NXP Secure Smart Card Controller P40C012/040/072 VD

Security Target Lite

Rev. 1.1 – 2014-09-25 **Evaluation documentation**

Final **Public**

NSCIB-CC-13-37658

Document Information

Info	Content
Keywords	CC, Security Target Lite, P40C012/040/072 VD
Abstract	Security Target Lite of the NXP Secure Smart Card Controller P40C012/040/072 VD, which is developed and provided by NXP Semiconductors, Business Unit Identification according to the Common Criteria for Information Technology Security Evaluation Version 3.1 at EAL5 augmented

Rev	Date	Description
1.0	07-August-2014	Final Version
1.1	25-September-2014	Updated after comments from the scheme.

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 P40C012/040/072 VD Security Target, 1.1, NXP Semiconductors, 2014-09-25.

1.2 TOE Reference

NXP Secure Smart Card Controller P40C012/040/072 VD

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 P40C012/040/072 VD 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 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. Notice, that the EEPROM is also referred to as Non-Volatile Memory (NVM) in this Security Target.

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, it provides a Hardware Abstraction Layer (HAL) (the HAL Software) simplifying the access to the hardware for the Security IC Embedded Software.

The documentation includes a Data Sheet with several addenda, such as Firmware Interface Specification or Special Function Register specification for different TOE modes, 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.
P40C012/040/072 VD provides high security for smart card applications and in particular for being used in the banking and finance market, in electronic commerce or in governmental applications. Hence, P40C012/040/072 VD 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 P40C012/040/072 VD 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) with different key lengths,
- support for large integer arithmetic operations like multiplication, addition and logical operations, which are suitable for public key cryptography and elliptic curve cryptography,
- a True Random Number Generator,
- memory management control,
- cyclic redundancy check (CRC) calculation, and
- an ISO/IEC 7816 contact interface with UART.

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.

Note: Large integer arithmetic operations are intended to be used for calculation of asymmetric cryptographic algorithms. Any asymmetric cryptographic algorithm utilizing the support for large integer arithmetic operations has to be implemented to the Security IC Embedded Software. The support for large integer arithmetic operations does not provide security functionality like cryptography. The Security IC Embedded Software that implements an asymmetric cryptographic algorithm is not included in this Security Target, but the support for large integer arithmetic operations is a security relevant component of the TOE which is still protected under the claims for the Security Feature SF.OPC. The same statements hold for the CRC calculation.

1.3.2 TOE Type

The TOE NXP Secure Smart Card Controller P40C012/040/072 VD is provided as 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

P40C012/040/072 VD is manufactured in an advanced 90nm CMOS technology. A block diagram of the IC hardware is depicted in Fig 1.1.

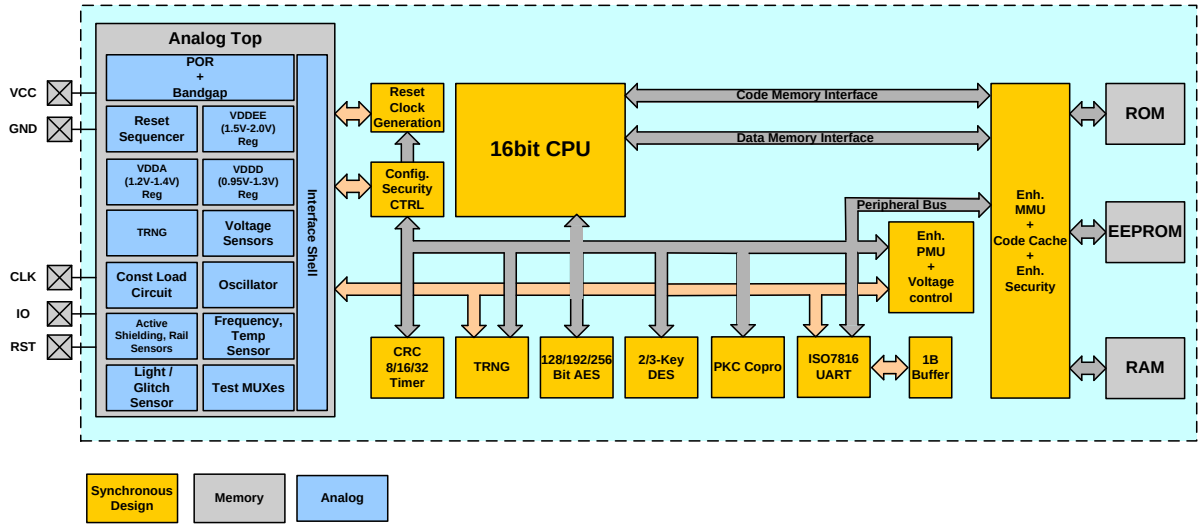

Fig. 1.1: Block Diagram

P40C012/040/072 VD 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. 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	P40C012/040/072 VD	VD	2014-02-06	wafer, module (dice have nameplate 9511D)
IC Dedicated Test Software	Test Software	00h	2014-02-06	in ROM[1]
IC Dedicated Support Software	Boot Software	00h	2014-02-06	in ROM[1]
	HAL Software	00h	2014-02-06	in ROM[1]
Document	Product data sheet SmartMX2 P40 family P40C012/040/072, Secure high-performance smart card controller, NXP Semiconductors [19]	262923	2014-06-27	Electronic Document
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Firmware interface specification, NXP Semiconductors [13]	275823	2014-06-27	Electronic Document
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, User Mode, NXP Semiconductors [17]	275722	2014-06-27	Electronic Document
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, System Mode, NXP Semiconductors [16]	267522	2014-06-27	Electronic Document
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Chip Health Mode, NXP Semiconductors [12]	269720	2014-05-21	Electronic Document
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Post Delivery Configuration, NXP Semiconductors [15]	269620	2014-05-21	Electronic Document
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Instruction Set Manual, NXP Semiconductors [14]	258121	2014-06-27	Electronic Document
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx VA and VD, Wafer specification, NXP Semiconductors [18]	269822	2014-06-03	Electronic Document
Document	Guidance and Operation Manual NXP Secure Smart Card Controller P40C012/040/072, Information on Guidance and Operation, NXP Semiconductors [9]	269422	2014-06-27	Electronic Document

Tab. 1.1: Components of the TOE

[1] according to ROM image 9511D_18_00_SSM_code.dat and the files listed in [9]

The IC Hardware is identified by its nameplate VD, that is located in the layout of the chip (see [18] how to inspect the nameplate). 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 [13]. The 9511D_18_00_SSM_code.dat file is the NXP ROM image which is mixed with the customer ROM image in the ROM Code Shop. Notice, that the developer of a composite product also needs several files (such as header- and c-files) for generating the hex-image of the composite product. These files are listed in [9] together with a hash-value for identification as evaluated versions.

1.4.2 Evaluated Configurations

The customer selects logical and physical configuration options of P40C012/040/072 VD 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 1.4.2.4.

1.4.2.1 Major configuration options

Three major configurations are present, which are denoted by the names P40C012 VD, P40C040 VD and P40C072 VD. All of them are equipped with an EEPROM of 72 kBytes. Their difference is the availability of EEPROM space.

Each major configuration is provided with several minor configuration options, which are introduced in Section 1.4.2.2. Each major configuration also provides customers with several options for reconfiguration (Post Delivery Configuration), which are described in Section 1.4.2.3 in detail.

The major configuration is chosen by the customer via Order Entry Form [11].

The Order Entry Form [11] 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 [11] belonging to.

1.4.2.2 Minor configuration options

Minor configurations are chosen by the customer via Order Entry Form [11] as detailed in Table 1.2. The Order Entry Form [11] identifies the minor configuration options, which are supported by a major configuration.

Product option	Choices	Description
Enable DES coprocessor	YES or NO	This option determines whether the DES coprocessor is enabled or disabled. Default value is YES.
Enable AES coprocessor	YES or NO	This option determines whether the AES coprocessor is enabled or disabled. Default value is YES.
Enable PKC coprocessor	YES or NO	This option determines whether the PKC coprocessor is enabled or disabled. Default value is YES.
Enable Post Delivery Configuration	YES or NO	This option determines whether the Post Delivery Configuration is enabled or disabled. Default value is YES.

Product option	Choices	Description
Enable Chip Health Mode	YES or NO	This option determines whether the Chip Health Mode is enabled or disabled. Default value is YES.
Enable Error Counter Handling	YES or NO	This option determines whether the error counter handling is enabled or disabled. Default value is YES.
Number of key parts in EE Keystore	Value between 00h and FFh	This value determines the number of keys in the EEPROM keystore. Default value is 00h.
Number of EEPROM Anti-tearing pages	Value between 00h and FFh	This value determines the number of anti-tearing pages in the EEPROM. Default value is 04h.
Application patch size	Value between 00h and FFh	This value determines the application patch size. Default value is 02h.
System Mode patch size	Value between 00h and FFh	This value determines the System Mode patch size. Default value is 01h.
Shared code patch size	Value between 00h and FFh	This value determines the Shared code patch size. Default value is 00h.

Tab. 1.2: Evaluated minor configuration options

1.4.2.3 Post Delivery Configuration

Post Delivery Configuration (PDC) can be applied by the customer himself after the TOE has been delivered to that customer. These options can be used to tailor the TOE to the specific customer requirements. The Post Delivery Configuration can be changed multiple times but must be set permanently by the customer before the TOE is delivered to phase 7 of the life-cycle.
The Post Delivery Configuration for the P40C012/040/072 VD is listed in Table 1.3.

PDC option	Description
DES	Disable the DES coprocessor if enabled via OEF.
AES	Disable the AES coprocessor if enabled via OEF.
PKCC	Disable the PKC coprocessor if enabled via OEF.
PDC	Disable the PDC if enabled via OEF.
EEPROM Size	Reduce the size of the available EEPROM memory in steps of 512 bytes.
NumEEKeys	Reduce the number of available Keys in EEPROM Keystore.
NumATPages	Reduce the number of available Anti tearing pages.

Tab. 1.3: Post Delivery Configuration

As indicated in the description of the PDC options, they can only be used to downgrade some configurations of the OEF. For instance, if DES is enabled in the OEF it can be disabled via PDC, but not vice versa.
By applying Post Delivery Configuration the NXP_ConfigData_Seg (the EEPROM segment holding IC configuration data) content is updated for the changed configuration options and can therefore be used for identification of the TOE after applying any Post Delivery Configuration. Further details regarding NXP_ConfigData_Seg content

and identification of the TOE after applying Post Delivery Configuration refer to [15].
The Post Delivery Configuration can be accessed using chip health mode functionality in combination with the ISO/IEC 7816 contact interface.

1.4.2.4 Evaluated package types

The commercial types are named according to the following format.

- P4*nxeeepp(p)/mvrrff*

Italic characters in the above format are replaced as described in Table 1.4 and Table 1.5 for to retrieve a commercial type name. The commercial type name is composed of fixed symbols, which are detailed in Table 1.4, and variable entries, which are filled in according to the rules in Table 1.5.

Variable	Description	Values	Evaluated Options
n	Number of P4 generation	numeric	'0' for evolution 0
x	Interface and Feature Configuration	alpha numeric	'C' for Contact interface
eee	Indication of Non-Volatile Memory Size	numeric	'012' for 12KB, '040' for 40KB, '072' for 72KB
pp(p)	Package delivery type	alpha numeric	see table 1.5
/	separator (mandatory)		
m	Manufacturer identifier	apha numeric	'9' for TSMC-Fab9
v	Version of mask set	alphabetic	'D' for HW version VD
rr	ROM code number, which identifies the ROM mask	alpha numeric	customer individual
ff	FabKey number, which identifies the EEPROM content at TOE delivery	alpha numeric	customer individual

Tab. 1.4: Variable Definitions for Commercial Type Names

Type	Description
U*x*	Wafer not thinner than 50μm (The letter "*x*" in "U*x*" stands for a capital letter or a number, which identifies the wafer type)
X*n*	Module (The letter "*n*" in "X*n*" stands for a capital letter or a number, which identifies the module type)

Tab. 1.5: Supported Package Types

For example, commercial type name P40C072X60/9Drrff denotes major configuration P40C072 VD in PCM3.1 contact chip card module (super 35 mm tape frame carrier; electronic fail die marking according to SECSII format). 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.
Notice that the IC Dedicated Software version is not explicitly reflected in the commercial type name, but can be retrieved via the GetVersion command described in [13].
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 ROM code number and the FabKey number of the commercial type name.
Minor configuration options as well as configuration options changed by means of Post Delivery Configuration are coded in the NXP_ConfigData_Seg EEPROM segment and can be read out for identification of the TOE. Further details regarding the NXP_ConfigData_Seg EEPROM segment content and identification of the TOE after applying Post Delivery Configuration refer to [15].

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 resp. 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. The Security IC Embedded Software may execute in System Mode or 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 modeled as an own mode (in that context sometimes referred to as 'Test Mode') that has extended access rights compared to System Mode.
The P40C012/040/072 VD is able to control two different logical phases. After production of the Security IC every start-up or reset completes with execution of the IC Dedicated Test Software. The test functionality is disabled at the end of the production test. Afterwards, every start-up or reset ends up in System Mode resp. User Mode and execution of the Security IC Embedded Software.

In case the minor configuration option 'Enable Post Delivery Configuration' is enabled and not finally locked by the customer, the resource configuration functionality allows the customer to enable or disable specific functionality of the hardware platform, refer to Table 1.3.
In case the minor configuration option 'Enable Chip Health Mode' is enabled, during the boot process routines either starting built-in self tests checking the functional integrity of the TOE or sending back identification items of the TOE can be activated by the user. System Mode and User Mode are 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, UART and the coprocessors.
The P40C012/040/072 VD 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, P40C012/040/072 VD 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 P40C012/040/072 VD incorporates 260 kBytes of ROM, 6144 Bytes of RAM and 72 kBytes of EEPROM. Access control to all three memory types is enforced by a Memory Management Unit (MMU). The HAL Software provides simplification of the access control together with the 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 three different key lengths of 128, 192 or 256 bit. The Public Key Crypto Coprocessor (PKCC) coprocessor supplies basic arithmetic functions to support implementation of asymmetric cryptographic algorithms by the Security IC Embedded Software. The random number generator provides true random numbers without pseudo random calculation. The CRC coprocessor provides CRC generation polynomial CRC-8, CRC-16 and CRC-32.
The P40C012/040/072 VD operates with a single external power supply of 1.8V, 3V or 5V nominal. The maximum external clock frequency used for synchronization of the ISO/IEC 7816 communication is 10 MHz nominal, the CPU and all coprocessors are supplied exclusively with an internally generated clock signal which frequency can be selected by the Security IC Embedded Software. The P40C012/040/072 VD provides power saving modes with reduced activity. These are named IDLE Mode and CLOCK STOP Mode.
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, ROM and EEPROM. 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 and/or in the UM_PatchCode_Seg for customers developing code for User Mode (see Figure 1.2; "User Mode Customer Code") and in the SM_Code_Seg and/or in the SM_PatchCode_Seg for customers developing code for System Mode (see Figure 1.2; "System Mode Customer Code"). Both are not part of the TOE.

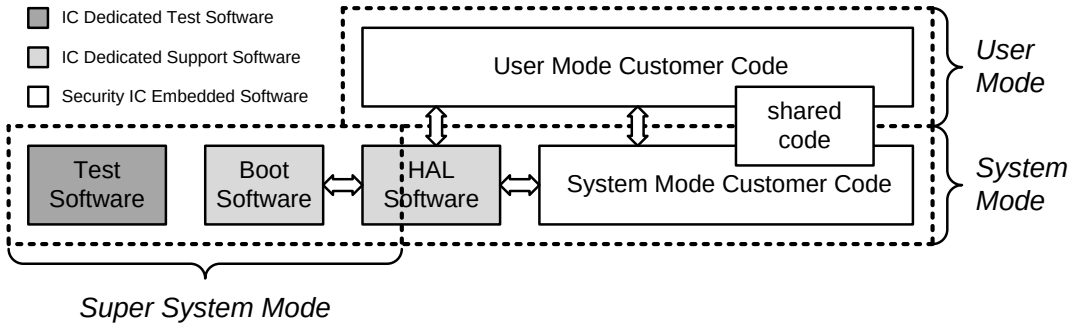

Fig. 1.2: Software Components of the TOE

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 shutdown 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) or User Mode (stored in UM_PatchCode_Seg) .

The IC Dedicated Support Software comprises the following two 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 and NXP_TrimData_Seg, respectively . The Boot Software is stored in the BootTest-Code_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, the Post-Delivery Configuration feature 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.

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:

- " Product data sheet SmartMX2 P40 family P40C012/040/072, Secure high-performance smart card controller, NXP Semiconductors, Document number 262923, 2014-06-27" [19] contains a basic functional description of the hardware
- " Product data sheet addendum SmartMX2 P40 family P40Cxxx, Firmware interface specification, NXP Semi- conductors, Document number 275823, 2014-06-27" [13] contains the interface description of HAL Software visible to the Security IC Embedded Software
- "Product data sheet addendum SmartMX2 P40 family P40Cxxx, User Mode, NXP Semiconductors, Document number 275722, 2014-06-27" [17] contains the a collection of Special Function Registers accessible in User Mode
- "Product data sheet addendum SmartMX2 P40 family P40Cxxx, System Mode, NXP Semiconductors, Document number 267522, 2014-06-27" [16] contains the a collection of Special Function Registers accessible in System Mode
- "Product data sheet addendum SmartMX2 P40 family P40Cxxx, Chip Health Mode, NXP Semiconductors, Document number 269720, 2014-05-21" [12] contains descriptions and guidelines for the Chip Health Mode
- "Product data sheet addendum SmartMX2 P40 family P40Cxxx, Post Delivery Configuration, NXP Semiconductors, Document number 269620, 2013-05-21" [15] contains descriptions and guidelines for the Post Delivery Configuration
- "Product data sheet addendum SmartMX2 P40 family P40Cxxx, Instruction Set Manual, NXP Semiconductors, Document number 258121, 2014-06-27" [14] contains a detailed specification of the CPU instructions
- "Product data sheet addendum SmartMX2 P40 family P40Cxxx VA and VD, Wafer specification, NXP Semiconductors, Document number 269822, 2014-06-03" [18] describes physical identification of the TOE and the secure delivery process
- "Guidance and Operation Manual NXP Secure Smart Card Controller P40C012/040/072, Information on Guid- ance and Operation, NXP Semiconductors, Document number 269422, 2014-06-27" [9] 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

The Security IC product life-cycle is scheduled in phases as introduced in the PP [20]. 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 the 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. The development and production environment of the TOE ranges from phase 2 to TOE Delivery.
With respect to Application Note 3 in [20] the TOE supports the authentic delivery using the 'Enable Chip Health Mode' and the FabKey feature. For further details please refer to the data sheet [19] and the guidance and operation manual [9].
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 ROM 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 DIF 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

1.4.6 TOE Intended Usage

The end-consumer environment of the TOE is phase 7 of the Security IC product life-cycle as defined in the PP [20]. In this phase the Security IC product is in usage by the end-consumer. Its method of use now depends on the Security IC Embedded Software. The Security ICs including the P40C012/040/072 VD can be used to assure authorized conditional access in a wide range of applications. Examples are identity cards, Banking Cards, Pay-TV, Portable communication SIM cards, Health cards and Transportation cards. The end- user environment covers a wide spectrum of very different functions, thus making it difficult to monitor and avoid abuse of the TOE. The TOE is intended to be used in an insecure environment, which does not protect against threats.
The device is developed for most high-end safeguarded applications, and is designed for embedding into chip cards according to ISO/IEC 7816 [10]. Usually a Security IC (e.g. a smart card) is assigned to a single individual only, but it may also be used by multiple applications in a multi-provider environment. Therefore the TOE might store and process secrets of several systems, which must be protected from each other. The TOE then must meet security requirements for each single security module. Secret data shall be used as input for calculation of authentication data, calculation of signatures and encryption of data and keys.
In development and production environment of the TOE the Security IC Embedded Software developer and system integrators such as the terminal software developer may use samples of the TOE for their testing purposes. It is not intended that they are able to change the behavior of the Security IC in another way than an end-consumer.

The user environment of the TOE ranges from TOE delivery to phase 7 of the Security IC product life-cycle, and must be a controlled environment up to phase 6.

Note: The phases from TOE Delivery to phase 7 of the Security IC Product life-cycle are not part of the TOE construction process in the sense of this Security Target. Information about these phases is just included to describe how the TOE is used after its construction. Nevertheless such security functionality of the TOE, that is independent of the Security IC Embedded Software, is active at TOE Delivery and cannot be disabled by the Security IC Embedded Software in the following phases.

1.4.7 Interface of the TOE

The electrical interface of the P40C012/040/072 VD are the pads to connect the lines power supply, ground, reset input, clock input, serial communication pad I/O. Communication with the TOE can be established via the contact interface through the ISO/IEC 7816 UART.
The logical interface of the TOE depends on the CPU mode and the associated software.

- Upon every start-up the Boot Software is executed in Super System Mode. This software initializes and configures the TOE. This comprises the selection of IC Dedicated Test Software (before TOE delivery) and of Security IC Embedded Software (after TOE delivery). Only in case the minor configuration option 'Enable Chip Health Mode' is enabled, starting of built-in self test routines and read-out of TOE identification items is supported. If this minor configuration option is disabled the Boot Software provides no interface. In this case there is no possibility to interact with this software. The Boot Software is stored in the BootTestCode_Seg.
- Before TOE delivery the logical interface is defined by the IC Dedicated Test Software. This IC Dedicated Test Software is executed in Super System Mode and comprises the test operating system used for production testing. IC Dedicated Test Software is stored in the BootTestCode_Seg.
- In System Mode and User Mode (after TOE Delivery) the software interface is the set of instructions, the bits in the special function registers that are related to these modes and the physical address map of the CPU including memories. The access to the special function registers as well as to the memories depends on the TOE mode configured by the Security IC Embedded Software.

Note: The logical interface of the TOE that is visible on the electrical interface after TOE Delivery is based on the Security IC Embedded Software developed by the software developer. The identification and authentication of the user in System Mode or User Mode must be controlled by the Security IC Embedded Software.

The chip surface can be seen as an interface of the TOE, too. This interface must be taken into account regarding environmental stress e.g. like temperature and in the case of an attack, for which the attacker manipulates the chip surface.

Note: An external voltage and timing supply as well as a logical interface are necessary for the operation of the TOE. Beyond the physical behavior the logical interface is defined by the Security IC Embedded Software.

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" [20] 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, [20]. Thus, the concepts are used in the same sense. For the definition of terms refer to [20]. This chapter does not need any supplement in the Security Target.
The TOE provides additional functionality, which is not covered in [20]. In accordance with Application Note 4 of [20] 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 [20].
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 [20].
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, [20].

3 Security Problem Definition

This chapter lists the assets, threats, assumptions and organizational security policies from the PP [20] 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 [20], 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.

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 [20], are valid for this Security Target. These threats are listed in Table 3.1.

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
T.RND	Deficiency of Random Numbers

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

In compliance with Application Note 5 in the PP [20] 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 these 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 [20], are valid for this Security Target. These security policies are listed in Table 3.3.

Name	Title
P.Process-TOE	Protection during TOE Development and Production

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

In compliance with Application Note 6 in the PP [20], 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
- Self Testing
- A function to reset the device
- Integrity support of data stored to NVM
- Reconfiguration of customer selectable options according to Post Delivery Configuration

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 [20], 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.Plat-Appl	Usage of Hardware Platform
A.Resp-Appl	Treatment of User Data

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

In compliance with Application Notes 7 and 8 in PP [20], 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 [20].

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 [20], 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

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

In compliance with Application Notes 9 and 10 in the PP [20], additional security objectives for the TOE are defined below based on additional functionality provided by the TOE.

O.HW_DES3 **Triple DES Encryption**
The TOE must provide the cryptographic functionality to calculate a Triple DES encryption and decryption of one block. The TOE supports directly the calculation of Triple DES with two keys (112 bit) and three keys (168 bit). Note: The TOE will ensure the confidentiality of the User Data (and especially cryptographic keys) during Triple DES operation. This is supported by O.Leak-Inherent.

O.HW_AES **AES Encryption**
The TOE shall provide the cryptographic functionality to calculate an AES encryption and decryption of one block. The TOE supports directly the calculation of AES with the key lengths 128, 192 and 256 bit. Note: The TOE will ensure the confidentiality of the User Data (and especially cryptographic keys) during AES operation. This is supported by O.Leak-Inherent.

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 User Mode and System Mode. In 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. The TOE shall provide System Mode with the ability to configure access rights for User Mode to Special Function Registers that interface to hardware components.

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.CUST_RECONFIG **Post Delivery Configuration**
The TOE shall provide the customer with the functionality to reconfigure parts of the TOE properties as specified for the Post Delivery Configuration.

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

Name	Title
O.HW_DES3	Triple DES Encryption
O.HW_AES	AES Encryption
O.INTEGRITY_CHK	Integrity Control of Transferred Data
O.CUST_RECONFIG	Post Delivery Configuration
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.Self-Test	Self Test
O.Reset	Reset function

Tab. 4.2: Security Objectives of the TOE defined in this ST

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 [20], are applied to this Security Target. These security objectives are listed in Table 4.3.

Name	Title
OE.Plat-Appl	Usage of Hardware Platform
OE.Resp-Appl	Treatment of User Data

Tab. 4.3: Security Objectives of the Development Environment defined in the Security IC Platform Protection Profile

Clarification related to "Usage of Hardware Platform (OE.Plat-Appl)"
The TOE supports cipher schemes as additional specific security functionality. If required the Security IC Embedded Software shall use these cryptographic services of the TOE and their interface as specified. When key-dependent functions implemented in the Security IC Embedded Software are just being executed, the Security IC Embedded Software must provide protection against disclosure of confidential data (User Data) stored and/or processed in the TOE by using the methods described under T.Leak-Inherent and T.Leak-Forced.
If the random number generator is used for leakage countermeasures, cryptographic operations (e.g. key generation) or cryptographic protocols (e.g. challenge response) these random numbers must be tested appropriately.
In case the Security IC Embedded Software operates multiple applications on the TOE, it can implement a memory management scheme based on security functionality of the TOE to ensure separation of these applications.
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 [20], are applied to this Security Target. These security objectives are listed in Table 4.4.

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

Tab. 4.4: Security Objectives of the Operational Environment defined in the Security IC Platform Protection Profile

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 security policy 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 4.5.

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

Tab. 4.5: Security Objectives of the Operational Environment defined in this ST

4.4 Security Objectives Rationale

Section 4.4 in the PP [20] provides a rationale how the threats, organisational security policies and assumptions are addressed by the security objectives defined in the PP [20]. Table 4.6 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 [20].

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

Security Problem Definition	Security Objective	Notes
A.Process-Sec-IC	OE.Process-Sec-IC	Phases 4–6
A.Plat-Appl	OE.Plat-Appl	Phase 1
A.Resp-Appl	OE.Resp-Appl	Phase 1

Tab. 4.6: Security Objectives (PP and ST) vs. Security Problem Definition (PP)

Table 4.7 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.HW_DES3 O.HW_AES O.Self-Test O.Reset O.CUST_RECONFIG O.NVM_INTEGRITY	
A.Check-Init	OE.Check-Init	Phases 1 and 4–6
A.Key-Function	OE.Plat-Appl OE.Resp-Appl	Phase 1 Phase 1

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

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.
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.HW_DES3	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.HW_AES	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.Self-Test	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.Reset	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.CUST_RECONFIG	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.NVM_INTEGRITY	This objective covers the security policy because it requires the TOE to partly implement the functionality 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.Plat-Appl	The definition of this objective of the PP [20] is further clarified in this Security Target: If required the Security IC Embedded Software shall use the cryptographic services of the TOE and its interface as specified. In addition, the Security IC Embedded Software (i) must implement operations on keys (if any) in such a manner that they do not disclose information about confidential data and (ii) must configure the memory management in a way that different applications are sufficiently separated. If the Security IC Embedded Software uses random numbers provided by the security service SS.RNG these random numbers must be tested as appropriate for the intended purpose. This addition ensures that A.Key-Function is still covered this objective although additional functions are being supported according to P.Add-Components.
OE.Resp-Appl	The definition of this objective of the PP [20] 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 this objective although additional functions are being supported according to P.Add-Components.

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 [20] defines extended security functional requirements 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 [20] 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 [20] 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 [20] 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 [20] 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
FMT_LIM.1[HW]	Limited Capabilities
FMT_LIM.2[HW]	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
FRU_FLT.2	Limited Fault Tolerance

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

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_SDI.2[HW]	Stored Data Integrity Monitoring and Action
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

6.1.1 SFRs of the Protection Profile

All SFRs, which are defined in the PP [20], 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 [20]. Others are newly defined in the PP [20].

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 [20]. 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 [20] and there is subject to two assignment operations. A third assignment operation is left open in the PP [20]. 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 Prepersonalisation Data and/or supplements of the Security IC Embedded Software* in the *NVM*.

For FCS_RNG.1.1 the PP [20] 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 [20] 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 [20] 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 [20]. 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 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 [20] according to [1].

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

Note: Application Note 20 in [20] 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 compliance with Application Note 12 in the PP [20] 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 [20].

As required by Application Note 14 in the PP [20] the secure state is described in Section 7.2.1 in the rationale for SF.OPC.

Regarding Application Note 15 in the PP [20] generation of additional audit data is not defined for requirements FRU_FLT.2 and FPT_FLS.1.

As required by Application Note 18 in the PP [20] 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 [7]*
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 cryptographic key sizes*of 128, 192 or 256 bit* that meet the following *standards*: • *FIPS Publication 197, Advanced Encryption Standard (AES), NIST Special Publication 800-38A, 2001 [7]*

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* • *Performing a Security 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 re-boots the IC. The Security Reset re-boots the device and decreases an error counter. Once the error counter is reaching 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 (EEPROM adjustment operation of the) TOE shall meet the requirement ”Stored data integrity monitoring and action (FDP_SDI.2)” as specified below.

FDP_SDI.2[HW]	**Stored Data Integrity Monitoring and Action**

Hierarchical-To	FDP_SDI.1
Dependencies	No dependencies.
FDP_SDI.2.1[HW]	The TSF shall monitor User Data stored in containers controlled by the TSF for *integrity violations* on all objects, based on the following attributes: *User Data including code stored in NVM*.
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*.

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 as part of the IC Dedicated Support Software and parts of the Security IC Embedded Software (System Mode Customer Code), executed as instructions by the CPU.	

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

Subject	PKCC	Public Key Crypto Coprocessor
Info	The Public Key Crypto Coprocessor (PKCC) configured by the Secure IC Embedded Software for implementation of asymmetric cryptographic algorithms and direct memory access to the PKCC_RAM_Seg for accessing operands and storing resulting data.	

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	PKCC_RAM_Seg	PKCC RAM Segment
Info	Located in the RAM memory, used for Public Key Crypto Coprocessor operations.	
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	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	NXP_TrimData_Seg	NXP Trim Data Segment
Info	Located in the EEPROM memory and has a fixed size. Stores trim values for all EEPROM pages.	
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 mainly 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.
Attribute	size	Configuration of size via SFR_MemSegCfg.

Object	UM_ROMConst_Seg	UM ROM Constant Segment
Info	Located in the ROM memory, stores constants for User Mode.	
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.	
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 r/w 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 ROM 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	RAM_CodeMirror_Se	RAM Code Mirror Segment
Info	Located in the RAM memory and its size depends on the physical module size. This is for test purposes and can there be used in Super System Mode only.	
Operation	execute	Execute code.
Attribute	enable	Enable r/w access via SFR_AccCtrlCfg.

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.

Object	UM_PatchCode_Seg	UM Patch Code Segment
Info	Located in the EEPROM memory. Contains the patch code of the TOE that is intended to replace or extend any User 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 and EEPROM.	
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 and EEPROM as well as access Special Function Registers.	
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_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 • Public Key Crypto 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:ROM3 *The SSM_Code is allowed to perform UM_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.read and SM_Code_Seg.execute.*

ACF12.MEM:SSM:ROM9 *The SSM_Code is allowed to perform SharedCode_Seg.execute.*

ACF12.MEM:SSM:EE1 *The SSM_Code is allowed to perform EE_UserData_Seg.read and EE_UserData_Seg.write.*

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:ROM3 *The SM_Code is allowed to perform UM_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.read and SM_Code_Seg.execute.*

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

ACF12.MEM:SM:EE1 *The SM_Code is allowed to perform EE_UserData_Seg.read and EE_UserData_Seg.write.*

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:ROM3 *The UM_Code is allowed to perform UM_ROMConst_Seg.read.*

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

ACF12.MEM:UM:ROM8 *The UM_Code is allowed to perform UM_Code_Seg.execute.*

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

ACF12.MEM:UM:EE1 *The UM_Code is allowed to perform EE_UserData_Seg.read.*

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:ROM1c *The SSM_Code is allowed to perform UM_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 EE_Mirror_Seg.enable grants this right or SYS0 has been called.*

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:EE3 *The SSM_Code is allowed to perform NXP_TrimData_Seg.read and NXP_TrimData_Seg.write if the attribute NXP_TrimData_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:EE5 *The SSM_Code is allowed to perform UM_PatchCode_Seg.read and UM_PatchCode_Seg.write if the attribute UM_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: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:SSM:RAM5 *The SSM_Code is allowed to perform PKCC_RAM_Seg.read and PKCC_RAM_Seg.write if the attribute PKCC_RAM_Seg.enable grants this right.*

ACF13.MEM:SSM:RAM6 *The SSM_Code is allowed to perform RAM_CodeMirror_Seg.execute if the attribute RAM_CodeMirror_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:ROM1c *The SM_Code is allowed to perform UM_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: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:EE3 *The SM_Code is allowed to perform NXP_TrimData_Seg.read and NXP_TrimData_Seg.write if the attribute NXP_TrimData_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:EE5 *The SM_Code is allowed to perform UM_PatchCode_Seg.read and UM_PatchCode_Seg.write if the attribute UM_PatchCode_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:SM:RAM5 *The SM_Code is allowed to perform PKCC_RAM_Seg.read and PKCC_RAM_Seg.write if the attribute PKCC_RAM_Seg.enable grants this right.*

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

ACF13.MEM:UM:EE5 *The UM_Code is allowed to perform UM_PatchCode_Seg.execute if the attribute UM_PatchCode_Seg.enable grants this right.*

ACF13.MEM:UM:RAM4 *The UM_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:RAM5 *The UM_Code is allowed to perform PKCC_RAM_Seg.read and PKCC_RAM_Seg.write if the attribute PKCC_RAM_Seg.enable grants this right.*

ACF13.MEM:UM:RAM5 *The PKCC is allowed to perform PKCC_RAM_Seg.read and PKCC_RAM_Seg.write if the attribute PKCC_RAM_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:GRP2	*The UM_Code is allowed to perform SFR_MemSegCfg.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:

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:

ACF14.SFR:UM:GRP4.1	*The UM_Code is not allowed to access any Special Function Register related to User Mode if SFR_AccCtrlCfg grants 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 except DM9_RAM_Seg* to *code executed in System Mode or Super System Mode.*
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)*
- *Temporary disabling and enabling of the security functionality EEPROM Size, NumEEKeys, NumATPages, AES, DES, PKCC*
- *Permanently disabling and enabling of the security functionality EEPROM Size, NumEEKeys, NumATPages, AES, DES, PKCC*

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 6.33 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 [20], 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

Name	Title
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. 6.33: Security Assurance Requirements

6.2.1 Refinements of the TOE Security Assurance Requirements

The Security Target claims conformance to the PP [20] and therefore it has to be conform to the refinements of the TOE security assurance requirements (see Application Note 22 in [20]). Because the refinements in the PP [20] 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 6.34 lists the influences of the refinements of the PP [20] on the Security Target. Most of the refined security assurance components have the same level in both documents (PP [20] and Security Target). The following two subsections apply the refinements to ALC_CMS.5 and ADV_FSP.5, which are different between the PP [20] and the Security Target.

