SN220 Series - Secure Element with Crypto Library

Security Target Lite

Rev. 1.5 — 29 September 2022
CC-21-0258298

Evaluation document
COMPANY PUBLIC

Document information

Information	Content
Keywords	NXP, SN200 Series, *SN220x Single Chip Secure Element and NFC Controller*, Crypto Library, Common Criteria, Security Target Lite
Abstract	This document is the Security Target of the Secure Element of the *SN220x Single Chip Secure Element and NFC Controller* Series with IC Dedicated Software, developed and provided by NXP Semiconductors. The Secure Element complies with Evaluation Assurance Level 6 of the Common Criteria for Information Technology Security Evaluation Version 3.1 with augmentations.

Revision history

Revision number	Date	Description
1.5	29.09.2022	Derived from full Security Target v1.5

1 ST Introduction

1.1 ST Reference

"*SN220 Series - Secure Element with Crypto Library*", Security Target Lite , Revision 1.5, NXP Semiconductors, 29 September 2022.

1.2 TOE Reference

The TOE is named ***"SN220 Series - Secure Element with Crypto Library"***. It consists of

- the Secure Element subsystem of the IC hardware platform SN220x[1],
- IC Dedicated Software (Crypto Library, Services Software and IC Dedicated Support Software), and
- documentation describing the usage of the TOE.

The TOE is available in the following configurations:

- Configuration 1 named ***B0.1 C13***.
- Configuration 2 named ***B0.1 C37***.

In this document the TOE is abbreviated to "*SN220_SE*" [2].

1.3 TOE Overview

1.3.1 Usage and major security functionality

The SN220x Single Chip Secure Element and NFC Controller Series combines on a single die an Embedded Secure Element and a NFC Controller. The two subsystems are called "SN220_SE" and "SN220_NFC". The NFC Controller ist not part of the TOE.

The Embedded Secure Element SN220_SE is based on a Flash-based secure microcontroller platform. A high frequency clocked ARM SC300 core along with state of the art cryptographic hardware coprocessors brings secured applications to a new level in performances and security (see Section 1.3.1.1). The TOE includes Security Software, composed of Services Software and a Crypto Library, that can be used by the Security IC Embedded Software (see Section 1.3.1.2).

The TOE is integral part of the SN220x IC. Note that SN220x without any Security IC Embedded Software for the TOE is available for NXP internal use only.

1.3.1.1 IC Hardware

The hardware part of the SN220_SE incorporates an high frequency clocked ARM SC300 processor, a Public-Key Cryptography (PKC) coprocessor and a Direct Memory Access (DMA) controller, which are all connected over a Memory Management Unit (MMU) to a bus system. This bus system gives access to memories, hardware peripherals and communication interfaces.

[1] The "x" in SN220x indicates the type of the SN220 series (representing e.g. the NFC Controller configuration)

[2] Both notations *SN220_SE* and *SN220x_SE* are used throughout documentation. Both terms shall be considered as synonym.

The ARM SC300 processor is a security enhanced variant of the ARM Cortex M3. It includes the SC300 core and the Nested Vector Interrupt Controller (NVIC). The core implements the ARMv7-M architecture, which supports a subset of the Thumb instruction set. The PKC coprocessor provides large integer arithmetic operations, which can be used by Security IC Embedded Software for asymmetric-key cryptography. Hardware peripherals include coprocessors for symmetric-key cryptography and for calculation of error-detecting codes, and also a random number generator. The DMA controller manages data transfers over communication interfaces like ISO/IEC 7816 compliant interface, Serial Peripheral Interface (SPI), I2C interface and the Secure Mailbox Interface. On-chip memories are Flash memory, ROM and RAMs. The Flash memory can be used to store data and code of Security IC Embedded Software. It is designed for reliable non-volatile storage.

SN220_SE is offered with the NXP Trust Provisioning Service, which involves secure reception, generation, treatment and insertion of customer data and code at NXP.

The documentation of SN220_SE includes a product data sheet, several product data sheet addenda, a user guidance and operation manual, and service documentation. This documentation describes secure configuration and secure use of SN220_SE as well as the services provided with it.

The security functionality of SN220_SE is designed to act as an integral part of a security system composed of SN220_SE and Security IC Embedded Software to strengthen it as a whole. Several security mechanisms of SN220_SE are completely implemented in and controlled by SN220_SE. Other security mechanisms must be treated by Security IC Embedded Software. All security functionality is targeted for use in a potential insecure environment, in which SN220_SE maintains

- correct operation of the security functionality,
- integrity and confidentiality of data and code stored to its memories and processed in the device,
- controlled access to memories and hardware components supporting separation of different applications.

