]	Subset Access Control (Memories)
	FDP_ACF.1[MEM]	Security Attribute Based Access Control (Memories)
	FMT_MSA.3[MEM]	Static Attribute Initialization (Memories)
	FMT_MSA.1[MEM]	Management of Security Attributes (Memories)
	FMT_SMF.1[HW]	Specification of Management Functions (Hardware)
SF.SFR_ACC	FMT_LIM.2[HW]	Limited Availability
	FDP_ACC.1[SFR]	Subset Access Control (Special Function Registers)
	FDP_ACF.1[SFR]	Security Attribute Based Access Control (Special Function Registers)
	FMT_MSA.3[SFR]	Static Attribute Initialization (Special Function Registers)
	FMT_MSA.1[SFR]	Management of Security Attributes (Special Function Registers)
	FMT_SMF.1[HW]	Specification of Management Functions (Hardware)
SF.REUSE	FDP_RIP.1[HW]	Subset Residual Information Protection

Tab. 7.8: TOE Security Functionality vs. Security Functional Requirements (ST Part1)

TSF	SFR	Title
SS.SW_DES	FCS_COP.1[SW_DES]	Cryptographic Operation (DES & TDES)
	FCS_COP.1[HW_DES]	Cryptographic Operation (DES)
SS.SW_AES	FCS_COP.1[SW_AES]	Cryptographic Operation (AES)
	FCS_COP.1[HW_AES]	Cryptographic Operation (AES)
SS.SW_RNG	FCS_RNG.1[DET]	Random Number Generation (Deterministic)
SF.Object_Reuse	FDP_RIP.1[SW]	Subset Residual Information Protection

Tab. 7.9: TOE Security Functionality vs. Security Functional Requirements (ST Part2)

7.2.2 Security architectural information

Since this ST claims the assurance requirement ASE_TSS.2, security architectural information on a very high level is supposed to be included in the TSS to inform potential customers on how the TOE protects itself against interference, logical tampering and bypassing. In the security architecture context, this covers the aspects self-

protection and non-bypassability.

As described in section 7.2.1, the aspects self-protection and non-bypassability are implemented by SF.PHY, SF.OPC, and SF.COMP.

SF.PHY covers the physical protection of the TOE and protects the TOE against tampering and bypassing of security features and security services. SF.OPC contributes by covering the aspects failure with preservation of a secure state and limited fault tolerance. This protects the TOE against interference of security feature and security services. SF.COMP limits the capability and availability of the Test Features and protects the TOE against bypassing of security features. In addition SF.COMP protects the capabilities and availabilities of the Application Management Software.

8 Bibliography

[1] A proposal for: Functionality classes for random number generators, Version 2.0, 18. September 2011.

[2] Anwendungshinweise und Interpretationen zum Schema, AIS31: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, Bundesamt für Sicherheit in der Informationstechnik. Version 2.0, September 18, 2011.

[3] Common Criteria for Information Technology Security Evaluation, Part 1 – Introduction and general model - Version 3.1 CCMB-2012-09-001, Revision 4, September 2012.

[4] Common Criteria for Information Technology Security Evaluation, Part 2 – Security functional components, Version 3.1 CCMB-2012-09-002, Revision 4, September 2012.

[5] Common Criteria for Information Technology Security Evaluation, Part 3 – Security assurance components, Version 3.1 CCMB-2012-09-003, Revision 4, September 2012.

[6] Common Methodology for Information Technology Security Evaluation CEM-99/045 Part 2 - Evaluation Methodology, Version 3.1 CCMB-2012-09-004, Revision 4, September 2012.

[7] E201382 EMBRACE VA and VB, Wafer and delivery specification, NXP Semiconductors, Document number 3406**.

[8] FIPS PUB 197 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, ADVANCED ENCRYPTION STANDARD (AES), National Institute of Standards and Technology, 2001 November 26.

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

[10] JIL: Attack Methods for Smartcards and Similar Devices, Version 1.5, February 2009.

[11] NIST Special Publication 800-38A Recommendation for BlockCipher Modes of Operation. http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

[12] NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf.

[13] NIST Special Publication 800-38C Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality - CBCMAC. http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C.pdf.

[14] NXP Secure Smart Card Controller E201382, Guidance, Delivery and Operation Manual, Document number 281114.

[15] Order Entry Form E201382 EMBRACE, NXP Semiconductors, online document, Business Unit Identification.

[16] Product data sheet addendum E201382 EMBRACE, Application Management, NXP Semiconductors, Document number 2799**.

[17] Product data sheet addendum E201382 EMBRACE, Crypto Library, NXP Semiconductors, Document number 2800**.

[18] Product data sheet addendum E201382 EMBRACE, Instruction Set Manual, NXP Semiconductors, Document number 2773**.

[19] Product data sheet addendum E201382 EMBRACE, System interface specification, NXP Semiconductors, Document number 2798**.

[20] Product data sheet E201382 EMBRACE, Secure smart card controller, NXP Semiconductors, Document number 2542**.

[21] Security IC Platform Protection Profile with Augmentation Packages, registered and certified by Bundesamt fuer Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, Rev 1.0, 13 January 2014.

[22] Supporting Document Mandatory Technical Document Application of Attack Potential to Smartcards, Version 2.7, Revision 1, March 2009, CCDB-2009-03-001 .

[23] Identification cards, Contactless integrated circuit cards, Proximity cards, Part 1: Physical characteristics, 06 2008.

9 Contents

10 Legal information

10.1 Definitions

Draft – The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

10.2 Disclaimers

Limited warranty and liability – Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes – NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use – NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications – Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control – This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products – This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

10.3 Licenses

ICs with DPA Countermeasures functionality

NXP ICs containing functionality implementing countermeasures to Differential Power Analysis and Simple Power Analysis are produced and sold under applicable license from Cryptography Research, Inc.

10.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.
<Patent ID> – owned by <Company name>

10.5 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

MIFARE – is a trademark of NXP B.V.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 25 September 2017

Document identifier: