NXP Secure Smart Card Controller P40C008/012/024/040/072 VE.001

Security Target Lite

Rev. 3.0 — 14 December 2020 **Evaluation document**
NSCIB-CC-0262848 **CC CONFIDENTIAL**

Document information

Information	Content
Keywords	CC, Security Target Lite, P40C008/012/024/040/072 VE.001
Abstract	Security Target Lite of the NXP Secure Smart Card Controller P40C008/012/024/040/072 VE.001, which is developed and provided by NXP Semiconductors, BL C&S, according to the Common Criteria for Information Technology Security Evaluation Version 3.1 at EAL5 augmented

Revision History

Rev.	Date	Description
3.0	2020-12-14	Based on full Security Target v3.1

1 Introduction

1.1 ST Reference

NXP Secure Smart Card Controller P40C008/012/024/040/072 VE.001, Security Target Lite, Version 3.0, NXP Semiconductors, 14 December 2020.

1.2 TOE Reference

NXP Secure Smart Card Controller P40C008/012/024/040/072 VE.001.

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 P40C008/012/024/040/072 VE.001 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 the TOE 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.

P40C008/012/024/040/072 VE.001 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, P40C008/012/024/040/072 VE.001 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 P40C008/012/024/040/072 VE.001 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. **Note:** The AES coprocessor is outside the scope of the current evaluation.
- 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 P40C008/012/024/040/072 VE.001 is provided as an IC hardware platform with IC Dedicated Software for various operating systems and applications with high security requirements.

1.3.3 Required non-TOE Hardware/Software/Firmware

None.

1.4 TOE Description

1.4.1 Physical Scope of TOE

P40C008/012/024/040/072 VE.001 is manufactured in an advanced 90nm CMOS technology. A block diagram of the IC hardware is depicted below.

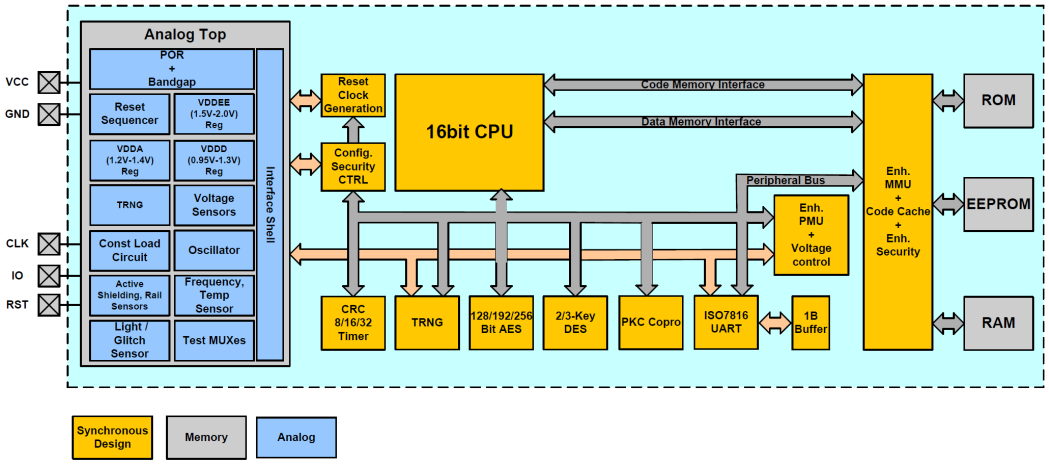

Figure 1. TOE hardware IC block diagram

P40C008/012/024/040/072 VE.001 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

Table 1. Components of the TOE

Type	Name	Release	Form of delivery
IC Hardware	P40C008/012/024/040/072 VE	VE	wafer, module (dice have nameplate 9511E)
IC Dedicated Test Software	Test Software	01h	On-chip software (ROM)
IC Dedicated Support Software	Boot Software	01h	On-chip software (ROM)
	HAL Software	01h	On-chip software (ROM)
Document	Product data sheet SmartMX2 P40 family P40C008/012/024/040/072, Secure high-performance smart card controller, NXP Semiconductors [7]	262936	Electronic document (PDF via NXP DocStore)
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Firmware interface specification, NXP Semiconductors [8]	275836	Electronic document (PDF via NXP DocStore)
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, User Mode, NXP Semiconductors [9]	275733	Electronic document (PDF via NXP DocStore)
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, System Mode, NXP Semiconductors [10]	267531	Electronic document (PDF via NXP DocStore)

Table 1. Components of the TOE*...continued*

Type	Name	Release	Form of delivery
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Chip Health Mode, NXP Semiconductors [11]	269730	Electronic document (PDF via NXP DocStore)
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Post Delivery Configuration, NXP Semiconductors [12]	269630	Electronic document (PDF via NXP DocStore)
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx, Instruction Set Manual, NXP Semiconductors [13]	258132	Electronic document (PDF via NXP DocStore)
Document	Product data sheet addendum SmartMX2 P40 family P40Cxxx VA, VD and VE, Wafer specification, NXP Semiconductors [14]	269832	Electronic document (PDF via NXP DocStore)
Document	Guidance and Operation Manual NXP Secure Smart Card Controller P40C008/012/024/040/072, Information on Guidance and Operation, NXP Semiconductors [15]	269433	Electronic document (PDF via NXP DocStore)

The TOE (hardware) is shipped to the customer by NXP. The available documentation can be downloaded by customers in PDF format directly from the NXP DocStore.

The IC Hardware is identified by its nameplate VE, that is located in the layout of the chip (see [14] 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 [8]. The NXP ROM image 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 [15] together with a hash-value for identification as evaluated versions.

1.4.2 Evaluated Configurations

The customer selects logical and physical configuration options of the TOE 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

Five major configurations are present with respect to the EEPROM size, which are denoted by the names P40C008 VE.001, P40C012 VE.001, P40C024 VE.001, P40C040 VE.001 and P40C072 VE.001. 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 during ordering.

1.4.2.2 Minor configuration options

Minor configurations are chosen by the customer during ordering as detailed in the following table.

Table 2. Evaluated minor configuration options

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.
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.

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 options for the TOE are listed in the following table.

Table 3. Post Delivery Configuration options

PDC option	Description
DES	Disable the DES coprocessor if enabled during ordering.
AES	Disable the AES coprocessor if enabled during ordering.
PKCC	Disable the PKC coprocessor if enabled during ordering.
PDC	Disable the PDC if enabled during ordering.
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.

Table 3. Post Delivery Configuration options*...continued*

PDC option	Description
NumATPages	Reduce the number of available Anti tearing pages.

As indicated in the description of the PDC options, they can only be used to downgrade some configurations. For instance, if DES is enabled as a minor configuration option during ordering, 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 [12].

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 4 and Table 5 for to retrieve a commercial type name. The commercial type name is composed of fixed symbols, which are detailed in Table 4, and variable entries, which are filled in according to the rules in Table 5.

Table 4. Variable Definitions for Commercial Type Names

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	’008’ for 8KB, ’012’ for 12KB, ’024’ for 24KB, ’040’ for 40KB, ’072’ for 72KB
pp(p)	Package delivery type	alpha numeric	see Table 5
/	separator (mandatory)		
m	Manufacturer identifier	alpha numeric	’9’ for TSMC-Fab9
v	Version of mask set	alphabetic	’E’ for HW version VE
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

Table 5. Supported Package Types

Type	Description
U*x*	Wafer not thinner than 50 µm (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)

For example, commercial type name P40C072X60/9Errff denotes major configuration P40C072 VE 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 [8].

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 during ordering and are 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 [12].

1.4.3 Logical Scope of TOE

1.4.3.1 Hardware Description

The TOE distinguishes three TOE modes:

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

The Super System Mode is not available to the Security IC Embedded Software. In Super System Mode the TOE executes the Boot Software and the IC Dedicated Test Software. Notice that parts of the HAL Software execute also in Super System Mode and other parts are executed in System Mode and can be accessed via so-called system calls either from User Mode or System Mode. 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 modelled as an own mode (in that context sometimes referred to as 'Test Mode') that has extended access rights compared to System Mode.

The TOE 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 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 TOE 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, The TOE 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 TOE 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. **Note:** The AES coprocessor is outside the scope of the current evaluation. 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 TOE 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 TOE 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 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 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 2; ”System Mode Customer Code”). Both are not part of the TOE.

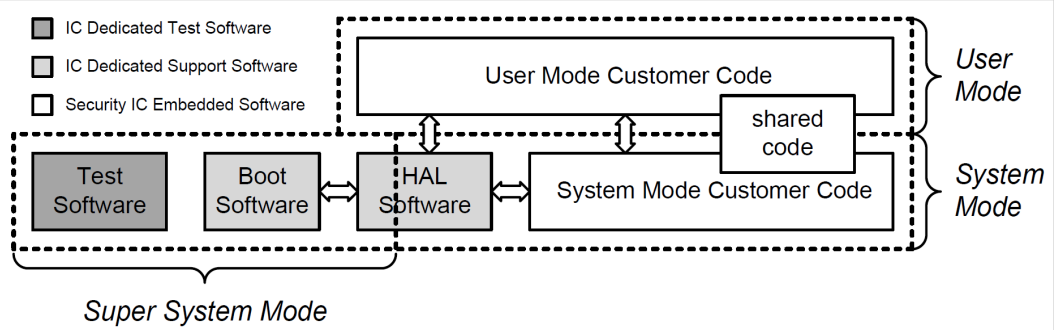

Figure 2. TOE software components

The IC Dedicated Software comprises the IC Dedicated Test Software and the IC Dedicated Support Software described in the following.

The IC Dedicated Test Software is developed by NXP and embedded in the Test Software. The IC Dedicated Test Software includes the test operating system, test routines for the various blocks of the circuitry, control flags for the status of the EEPROM’s manufacturer area and shut down functions to ensure that security relevant test routines cannot be executed illegally after phase 3. This is stored in the NXP_ConfigData_Seg. Moreover, the IC Dedicated Test Software is used to download patch code related to System Mode (stored in SM_PatchCode_Seg) 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 BootTestCode_Seg.
2. The HAL Software is partly stored in the BootTestCode_Seg and partly stored in the SM_Code_Seg and accessed by the Security IC Embedded Software via system calls. It provides basic NVM access, 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 documentation delivered with the TOE contain a functional description and guidelines for the use of the security functionality, as needed to develop Security IC Embedded Software. The documentation provided with the TOE is listed in Table 1. 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 [6]. 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 [6] 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 [7] and the guidance and operation manual [15].

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 nonfunctional 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 [6]. 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 P40C008/012/024/040/072 VE.001 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 enduser 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. 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 P40C008/012/024/040/072 VE.001 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 behaviour the logical interface is defined by the Security IC Embedded Software.

2 Conformance Claims

2.1 CC Conformance Claim

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, Revision 5, CCMB-2017-04-001, April 2017 [2].
- Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, CCMB-2017-04-002, April 2017 [3].
- Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, CCMB-2017-04-003, April 2017 [4].

For the evaluation the following methodology will be used:

- Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, CCMB-2017-04-004, April 2017 [5].

2.2 Package Claim

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

Note: The Protection Profile (PP) ”Security IC Platform Protection Profile with Augmentation Packages” [6] to which this Security Target claims conformance (refer to Section 2.3) 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.

2.3 PP Claim

This Security Target claims strict conformance to the Security IC Platform Protection Profile with Augmentation Packages [6]. 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 Protection Profile (PP) ”Security IC Platform Protection Profile with Augmentation Packages” [20] defines ”Augmentation Packages”. This Security Target does not use any of the packages defined in the PP [20]. The TOE provides additional functionality, which is not covered in [6]. In accordance with Application Note 4 of [6] this additional functionality is added using the policy P.Add-Components (see Section 3.3).

2.4 Conformance Claim Rationale

As the Protection Profile [6] requires strict conformance, no conformance claim requirement is needed in this Security Target.

3 Security Problem Definition

This section lists the assets, threats, organisational security policies and assumptions from the Protection Profile [6] and describes extensions to these elements in detail.

3.1 Description of Assets

The assets to be protected (related to standard functionality) are described in Section 3.1 of the Protection Profile [6] and are listed below:

- The user data of the Composite TOE.
- The Security IC Embedded Software, stored and in operation.
- The security services provided by the TOE for the Security IC Embedded Software.

These assets are related to the following high-level security concerns:

- Integrity of user data of the Composite TOE.
- Confidentiality of user data of the Composite TOE being stored in the TOE's protected memory areas.
- Correct operation of the security services provided by the TOE for the Security IC Embedded Software.
- Deficiency of random numbers.

To be able to protect these assets the TOE shall self-protect its security functionality. Critical information about the security functionality shall be protected by the development environment and the operational environment. Critical information may include:

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

For details see Section 3.1 of the Protection Profile [6].

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

3.2 Threats

All threats for the TOE which are defined in section 3.2 of the Protection Profile are applied to this Security Target and are listed in Table 6.

Table 6. Threats defined in the Protection Profile (PP-0084)

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

For details see Section 3.2 of the Protection Profile [6].

The TOE provides additional functionality to protect against threats that appear when the TOE is used for multiple applications. The TOE provides the Security IC Embedded Software running in System Mode with control of access to memories and hardware components by different applications running in User Mode. In this context, User Data of different applications is stored to such memory and processed by such hardware components. The Security IC Embedded Software controls all this User Data. Any access to User Data assigned to one application by another application contradicts separation between different applications and is considered as a threat.

The following additional threat is therefore defined in this Security Target:

Table 7. Additional threat defined in this Security Target

Name	Title
T.Unauthorised-Access	Unauthorised Data Modification

T.Unauthorised-Access	**Unauthorised Memory or Hardware Access** 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.

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.

3.3 Organisational Security Policies

All organisational security policies defined in the Protection Profile are valid for this Security Target and are listed in Table 8. For details see Section 3.3 of the Protection Profile [6].

Table 8. Organisational security policies defined in the Protection Profile (PP-0084)

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

This Security Target defines one additional organisational security policy as detailed in the following.

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.

Table 9. Additional organisational security policy defined in this Security Target

Name	Title
P.Add-Components	Additional Specific Security Components

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
- 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

3.4 Assumptions

All assumptions defined in the Protection Profile are valid for this Security Target and are listed in Table 10. For details see Section 3.4 of the Protection Profile [6].

Table 10. Assumptions defined in the Protection Profile (PP-0084)

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

In compliance with Application Notes 6 and 7 in the Protection Profile [6], this Security Target defines two additional assumptions as follows:

Table 11. Additional assumptions defined in this Security Target

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

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 [6].

A.Key-Function — **Usage of Key-dependent Functions**

Key-dependent functions (if any) shall be implemented in the Security IC Embedded Software in a way that they are not susceptible to leakage attacks (as described

under T.Leak-Inherent and T.Leak-Forced). Note that here the routines which may compromise keys when being executed are part of the Security IC Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines which are part of the TOE and (ii) the processing of User Data including cryptographic keys.

4 Security Objectives

4.1 Security Objectives for the TOE

All security ojectives for the TOE which are defined in section 4.1 of the Protection Profile are applied to this Security Target and are listed in Table 12.

Table 12. Security Objectives of the TOE (PP-0084)

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

Regarding the Application Notes 8 and 9 in the Protection Profile [6], additional security objectives that are based on additional functionality provided by the TOE are defined below:

Table 13. Additional security objectives defined in this Security Target

Name	Title
O.HW_DES3	Triple DES Encryption
O.INTEGRITY_CHK	Integrity Control of Transferred Data
O.NVM_INTEGRITY	Integrity Support of data stored to NVM
O.MEM_ACCESS	Area based Memory Access Control
O.SFR_ACCESS	Special Function Register Access Control
O.Self-Test	Self Test
O.Reset	Reset function
O.CUST_RECONFIG	Post Delivery Configuration

O.HW_DES3 — **Triple DES Encryption**

The TOE shall 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.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.

4.2 Security Objectives for the Security IC Embedded Software

All security objectives for the Security IC Embedded Software which are defined in section 4.2 of the Protection Profile are applied to this Security Target and are listed in Table 14.

Table 14. Security Objectives for the Security IC Embedded Software (PP-0084)

Name	Title
OE.Resp-Appl	Treatment of User Data

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

All security objectives for the operational environment which are defined in section 4.3 of the Protection Profile are applied to this Security Target and are listed in Table 15.

Table 15. Security Objectives for the Operational Environment (PP-0084)

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

The following additional security objective for the operational environment is defined in this Security Target:

Table 16. Additional security objective for the operational environment defined in this Security Target

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

This additional security objective for the operational environment derives from assumption A.Check-Init. The TOE provides specific functionality that requires the TOE Manufacturer to implement measures for unique identification

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

OE.Check-Init — **Check of initialization data by the Security IC Embedded Software**

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

4.4 Security Objectives Rationale

Section 4.4 in the Protection Profile [6] provides a rationale how the threats, organisational security policies and assumptions are addressed by the security objectives defined in the Protection Profile. The following table 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 Protection Profile [6].

Table 17. Security Objectives (PP and ST) vs. Security Problem Definition (PP)

Security Problem Definition	Security Objective
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
A.Process-Sec-IC	OE.Process-Sec-IC
A.Resp-Appl	OE.Resp-Appl

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

Table 18. Security Objectives (PP and ST) vs. Security Problem Definition (ST)

Security Problem Definition	Security Objective
T.Unauthorised-Access	O.MEM_ACCESS O.SFR_ACCESS
P.Add-Components	O.HW_DES3 O.Self-Test O.Reset O.CUST_RECONFIG O.NVM_INTEGRITY
A.Check-Init	OE.Check-Init
A.Key-Function	OE.Resp-Appl

The rationale for the mapping is given below:

Justification related to T.Unauthorised-Access:

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

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

Justification related to P.Add-Components:

Objective	Rationale
O.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.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.

Justification related to A.Check-Init:

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

Justification related to A.Key-Function:

Objective	Rationale
OE.Resp-Appl	The definition of this objective of the PP [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 by this objective.

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

5 Extended Components Definition

This Security Target does not define extended components.

Note that the Security IC Platform Protection Profile with Augmentation Packages [6] defines extended security functional requirements FCS_RNG.1, FMT_LIM.1, FMT_LIM.2, FAU_SAS.1 and FDP_SDC.1 in chapter 5, which are included in this Security Target.

6 Security Requirements

This chapter defines the security requirements that shall be met by the TOE. These security requirements are composed of the security functional requirements and the security assurance requirements that the TOE must meet in order to achieve its security objectives. CC allows several operations to be performed on security requirements (on the component level); refinement, selection, assignment, and iteration are defined in section 8.1 of CC Part 1 [2]. These operations are used in the PP [6] 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 [6] 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 [6] 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 [6] and this Security Target. Table 19 and Table 20 summarize the SFRs defined in the PP and ST, respectively.

Table 19. SFRs defined in the Security IC Protection Profile

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

Table 19. SFRs defined in the Security IC Protection Profile...*continued*

Name	Title
FPT_PHP.3	Resistance to Physical Attack
FRU_FLT.2	Limited Fault Tolerance

Table 20. SFRs defined in this Security Target

Name	Title
FCS_COP.1[HW_DES]	Cryptographic Operation (DES)
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)
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

6.1.1 SFRs of the Protection Profile

All SFRs, which are defined in the PP [6], are summarized in Table 19. Some of these SFRs are defined in CC Part 2 [3] and eventually subject to refinement, selection, assignment and/or iteration operation in the PP [6]. Others are newly defined in the PP [6].

6.1.1.1 FDP_ITT.1[HW]

The TOE shall meet the requirement "Basic internal transfer protection" 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*[1] to prevent the *disclosure and modification*[2] of user data when it is transmitted between physically-separated parts of the TOE.

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

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.

6.1.1.2 FPT_ITT.1[HW]

The TOE shall meet the requirement "Basic internal TSF data transfer protection" 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*[3] 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.

6.1.1.3 FAU_SAS.1[HW]

The TOE shall meet the requirement "Audit storage" as defined in the PP [6], and as specified below.

FAU_SAS.1[HW]	**Audit storage**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FAU_SAS.1.1[HW]	The TSF shall provide *the test process before TOE Delivery* with the capability to store *the Initialisation Data and/or Pre-personalisation Data and/or supplements of the Security IC Embedded Software*[4] in the *NVM*[5].

6.1.1.4 FDP_SDC.1[HW]

The TOE shall meet the requirement "Stored data confidentiality" as defined in the PP [6], and as specified below.

FDP_SDC.1[HW]	**Stored data confidentiality**

2 [selection: *disclosure, modification, loss of use*]
3 [[selection: *disclosure, modification*]]
4 [selection: *the Initialisation Data, Pre-personalisation Data, [assignment: other data]*]
5 [assignment: *type of persistent memory*]

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SDC.1.1[HW]	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the *RAM and EEPROM*[6].

6.1.1.5 FDP_SDI.2[HW]

The TOE shall meet the requirement "Stored data integrity monitoring and action" as defined in the PP [6], and as specified below.

FDP_SDI.2[HW]	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1[HW]	The TSF shall monitor user data stored in containers controlled by the TSF for *modification, deletion, repetition or loss of data*[7] on all objects, based on the following attributes: *integrity check information associated with the data storied in memories*[8].
FDP_SDI.2.2[HW]	Upon detection of a data integrity error, the TSF shall *perform an error correction if possible or trigger a Security Reset if not*[9].

6.1.1.6 FCS_RNG.1[HW]

The TOE shall meet the requirement "Random number generation (Class PTG.2)" as defined in the PP [6] according to [1] , and as specified below.

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*[10] random number generator that implements:[11] (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.

6 [assignment: *memory area*]
7 [assignment: *integrity errors*]
8 [assignment: *user data attributes*]
9 [assignment: *action to be taken*]
10 [selection: *physical, hybrid physical, hybrid deterministic*]
11 [assignment: *list of security capabilities*]

(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*[12].

(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*[13]. 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*[14] that meet:

(PTG.2.6) Test procedure A[15] 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.

6.1.2 Additional SFRs regarding Cryptographic Support

6.1.2.1 FCS_COP.1[HW_DES]

The TOE shall meet the requirement "Cryptographic Operation (DES)" 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

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

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

14 [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*]

15 [assignment: *additional standard test suites*]. Assignment is empty as per Application Note 44 of the PP.

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*[16] in accordance with the specified cryptographic algorithm *Triple Data Encryption Algorithm (TDEA)*[17] and cryptographic key sizes *112 or 168 bit*[18] that meet the following:[19]

- *FIPS PUB 46-3* [16]*, keying options 1 and 2.*

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.

6.1.3 Additional SFRs regarding Protection of TSF

6.1.3.1 FPT_TST.1

The TOE shall meet the requirement "TSF testing" 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*[20] to demonstrate the correct operation of *(i) the active shielding, (ii) the sensors*[21].
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*[22].
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of *stored TSF executable code*[23].

16 [assignment: *list of cryptographic operations*]
17 [assignment: *cryptographic algorithm*]
18 [assignment: *cryptographic key sizes*]
19 [assignment: *list of standards*]
20 [selection: *during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]*]
21 [selection: *[assignment: parts of TSF], the TSF*]
22 [selection: *[assignment: parts of TSF data], TSF data*]
23 [selection: *[assignment: parts of TSF], TSF*]

6.1.4 Additional SFRs regarding Security Management

6.1.4.1 FMT_SMF.1[SW]

The TOE shall meet the requirement "Specification of Management Functions (Software)" 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:[24] • 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

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

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

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.

Object	UM_RAM_Seg	UM RAM Segment
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 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.

Object	UM_ROMConst_Seg	UM ROM Constant Segment
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 read and execute access via SFR_ AccCtrlCfg.

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

Object	UM_Code_Seg	UM Code Segment
Attribute	enable	Enable read and execute access via SFR_ AccCtrlCfg.

Object	RAM_CodeMirror_Seg	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: • 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

6.1.6.5.1 FDP_ACC.1[MEM]

The TOE shall meet the requirement "Subset access control (Memories)" 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*[25] on *all code running on the TOE, all memories and all memory operations*[26].
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.

6.1.6.5.2 FDP_ACC.1[SFR]

The TOE shall meet the requirement "Subset access control (Special Function Registers" as specified below.

FDP_ACC.1[SFR]	**Subset access control (Special Function Registers**

25 [assignment: *access control SFP*]
26 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]

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*[27] on *all code running on the TOE, all Special Function Registers and all Special Function Register operations*[28].
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 nonexisting Special Function Registers is treated as a security violation and will trigger a Security Reset.

6.1.6.5.3 FDP_ACF.1[MEM]

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

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 initialisation
FDP_ACF.1.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy*[29] to objects based on the following: *all subjects and objects and the attributes themselves defined as the objects SFR_SysMgmt and SFR_MemSegCfg*[30].
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:[31] 1. *The SSM_Code is allowed to perform SM_ROMConst_Seg.read.* 2. *The SSM_Code is allowed to perform UM_ROMConst_Seg.read.* 3. *The SSM_Code is allowed to perform SharedConst_Seg.read.* 4. *The SSM_Code is allowed to perform XCall_Table_Seg.execute.*

27 [assignment: *access control SFP*]
28 [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*]
29 [assignment: *access control SFP*]
30 [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]
31 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]

5. *The SSM_Code is allowed to perform SM_Code_Seg.read and SM_Code_Seg.execute.*
6. *The SSM_Code is allowed to perform SharedCode_Seg.execute.*
7. *The SSM_Code is allowed to perform EE_UserData_Seg.read and EE_UserData_Seg.write.*
8. *The SSM_Code is allowed to perform SM_RAM_Seg.read and SM_RAM_Seg.write.*
9. *The SSM_Code is allowed to perform UM_RAM_Seg.read and UM_RAM_Seg.write.*
10. *The SM_Code is allowed to perform SM_ROMConst_Seg.read.*
11. *The SM_Code is allowed to perform UM_ROMConst_Seg.read.*
12. *The SM_Code is allowed to perform SharedConst_Seg.read.*
13. *The SM_Code is allowed to perform XCall_Table_Seg.execute.*
14. *The SM_Code is allowed to perform SM_Code_Seg.read and SM_Code_Seg.execute.*
15. *The SM_Code is allowed to perform SharedCode_Seg.execute.*
16. *The SM_Code is allowed to perform EE_UserData_Seg.read and EE_UserData_Seg.write.*
17. *The SM_Code is allowed to perform SM_RAM_Seg.read and SM_RAM_Seg.write.*
18. *The SM_Code is allowed to perform UM_RAM_Seg.read and UM_RAM_Seg.write.*
19. *The UM_Code is allowed to perform UM_ROMConst_Seg.read.*
20. *The UM_Code is allowed to perform SharedConst_Seg.read.*
21. *The UM_Code is allowed to perform UM_Code_Seg.execute.*
22. *The UM_Code is allowed to perform SharedCode_Seg.execute.*
23. *The UM_Code is allowed to perform EE_UserData_Seg.read.*
24. *The UM_Code is allowed to perform UM_RAM_Seg.read and UM_RAM_Seg.write.*
25. *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 authorise access of subjects to objects based on the following additional rules:[32]

1. *The SSM_Code is allowed to perform ROM_Mirror_Seg.read if the attribute ROM_Mirror_Seg.enable grants this right.*
2. *The SSM_Code is allowed to perform BootTestCode_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*

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

3. *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.*
4. *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.*
5. *The SSM_Code is allowed to perform SharedCode_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*
6. *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.*
7. *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.*
8. *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.*
9. *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.*
10. *The SSM_Code is allowed to perform SM_PatchCode_Seg.execute if the attribute SM_PatchCode_Seg.enable grants this right.*
11. *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.*
12. *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.*
13. *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.*
14. *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.*
15. *The SSM_Code is allowed to perform RAM_CodeMirror_Seg.execute if the attribute RAM_CodeMirror_Seg.enable grants this right.*
16. *The SM_Code is allowed to perform ROM_Mirror_Seg.read if the attribute ROM_Mirror_Seg.enable grants this right.*
17. *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.*
18. *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.*
19. *The SM_Code is allowed to perform SharedCode_Seg.read via ROM_Mirror_Seg if the attribute ROM_Mirror_Seg.enable grants this right.*

20. *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.*
21. *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.*
22. *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.*
23. *The SM_Code is allowed to perform SM_PatchCode_Seg.execute if the attribute SM_PatchCode_Seg.enable grants this right.*
24. *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.*
25. *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.*
26. *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.*
27. *The UM_Code is allowed to perform EE_UserData_Seg.write if the attribute EE_UserData_Seg.enable grants this right.*
28. *The UM_Code is allowed to perform SM_PatchCode_Seg.execute if the attribute SM_PatchCode_Seg.enable grants this right.*
29. *The UM_Code is allowed to perform UM_PatchCode_Seg.execute if the attribute UM_PatchCode_Seg.enable grants this right.*
30. *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.*
31. *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.*
32. *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 following additional rules: *none*[33].

6.1.6.5.4 FDP_ACF.1[SFR]

The TOE shall meet the requirement "Security attribute based access control (Special Function Registers" as specified below.

FDP_ACF.1[SFR] Security attribute based access control (Special Function Registers

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

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control, FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1[SFR]	The TSF shall enforce the *Hardware Access Control Policy*[34] to objects based on the following: *all subjects and objects and the attributes itself defined as the object SFR_AccCtrlCfg*[35].
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:[36] 1. *The SSM_Code is allowed to perform SFR_SysMgmt.access.* 2. *The SSM_Code is allowed to perform SFR_MemSegCfg.access.* 3. *The SSM_Code is allowed to perform SFR_Testing.access.* 4. *The SSM_Code is allowed to perform SFR_HWComp.access.* 5. *The SM_Code is allowed to perform SFR_SysMgmt.access.* 6. *The SM_Code is allowed to perform SFR_MemSegCfg.access.* 7. *The SM_Code is allowed to perform SFR_HWComp.access.* 8. *The UM_Code is allowed to perform SFR_MemSegCfg.access.* 9. *The UM_Code is allowed to perform SFR_HWComp.access*.
FDP_ACF.1.3[SFR]	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:[37] 1. *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 following additional rules:[38] 1. *The UM_Code is not allowed to access any Special Function Register related to User Mode if SFR_AccCtrlCfg grants this right.* 2. *The UM_Code is not allowed to perform SFR_HWComp.read for Special Function Registers SFR.CRC_DATAH, SFR.CRC_DATAL.* 3. *The UM_Code is not allowed to perform SFR_HWComp.read for Special Function Registers used as DES key registers.*

34 [assignment: *access control SFP*]
35 [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]
36 [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*]
37 [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*]
38 [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*]

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.

6.1.6.6.1 FMT_MSA.3[MEM]

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

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*[39] to provide *restrictive*[40] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[MEM]	The TSF shall allow the *no subject*[41] 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.

6.1.6.6.2 FMT_MSA.3[SFR]

The TOE shall meet the requirement "Static attribute initialization (Special Function Registers)" as specified below.

FMT_MSA.3[SFR]	**Static attribute initialization (Special Function Registers)**
Hierarchical to:	No other components.

39 [assignment: *access control SFP, information flow control SFP*]
40 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
41 [assignment: *the authorised identified roles*]

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*[42] to provide *restrictive*[43] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[SFR]	The TSF shall allow the *no subject*[44] 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.

6.1.6.6.3 FMT_MSA.1[MEM]

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

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*[45] to restrict the ability to *modify*[46] the security attributes *defined as the object SFR_MemSegCfg except DM9_RAM_Seg*[47] to *code executed in System Mode or Super System Mode*[48].

6.1.6.6.4 FMT_MSA.1[SFR]

The TOE shall meet the requirement "Management of security attributes (Special Function Registers)" as specified below.

FMT_MSA.1[SFR]	**Management of security attributes (Special Function Registers)**

42 [assignment: *access control SFP, information flow control SFP*]
43 [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
44 [assignment: *the authorised identified roles*]
45 [assignment: *access control SFP(s), information flow control SFP(s)*]
46 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
47 [assignment: *list of security attributes*]
48 [assignment: *the authorised identified roles*]

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*[49] to restrict the ability to *modify*[50] the security attributes *defined in the Special Function Registers*[51] to *code executed in a TOE mode which has write access to the respective Special Function Registers*[52].

6.1.6.6.5 FMT_SMF.1[HW]

The TOE shall meet the requirement "Specification of Management Functions (Hardware)" as specified below.

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:[53] • *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, DES, PKCC* • *Permanently disabling and enabling of the security functionality EEPROM Size, NumEEKeys, NumATPages, 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.

49 [assignment: *access control SFP(s), information flow control SFP(s)*]
50 [selection: *change_default, query, modify, delete, [assignment: other operations]*]
51 [assignment: *list of security attributes*]
52 [assignment: *the authorised identified roles*]
53 [assignment: *list of management functions to be provided by the TSF*]

6.2 Security Assurance Requirements

The following table lists all security assurance components that are valid for this Security Target.

Table 21. Security Assurance Requirements

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

6.2.1 Refinements of the TOE Security Assurance Requirements

In compliance to Application Note 23 in the PP, this Security Target has to conform to all refinements of the security assurance requirements in the PP. Because the refinements in the PP are defined for the security assurance components of EAL4 (augmented by ALC_DVS.2 and AVA_VAN.5), some refinements have to be applied to assurance components of the higher level EAL5 stated in the Security Target.

Most of the security assurance components mentioned in the PP and in this Security Target have the same component level and therefore for these components the refinements from the PP are valid for this Security Target without change. The following

two subsections apply the refinements to ALC_CMS.5 and ADV_FSP.5, which are different between the PP and this Security Target.

6.2.1.1 Refinements Regarding 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 Protection Profile 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 original component ALC_CMS.4 can be found in section 6.2.1.3 of the Protection Profile and is not repeated here.

6.2.1.2 Refinements regarding 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 Protection Profile 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 and is not cited here.

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

Section 6.3.1 in the Protection Profile provides a rationale for the mapping between security functional requirements and security objectives defined in the Protection Profile. This rationale is not repeated here.

This Security Target defines additional SFRs for the TOE. 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.

Table 22. Security Functional Requirements mapping to Security Objectives

Security Objective	Security Functional Requirement
O.HW_DES3	FCS_COP.1[HW_DES]
O.INTEGRITY_CHK	FDP_ITT.1[HW] FPT_ITT.1[HW]

Table 22. Security Functional Requirements mapping to Security Objectives*...continued*

Security Objective	Security Functional Requirement
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.1[MEM] FMT_MSA.3[MEM] FMT_SMF.1[HW]
O.SFR_ACCESS	FDP_ACC.1[SFR] FDP_ACF.1[SFR] FMT_MSA.1[SFR] FMT_MSA.3[SFR] FMT_SMF.1[HW]
O.Self-Test	FPT_TST.1
O.Reset	FMT_SMF.1[SW]

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.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 Protection Profile are independent of the additional dependencies listed in the table below. The dependencies of the Protection Profile are fulfilled within the Protection Profile and at least one dependency is considered to be satisfied. The following discussion demonstrates how the SFR dependencies (defined by Part 2 of the Common Criteria [3]) satisfy the requirements specified in Section 6.1.

The dependencies and their fullfilment are listed in the tables below:

Table 23. Dependencies of Security Functional Requirements (PP-0084)

SFR	Dependency	Fullfilled in ST
FAU_SAS.1[HW]	No dependencies.	No dependency
FCS_RNG.1[HW]	No dependencies.	No dependency
FDP_ITT.1[HW]	[FDP_ACC.1 Subset access control, or FDP_ IFC.1 Subset information flow control]	Yes
FDP_IFC.1	FDP_IFF.1 Simple security attributes	See discussion in the PP
FDP_SDC.1[HW]	No dependencies.	No dependency
FDP_SDI.2[HW]	No dependencies.	No dependency
FMT_LIM.1[HW]	FMT_LIM.2 Limited availability.	Yes
FMT_LIM.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

Table 24. Dependencies of Security Functional Requirements (Security Target)

SFR	Dependency	Fullfilled in 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.

Table 24. Dependencies of Security Functional Requirements (Security Target)*...continued*

SFR	Dependency	Fullfilled in ST
FDP_ACC.1[MEM]	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1[MEM]
FDP_ACC.1[SFR]	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1[SFR]
FDP_ACF.1[MEM]	FDP_ACC.1 Subset access control FMT_ MSA.3 Static attribute initialization	FDP_ACC.1[MEM] FMT_MSA.3[MEM]
FDP_ACF.1[SFR]	FDP_ACC.1 Subset access control FMT_ MSA.3 Static attribute initialization	FDP_ACC.1[SFR] FMT_MSA.3[SFR]
FMT_MSA.1[MEM]	[FDP_ACC.1 Subset access control, or FDP_ IFC.1 Subset information flow control] FMT_ SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1[MEM] FMT_SMF.1[HW] For FMT_SMR.1, see discussion below.
FMT_MSA.1[SFR]	[FDP_ACC.1 Subset access control, or FDP_ IFC.1 Subset information flow control] FMT_ SMR.1 Security roles FMT_SMF.1 Specification of Management Functions	FDP_ACC.1[SFR] FMT_SMF.1[HW] For FMT_SMR.1, see discussion below.
FMT_MSA.3[MEM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[MEM] For FMT_SMR.1, see discussion below.
FMT_MSA.3[SFR]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SFR] For FMT_SMR.1, see discussion below.
FMT_SMF.1[HW]	No dependencies.	No dependency
FMT_SMF.1[SW]	No dependencies.	No dependency
FPT_TST.1	No dependencies.	No dependency

The developer of the Security IC Embedded Software must ensure that the additional security functional requirements FCS_COP.1[HW_DES] 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] completely address the appropriate management of cryptographic keys used by the specified cryptographic function and the management of access control rights as specified for the memory access control function. All requirements concerning these management functions shall be fulfilled by the environment (Security IC Embedded Software).

The functional requirements [FDP_ITC.1, or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4 are not included in this Security Target since the TOE only provides a pure engine for encryption and decryption without additional features for the handling of cryptographic keys. Therefore the Security IC Embedded Software must fulfill these requirements related to the needs of the realized application.

The dependency FMT_SMR.1 introduced by the two components FMT_MSA.1[MEM] respectively FMT_MSA.1[SFR] and FMT_MSA.3[MEM] respectively FMT_MSA.3[SFR] must be fulfilled by the Security IC Embedded Software. The definition and maintenance of the roles that act on behalf of the functions provided by the hardware must be subject of the Security IC Embedded Software.

6.3.3 Rationale for the Assurance Requirements

The selection of assurance components is based on the underlying PP [6]. 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 [6] 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 [4]. 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 [6], 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 [6] 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] 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] and of FDP_ACC.1[MEM] respectively FDP_ACC.1[SFR] with FDP_ACF.1[MEM] respectively FDP_ACF.1[SFR] as well as the dependent security functional requirements.

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

7 TOE Summary Specification

7.1 Portions of the TOE Security Functionality

The TOE Security Functionality (TSF) directly corresponds to the TOE security functional requirements defined in Section 6. The Security Functionality provided by the TOE is split into Security Services (SS) and Security Features (SF). Both are active and applicable to phases 4 to 7 of the Security IC product life-cycle.

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, for example, the PKCC for asymmetric cryptographic algorithms.

7.1.1 Security Services

7.1.1.1 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 [1]. 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 [15]). 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.

7.1.1.2 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 46-3 [16] by means of a hardware coprocessor which provides Single-DES. The document [15] 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 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.

7.1.1.3 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.

7.1.1.4 SS.RESET Reset Functionality

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

7.1.1.5 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 DES coprocessor, 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 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_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

7.1.2.1 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 Triple-DES co-processor, the memories, registers, I/O interfaces and the other system peripherals) during the execution of the IC Dedicated Support Software and Security IC Embedded Software. This includes all specific security features of the TOE which are able to provide an active response.

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

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

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

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

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

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

7.1.2.2 SF.PHY Protection against Physical Manipulation

The feature SF.PHY protects the TOE against manipulation of

- the hardware
- the IC Dedicated Software in the ROM
- the Security IC Embedded Software in the NVM and
- the application data in the RAM and NVM including the configuration data stored in NXP_ConfigData_Seg.

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

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

- **Layout Protection:** Set of security mechanisms that hamper reverse engineering of the IC, such as layout randomization, active and passive shielding, techniques to hide shielding, multilayer interconnection, wide bus widths and dummy routing.
- **Code- & Datapath Integrity Protection:** Set of security mechanisms that ensure that manipulations on data or code stored and transmitted from memories to the CPU are detected with high probability. This includes integrity protection of the whole code

and data path including CPU internals. Integrity verification is always done before the according data is processed, for example, by an ALU operation.

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

7.1.2.3 SF.LOG Logical Protection

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

The cryptographic coprocessor includes special features to hamper SPA/DPA analysis of shape and amplitude of the power consumption and ensures that the calculation time is independent from any key and plain/cipher text. These include blinding and randomization techniques.

Specific features as described for SF.PHY (for example, the encryption features) and for SF.OPC (e.g. the filter feature) support the logical protection. For instance, the encryption of the whole data and code path including memory and register contents.

7.1.2.4 SF.COMP Protection of Mode Control

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

The control of the TOE modes prevents the abuse of test functions after TOE delivery. Additionally it also ensures that features used during the boot sequence to configure the

TOE cannot be abused. Hardware circuitry and the Boot Software determine whether the test functionality is available or not. If it is available, the TOE starts the IC Dedicated Test Software in the System Mode. Otherwise, the TOE switches to the User Mode or System Mode and starts execution of the Security IC Embedded Software.

The switch to the IC Dedicated Test Software is prevented after TOE delivery because specific electronic fuses guarantee that the IC Dedicated Test Software cannot be selected. The System Mode is the more privileged TOE mode, the User Mode is the less privileged TOE mode. The Boot Software is executed in Super System Mode. HAL Software is executed in Super System Mode (the parts acting as helper function for IC Dedicated Test Software and Boot Software) and System Mode. 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.

7.1.2.5 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. The memories are split into several segments (Objects) for which Operations and Attributes are defined. 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.

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

Table 25. TOE Security Functionality vs. Security Functional Requirements (PP0084)

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

Table 26. TOE Security Functionality vs. Security Functional Requirements

TSF	SFR	Title
SS.HW_DES3	FCS_COP.1[HW_DES]	Cryptographic Operation (DES)
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)
	FDP_ACF.1[MEM]	Security Attribute Based Access Control (Memories)
	FMT_MSA.3[MEM]	Static Attribute Initialization (Memories)

Table 26. TOE Security Functionality vs. Security Functional Requirements *...continued*

TSF	SFR	Title
	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)

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.

The aspects self-protection and non-bypassability are implemented by SF.PHY, SF.OPC, and SF.COMP. SF.PHY covers the physical protection of the TOE and protects the TOE against tampering and bypassing of security features and security services. SF.OPC contributes by covering the aspects failure with preservation of a secure state and limited fault tolerance. This protects the TOE against interference of security feature and security services. SF.COMP limits the capability and availability of the Test Features and protects the TOE against bypassing of security features.

8 Bibliography

8 . 1 Evaluation documents

[1] A proposal for: Functionality classes for random number generators, Wolfgang Killmann, T-Systems GEI GmbH, Werner Schindler, Bundesamt für Sicherheit in der Informationstechnik (BSI), Version 2.0, 18 September 2011.

[2] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, CCMB-2017-04-001, April 2017.

[3] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, CCMB-2017-04-002, April 2017.

[4] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, CCMB-2017-04-003, April 2017.

[5] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, CCMB-2017-04-004, April 2017.

[6] Security IC Platform Protection Profile with Augmentation Packages, Registered and Certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, Version 1.0, 13 January 2014.

8 . 2 Developer documents

[7] SmartMX2 P40 family P40C008/012/024/040/072, Product data sheet, DocStore number 262936, NXP Semiconductors, Revision 3.6, 16 February 2017.

[8] SmartMX2 P40 family P40C008/012/024/040/072, Firmware interface specification, Product data sheet addendum, DocStore number 275836, NXP Semiconductors, Revision 3.6, 10 March 2017.

[9] SmartMX2 P40 family P40C008/012/024/040/072, User Mode, Product data sheet addendum, DocStore number 275733, NXP Semiconductors, Revision 3.3, 17 June 2016.

[10] SmartMX2 P40 family P40C008/012/024/040/072, System Mode, Product data sheet addendum, DocStore number 267531, NXP Semiconductors, Revision 3.1, 17 June 2016.

[11] SmartMX2 P40 family P40C008/012/024/040/072, Chip Health Mode, Product data sheet addendum, DocStore number 269730, NXP Semiconductors, Revision 3.0, 1 April 2015.

[12] SmartMX2 P40 family P40C008/012/024/040/072, Post Delivery Configuration, Product data sheet addendum, DocStore number 269630, NXP Semiconductors, Revision 3.0, 1 April 2015.

[13] SmartMX2 P40 family P40C008/012/024/040/072, Instruction Set Manual, Product data sheet addendum, DocStore number 258132, NXP Semiconductors, Revision 3.2, 26 June 2015.

[14] SmartMX2 P40 family P40C008/012/024/040/072, Wafer Specification, Product data sheet addendum, DocStore number 269832, NXP Semiconductors, Revision 3.2, 30 May 2015.

[15] NXP Secure Smart Card Controller P40C008/012/024/040/072, Information on Guidance and Operation, Guidance and operation manual, DocStore number 269433, NXP Semiconductors, Revision 3.3, 24 February 2017.

8 . 3 Standards

[16] FIPS PUB 46-3: Data Encryption Standard (DES), Federal Information Processing Standards Publication 46-3, US Department of Commerce/National Institute of Standards and Technology, 25 October 1999.

9 Legal information

9.1 Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

9.2 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. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. 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 and its suppliers accept 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.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

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.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications. In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

9.3 Trademarks

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

Tables

Figures

Contents

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

Date of release: 14 December 2020