This is ensured by the construction of SN220_SE and its security functionality.

SN220_SE basically provides

- hardware to perform computations on multiprecision integers, which are suitable for public-key cryptography,
- hardware to calculate the Data Encryption Standard with up to three keys,
- hardware to calculate the Advanced Encryption Standard (AES) with different key lengths,
- hardware to support Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) and Counter (CTR) modes of operation for symmetric-key cryptographic block ciphers,
- hardware to support Galois/Counter Mode (GCM) of operation and Galois Message Authentication Code (GMAC) for symmetric-key cryptographic block ciphers,
- hardware to calculate Cyclic Redundancy Checks (CRC),
- hardware to serve with True Random Numbers,
- hardware and service software to control access to memories and hardware components.

In addition, SN220_SE embeds sensors, which ensure proper operating conditions of the device. Integrity protection of data and code involves error correction and error detection codes, light sensing and other security functionality. Encryption and masking mechanisms are implemented to preserve confidentiality of data and code. The IC hardware is shielded against physical attacks.

Also the IC Dedicated Support Software is considered part of the IC Hardware, as it is stored to the ROM of the TOE. It consists of the Factory OS, the Boot OS and the Flash Driver Software. The IC Dedicated Support Software is described in Section 1.4.3.2.

1.3.1.2 Security Software

The IC Dedicated Software provides Security Software that can be used by the Security IC Embedded Software. The Security Software is composed of Services Software and Crypto Library.

The **Services Software** consists of Flash Services Software, Services Framework Software and the part of the Services HAL (Hardware Abstraction Layer). The Flash Services Software manages technical demands of the Flash memory and serves the Security IC Embedded Software with an interface for Flash erase and/or programming. The Services Framework Software represents a collection of different abstractions and utility functions that provide a runtime environment to the individual Services. The Services HAL provides an interface for the Services Software to the hardware that controls the Flash memory.

The Services Software is considered part of the Service Code and is stored in the ROM memory of the TOE with the exception of a small amount of code kept in Flash for backward compatibility purpose.

The **Crypto Library** consists of several binary packages that are pre-loaded to the ROM memory of the TOE with the exception of micro-code for public key cryptography co-processor for usage by the Security IC Embedded Software. The Crypto Library provides

- AES
- Triple-DES (3DES)
- Multi-precision arithmetic operations including exact division, secure modular addition, secure modular subtraction, secure modular multiplication, secure modular inversion, secure arithmetic comparison and secure exact addition.
- RSA
- RSA key generation
- RSA public key computation
- ECDSA (ECC over GF(p)) signature generation and verification
- ECC over GF(p) key generation
- ECDH (ECC Diffie-Hellmann) key exchange
- MontDH (Diffie Hellman key exchange on Montgomery Curves over GF(p)) key generation
- MontDH (Diffie Hellman key exchange on Montgomery Curves over GF(p)) key exchange
- EdDSA (Edwards-curve Digital Signature Algorithm) signature generation and verification
- EdDSA (Edwards-curve Digital Signature Algorithm) key generation
- ECDAA related functions
- Full point addition (ECC over GF(p))
- Standard security level SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE128/256 algorithms

- High security level SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE128/256 algorithms
- HMAC algorithms
- eUICC authentication functions (MILENAGE, TUAK and CAVE)
- Hash-based key derivation function according to ANSI X9.63

In addition, the Crypto Library implements a software (pseudo) random number generator which is initialized (seeded) by the hardware random number generator of the TOE. The Crypto Library also provides a secure copy routine, a secure memory compare routine, cyclic redundancy check (CRC) routines, and includes internal security measures for residual information protection.

Note that the Crypto Library also implements

- KoreanSeed
- OSCCA SM2, OSCCA SM3 and OSCCA SM4
- Felica

However these library elements are not in the scope of evaluation.

The Crypto Library is considered part of the Shared Library functions and is stored in the ROM memory of the TOE with the exception of the PKC coprocessor microcode being stored in FLASH.

1.3.2 TOE Type

The TOE is a Security Integrated Circuit Platform for various operating systems and applications with high security requirements.

1.3.3 Security During Development and Production

The Security IC product life cycle is scheduled in phases, which are defined in the Protection Profile [5].

Phase 2 *IC Development*, phase 3 *IC Manufacturing* as well as phase 4 *IC Packaging* of this life cycle are part of this Security Target. The TOE Delivery is at the end of phase 4.

The development environment of SN220_SE always ranges from phase 2 *IC Development* to TOE Delivery. All other phases are part of the operational environment. This addresses Application Note 1 in in the Protection Profile [5].

In phase 2 *IC Development* of SN220_SE access to sensitive design data of SN220_SE is restricted to people, who are involved in the development of the product.

In phase 3 *IC Manufacturing* the TOE as integral part of SN220x IC are produced and tested on wafers. In this phase NXP also serves as Composite Product Manufacturer by optionally storing Security IC Embedded Software to the Flash of SN220_SE. The NXP Trust Provisioning Service ensures confidentiality and integrity of any customer data in this phase. This incudes secure treatment and insertion of data and code received from the customer as well as random or derived data, which are generated by NXP.

In phase 4 *IC Packaging* SN220x ICs including the TOE are embedded into packages.

The delivery processes between all involved sites provide accountability and traceability of the dies. Authentic delivery of the TOE is supported by its NXP Trust Provisioning Service as described in [45].

1.3.4 Required non-TOE Hardware/Software/Firmware

Besides the SN220_SE the SN220x Single Chip Secure Element and NFC Controller comprises a NFC controller (SN220_NFC) and a shared Power Management Unit (SN220_PMU).

For operation the SN220_SE requires full function of the SN220_PMU subsystem, that is controlled by software of the SN220_NFC subsystem (see Figure 1).

The TOE does not include communication drivers in the IC Dedicated Support Software. Those need to be part of the Security IC Embedded Software.

1.4 TOE Description

1.4.1 Physical Scope of TOE

The SN220x IC is build upon two subsystems: "SN220_SE" and "SN220_NFC". Both subsystem use a shared Power Management Unit ("SN220_PMU"). The toplevel block diagram of SN220x is depicted in Figure 1.

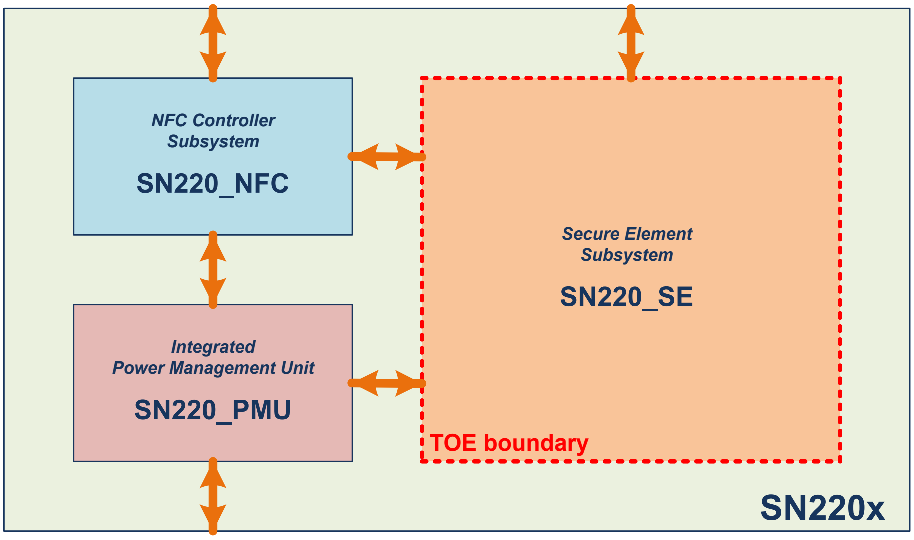


Figure 1. Toplevel block diagram of SN220x

The SN220_SE subsystem is built of IC hardware and IC Dedicated Software, and includes documentation. A block diagram of the TOE and its interfaces is depicted in Figure 2.

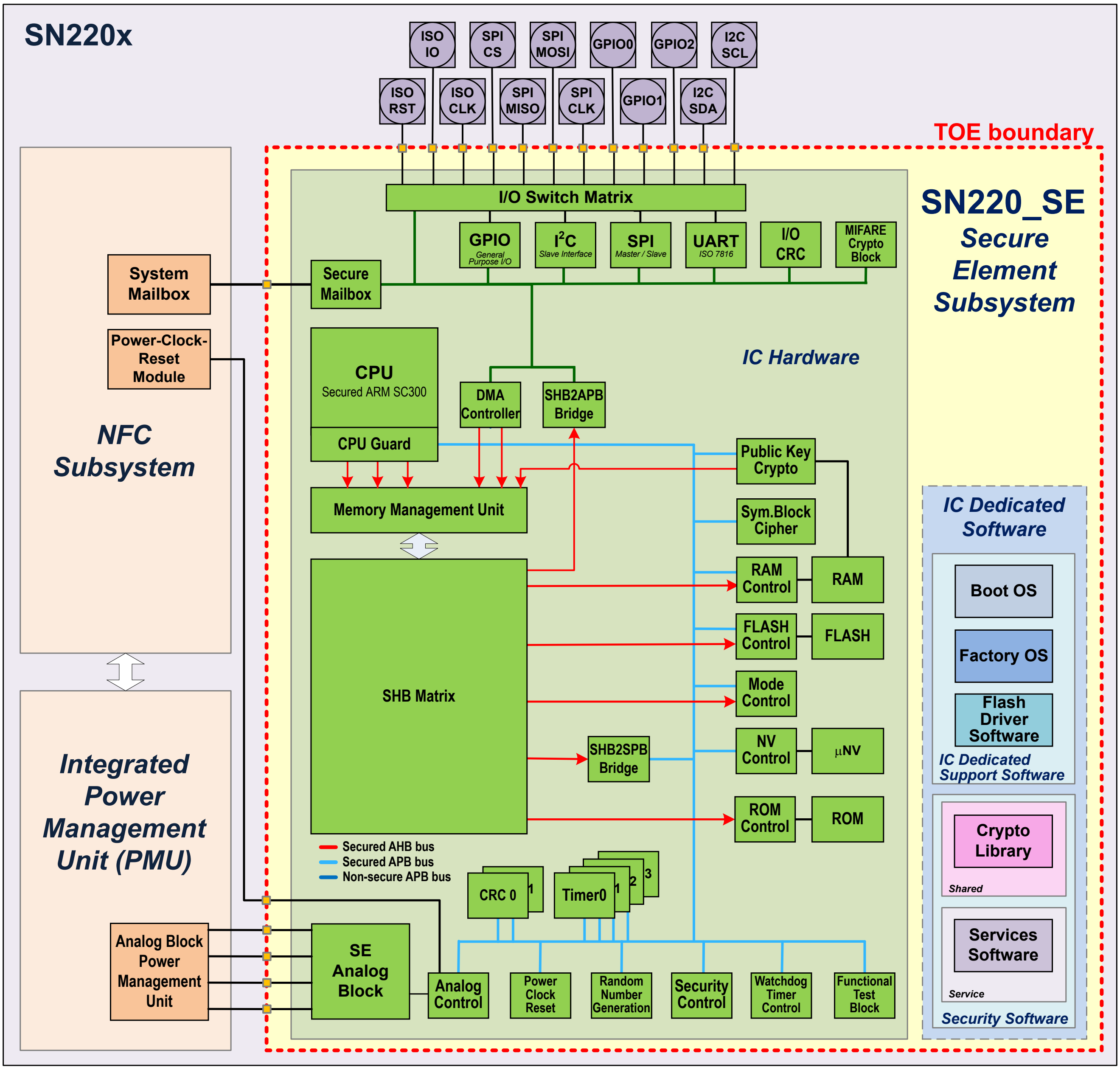


Figure 2. Block diagram of SN220_SE with Interfaces

The IC Dedicated Software of SN220_SE comprises

- IC Dedicated Support Software, composed of
  - Test software named *Factory OS*
  - Boot software named *Boot OS*
  - Memory Driver software named *Flash Driver Software*
- Security Software, composed
  - Services Software named *Services Software*
  - Library Software named *Crypto Library*

All other software is called Security IC Embedded Software and is not part of the TOE.

1.4.2 Evaluated Configurations

Each configuration of the TOE consists of a physical configuration (i.e. hardware component incl. ROM code and related documentation) and a logical configuration (i.e. Software components and configuration data stored to Flash memory).

The definition of the configuration identifiers of SN220_SE is detailed in Table 1.

Table 1. Configuration identifiers of the TOE

Name	Symbol	Description
Series	*srs*	Series identifier in NXP product family
IC version	*xy.z*	*x*: base layer identifier of the development type *y*: fixed metal masks identifier of the development type *z*: customizable metal masks identifier of the development type, includes the IC Dedicated Software stored to ROM
NXP software	*wn*	*w*: NXP software combination identifier of the development type, identifies the IC Dedicated Software stored to Flash *n*: version identifier of the NXP software combination, identifies software version data stored to Flash
NXP hardware configuration	*v*	Version identifier of the NXP hardware configuration, identifies the version of configuration data stored to Flash

The symbols in the second column in Table 1 build the product name of a physical configuration according to the following rule:

srs xy.z wnv

Evaluated **physical** configuration of the TOE is

- *SN220_SE **B0.1***

All components of SN220_SE B0.1 that are common for any logical configuration are listed in Table 2 with their respective version numbers.

Evaluated **logical** configurations of the TOE stored to flash memory are

- *SN220_SE **B0.1 C13***
- *SN220_SE **B0.1 C37***

All components that are specific for SN220_SE B0.1 C13 are listed in Table 3. All components that are specific for SN220_SE B0.1 C37 are listed in Table 4.

Table 2. Components of SN220_SE B0.1 common for any logical configuration

Category	Component	Identification	Delivery form
IC Hardware	base layer and fixed metal masks	B0.1	Package
Documentation, Product Data Sheet	SN220x_SE High-performance secure element subsystem, Product data sheet	[17]	Electronic Document (PDF via NXP Docstore)
	SN220x_SE - SFR Tables for Coburg core	[18]	Electronic Document (PDF via NXP Docstore)
Documentation, Product Data Sheet Addendum	SN220x Wafer and Delivery Specification, Product data sheet addendum	[19]	Electronic Document (PDF via NXP Docstore)
	P73 family SC300 User Manual, Product Data sheet addendum	[20]	Electronic Document (PDF via NXP Docstore)
	P73 family DMA Controller PL080 User manual, Product data sheet addendum	[22]	Electronic Document (PDF via NXP Docstore)

Table 2. Components of SN220_SE B0.1 common for any logical configuration *...continued*

Category	Component	Identification	Delivery form
	P73 Family Chip Health Mode, Application note	[46]	Electronic Document (PDF via NXP Docstore)
	P73 Family Code Signature Watchdog, Application note	[23]	Electronic Document (PDF via NXP Docstore)
	ARM®v7-M Architecture Reference Manual	[21]	Electronic Document (www.arm.com)

Table 3. Components of SN220_SE B0.1 specific for C13

Category	Component	Identification	Delivery form
IC Dedicated Support Software	Factory OS	9.0.4	On-chip software. Stored to the ROM of the TOE
	Boot OS (ROM)	9.0.3	On-chip software. Stored to the ROM of the TOE
	Flash Driver Software	9.0.2	On-chip software. Stored to the ROM of the TOE
Configuration Data	Factory Page	21043	On-chip configuration page. Stored to the ROM area of the TOE
	System Page Common	21031	On-chip configuration page. Stored to the ROM area of the TOE
	BootOS Patch	9.0.3 PL1 v1	On-chip configuration page. Stored to the FLASH area of the TOE
Security Software	Services Software	9.17.4	On-chip software. Stored to the ROM area of the TOE
	Crypto Library	2.2.0	On-chip software. Stored to the ROM and FLASH area of the TOE [1]
Documentation, User Guidance and Operation Manual	SN220_SE Information on Guidance and Operation	[10]	Electronic Document (PDF via NXP Docstore)
	SN220 Services User Manual - API and Operational Guidance	[11]	Electronic Document (PDF via NXP Docstore)
	SN220 Services Addendum - Additional API and Operational Guidance	[13]	Electronic Document (PDF via NXP Docstore)
	SN220x Crypto Library Information on Guidance and Operation	[15]	Electronic Document (PDF via NXP Docstore)
Documentation, User Manuals Crypto Library	User Manual: RNG	[24]	Electronic Document (PDF via NXP Docstore)
	User Manual: Utils	[41]	Electronic Document (PDF via NXP Docstore)
	User Manual: Utils Math	[42]	Electronic Document (PDF via NXP Docstore)
	User Manual: SymCfg	[39]	Electronic Document (PDF via NXP Docstore)
	User Manual: RSA	[32]	Electronic Document (PDF via NXP Docstore)
	User Manual: RSA Key Generation	[34]	Electronic Document (PDF via NXP Docstore)
	User Manual: ECC over GF(p)	[35]	Electronic Document (PDF via NXP Docstore)
	User Manual: ECDAA	[36]	Electronic Document (PDF via NXP Docstore)
	User Manual: SHA	[27]	Electronic Document (PDF via NXP Docstore)
	User Manual: SecSHA	[28]	Electronic Document (PDF via NXP Docstore)
	User Manual: SHA3	[29]	Electronic Document (PDF via NXP Docstore)
	User Manual: SecSHA3	[30]	Electronic Document (PDF via NXP Docstore)
	User Manual: HMAC	[31]	Electronic Document (PDF via NXP Docstore)
	User Manual: HASH	[26]	Electronic Document (PDF via NXP Docstore)