SAR	Title	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	

SAR	Title	Note
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[1]	

Tab. 6.34: 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 [20] 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 [20] 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 [20] regarding ADV_FSP.4 is concerned with the complete representation of 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 [20] and is not cited here.

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

Section 6.3.1 in [20] provides a rationale for the mapping between security functional requirements and security objectives defined in the PP [20]. 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 resp. development environment.

[1] According to the Application Note 30 in [20] the Security Target should indicate the version of the document Supporting Document Mandatory Technical Document Application of Attack Potential to Smart Cards [21] used for the vulnerability analysis.

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

Tab. 6.35: Security Functional Requirements vs. Security Objectives (PP)

The Security Target additionally defines the SFRs for the TOE that are listed in Table 6.36. 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.HW_DES3	FCS_COP.1[HW_DES]
O.HW_AES	FCS_COP.1[HW_AES]
O.INTEGRITY_CHK	FDP_ITT.1[HW] FPT_ITT.1[HW]

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

Tab. 6.36: Security Functional Requirements vs. Security Objectives (ST)

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

Justification related to O.HW_DES3:

SFR	Rationale
FCS_COP.1[HW_DES]	This SFR requires the TOE to support Triple DES encryption and decryption of one block as required by the objective.

Justification related to O.HW_AES:

SFR	Rationale
FCS_COP.1[HW_AES]	This SFR requires the TOE to support AES encryption and decryption of one block as required by the objective.

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.CUST_RECONFIG:

SFR	Rationale
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.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.
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.Self-Test:

SFR	Rationale
FPT_TST.1	This SFR requires self-testing of the TOE during start-up and some self testing functionality provided 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.

6.3.2 Dependencies of Security Functional Requirements

The dependencies listed in the PP [20] are independent of the additional dependencies listed in the table below. The dependencies of the PP [20] are fulfilled within the PP [20] 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
FMT_LIM.1[HW]	FMT_LIM.2 Limited availability.	Yes
FMT_LIM.2[HW]	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. 6.46: 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.	See FDP_ACF.1[MEM].

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FDP_ACC.1[SFR]	FDP_ACF.1 Security attribute based access control.	See FDP_ACF.1[SFR].
FDP_ACF.1[MEM]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	See FDP_ACC.1[MEM], FMT_MSA.3[MEM].
FDP_ACF.1[SFR]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	See FDP_ACC.1[SFR], FMT_MSA.3[SFR].
FDP_SDI.2[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	See 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	See 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	See FMT_MSA.1[MEM]. For FMT_SMR.1, see discussion below.
FMT_MSA.3[SFR]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	See 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. 6.47: Dependencies of Security Functional Requirements (ST)

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] resp. FMT_MSA.1[SFR] and FMT_MSA.3[MEM] resp. 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 [20]. 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 [20] 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 [20], 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 [20] 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 and O.Leak-Forced 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] 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] and of FDP_ACC.1[MEM] resp. FDP_ACC.1[SFR] with FDP_ACF.1[MEM] resp. 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 behavior 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.

Note: Parts of the security functionality are configured at the end of phase 3 and all security functionality is active after phase 3 or phase 4 depending on the delivery form.

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, e.g. the PKCC for asymmetric cryptographic algorithms.

7.1.1 Security Services

Tables 7.1 (for PP) and 7.2 (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.HW_DES3	Triple-DES Operations
SS.HW_AES	AES Operations
SS.SELF_TEST	Self Test
SS.RESET	Reset Functionality
SS.RECONFIG	Post Delivery Configuration

Tab. 7.2: Security Services defined in the extended scope of this Security Target

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]. Note that statistical tests are requested from the Security IC Embedded Software (refer to [9]). This means that the Random Number Generator together with the according online test features to guarantee its correct operation and quality of randomness provided by the Security IC Embedded Software is suitable for generation of signature key pairs, generation of session keys for symmetric encryption mechanisms, random padding bits, zero-knowledge proofs and the generation of seeds for DRNGs.

SS.HW_DES3 **Triple-DES Operations**

SS.HW_DES3 provides DES encryption and decryption based on 112 bit and 168 bit keys.

The TOE provides the Single DES according to the Data Encryption Standard (DES). SS.HW_DES3 is a modular basic cryptographic function, which provides the TDEA as defined by FIPS PUB 46 [8] by means of a hardware coprocessor which provides Single-DES. The document [9] provides guidance how to use the hardware coprocessor such that (a) the 3-key Triple-DEA according to keying option 1 and (b) the 2-key Triple-DEA according to keying option 1 and 2 in FIPS PUB 46-3 [8] can be implemented by the Security IC Embedded Software. Also the key management for the 2-key (112 bit) Triple-DEA shall be provided by the Security IC Embedded Software. For encryption the Security IC Embedded Software provides 8 bytes of the plain text and SS.HW_DES3 calculates 8 bytes cipher text. The calculation output is read by the Security IC Embedded Software. For decryption the Security IC Embedded Software provides 8 bytes of cipher text and SS.HW_DES3 calculates 8 bytes plain text. The calculation output is read by the Security IC Embedded Software.

SS.HW_AES **AES Operations**

SS.HW_AES provides AES encryption and decryption based on 128 bit keys.

The TOE provides the Advanced Encryption Standard (AES) algorithm according to the Advanced Encryption Standard as defined by FIPS PUB 197 [7]. SS.HW_AES is a modular basic cryptographic function, which provides the AES algorithm by means of a hardware coprocessor and supports the AES algorithm with key length of 128, 192 and 256 bit. The key management for the AES algorithm shall be provided by the Security IC Embedded Software. For encryption the Security IC Embedded Software provides 16 bytes of the plain text and SS.HW_AES calculates 16 bytes cipher text. The calculation output is read by the Security IC Embedded Software. For decryption the Security IC Embedded Software provides 16 bytes of cipher text and SS.HW_AES calculates 16 bytes plain text. The calculation output is read by the Security IC Embedded Software.

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 (i.e. ”Power-On Reset”) and a security relevant reset (Security Reset) which can be executed only a limited time before the device is disabled permanently. The IC can also be terminated with one call, where the error counter is set to its end state.

SS.RECONFIG **Post Delivery Configuration**

SS.RECONFIG realizes the Post Delivery Configuration. These can be used by the customer to set the accessible size of the EEPROM, enable or disable the PKCC co-processor, the AES co-processor, the DES co-processor, the number of keys in the EEPROM key store and the number of Anti-Tearing pages. The configuration values of the Post Delivery Configuration are stored in a special area in the NXP_ConfigData_Seg.

Note that if the PKCC coprocessor, the AES coprocessor and the DES coprocessor are disabled, both will no longer be available to the Security IC Embedded Software and attempting to use it will raise an exception. This means the availability of SS.HW_AES and SS.HW_DES3 is configurable. The customer can change the values of the Post Delivery Configuration through invoking the Post Delivery Configuration functionality in Boot Software (see SF.MEM_ACC) . This functionality is invoked by using the chip health mode via the ISO/IEC 7816 interface and applying the required Post Delivery Configuration commands.
The customer can change these values as many times as he wishes. However, once he calls the Boot Software using the chip health mode via the ISO/IEC 7816 interface with a certain parameter set to a specific value, the options are locked permanently, and can no longer be changed. The options must be locked before the TOE is delivered to the customer before phase 7 of the life-cycle.

7.1.2 Security Features

Tables 7.3 (for PP) and 7.4 (for ST) list the Security Services 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. 7.3: 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

Tab. 7.4: Security Features defined in the extended scope of this Security Target

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 my 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 triggers 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 in 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 ROM and

(iv) the application data in the RAM and EEPROM 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 my 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 resp. 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 via e.g. 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 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. w.r.t. 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 (e.g. 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 (NXP_ConfigData_Seg), 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. For the Security IC Embedded Software, User Mode and System Mode are available.
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 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.

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.

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	FRU_FLT.2	Limited Fault Tolerance
	FPT_FLS.1	Failure with Preservation of Secure State
SF.PHY	FPT_PHP.3	Resistance to Physical Attack
	FDP_ITT.1[HW]	Basic Internal Transfer Protection
	FPT_ITT.1[HW]	Basic Internal TSF Data Transfer Protection
	FMT_SMF.1[HW]	Specification of Management Functions (Hardware)
	FDP_SDI.2[HW]	Stored Data Integrity Monitoring and Action
SF.LOG	FDP_ITT.1[HW]	Basic Internal Transfer Protection
	FPT_ITT.1[HW]	Basic Internal TSF Data Transfer Protection
	FDP_IFC.1	Subset Information Flow Control
SF.COMP	FMT_LIM.1[HW]	Limited Capabilities
	FMT_LIM.2[HW]	Limited Availability
	FAU_SAS.1[HW]	Audit Storage

Tab. 7.5: TOE Security Functionality vs. Security Functional Requirements (PP0035)

TSF	SFR	Title
SS.HW_DES3	FCS_COP.1[HW_DES]	Cryptographic Operation (DES)
SS.HW_AES	FCS_COP.1[HW_AES]	Cryptographic Operation (AES)
SS.SELF_TEST	FPT_TST.1	TSF Testing
SS.RESET	FMT_SMF.1[SW]	Specification of Management Functions (Software)
SS.RECONFIG	FMT_SMF.1[HW]	Specification of Management Functions (Hardware)
SF.MEM_ACC	FMT_LIM.2[HW]	Limited Availability
	FDP_ACC.1[MEM]	Subset Access Control (Memories)

TSF	SFR	Title
	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)

Tab. 7.6: TOE Security Functionality vs. Security Functional Requirements

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 feature.

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

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

[9] Guidance and Operation Manual NXP Secure Smart Card Controller P40C012/040/072, Information on Guidance and Operation, NXP Semiconductors, Document number 269422, 2014-06-27.

[10] ISO/IEC 7816-2:1996 Information technology - Identification cards - Integrated circuit(s) cards with contacts, Part 2: Dimensions and location of contacts.

[11] Order Entry Form P40C012/040/072, NXP Semiconductors, online document, Business Unit Identification.

[12] Product data sheet addendum SmartMX2 P40 family P40Cxxx, Chip Health Mode, NXP Semiconductors, Document number 269720, 2014-05-21.

[13] Product data sheet addendum SmartMX2 P40 family P40Cxxx, Firmware interface specification, NXP Semiconductors, Document number 275823, 2014-06-27.

[14] Product data sheet addendum SmartMX2 P40 family P40Cxxx, Instruction Set Manual, NXP Semiconductors, Document number 258121, 2014-06-27.

[15] Product data sheet addendum SmartMX2 P40 family P40Cxxx, Post Delivery Configuration, NXP Semiconductors, Document number 269620, 2013-05-21.

[16] Product data sheet addendum SmartMX2 P40 family P40Cxxx, System Mode, NXP Semiconductors, Document number 267522, 2014-06-27.

[17] Product data sheet addendum SmartMX2 P40 family P40Cxxx, User Mode, NXP Semiconductors, Document number 275722, 2014-06-27.

[18] Product data sheet addendum SmartMX2 P40 family P40Cxxx VA and VD, Wafer specification, NXP Semiconductors, Document number 269822, 2014-06-03.

[19] Product data sheet SmartMX2 P40 family P40C012/040/072, Secure high-performance smart card controller, NXP Semiconductors, Document number 262923, 2014-06-27.

[20] Security IC Platform Protection Profile, registered and certified by Bundesamt fuer Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0035, Rev 1.0, 15 June 2007.

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

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: 2014-09-25

Document identifier: NSCIB-CC-13-37658