Table 3. Components of SN220_SE B0.1 specific for C13*...continued*

Category	Component	Identification	Delivery form
	User Manual: TwdEdMontGfp	[37]	Electronic Document (PDF via NXP Docstore)
	User Manual: eUICC	[38]	Electronic Document (PDF via NXP Docstore)
	User Manual: Kdf	[43]	Electronic Document (PDF via NXP Docstore)

[1] Header files are provided, as described in [15]

Table 4. Components of SN220_SE B0.1 specific for C37

Category	Component	Identification	Delivery form
IC Dedicated Support Software	Factory OS	10.0.2	On-chip software. Stored to the ROM of the TOE
	Boot OS (ROM)	10.0.2	On-chip software. Stored to the ROM of the TOE
	Flash Driver Software	10.0.0	On-chip software. Stored to the ROM of the TOE
Configuration Data	Factory Page	21043	On-chip configuration page. Stored to the ROM area of the TOE
	System Page Common	21031	On-chip configuration page. Stored to the ROM area of the TOE
	BootOS Patch	10.0.2 PL1 v1	On-chip configuration page. Stored to the FLASH area of the TOE
Security Software	Services Software	10.17.6	On-chip software. Stored to the ROM area of the TOE
	Crypto Library	2.3.1	On-chip software. Stored to the ROM and FLASH area of the TOE [1]
Documentation, User Guidance and Operation Manual	SN220_SE Information on Guidance and Operation	[10]	Electronic Document (PDF via NXP Docstore)
	SN220 Services User Manual - API and Operational Guidance	[12]	Electronic Document (PDF via NXP Docstore)
	SN220 Services Addendum - Additional API and Operational Guidance	[14]	Electronic Document (PDF via NXP Docstore)
	SN220x Crypto Library Information on Guidance and Operation	[16]	Electronic Document (PDF via NXP Docstore)
Documentation, User Manuals Crypto Library	User Manual: RNG	[25]	Electronic Document (PDF via NXP Docstore)
	User Manual: Utils	[41]	Electronic Document (PDF via NXP Docstore)
	User Manual: Utils Math	[42]	Electronic Document (PDF via NXP Docstore)
	User Manual: SymCfg	[40]	Electronic Document (PDF via NXP Docstore)
	User Manual: RSA	[33]	Electronic Document (PDF via NXP Docstore)
	User Manual: RSA Key Generation	[34]	Electronic Document (PDF via NXP Docstore)
	User Manual: ECC over GF(p)	[35]	Electronic Document (PDF via NXP Docstore)
	User Manual: ECDAA	[36]	Electronic Document (PDF via NXP Docstore)
	User Manual: SHA	[27]	Electronic Document (PDF via NXP Docstore)
	User Manual: SecSHA	[28]	Electronic Document (PDF via NXP Docstore)
	User Manual: SHA3	[29]	Electronic Document (PDF via NXP Docstore)
	User Manual: SecSHA3	[30]	Electronic Document (PDF via NXP Docstore)
	User Manual: HMAC	[31]	Electronic Document (PDF via NXP Docstore)
	User Manual: HASH	[26]	Electronic Document (PDF via NXP Docstore)
	User Manual: TwdEdMontGfp	[37]	Electronic Document (PDF via NXP Docstore)

Table 4. Components of SN220_SE B0.1 specific for C37*...continued*

Category	Component	Identification	Delivery form
	User Manual: eUICC	[38]	Electronic Document (PDF via NXP Docstore)
	User Manual: Kdf	[43]	Electronic Document (PDF via NXP Docstore)

[1] Header files are provided, as described in [16]

Logical configuration options are provided for each physical configuration of SN220_SE, which do not modify the physical scope described in Section 1.4.1. Evaluated logical configuration options are all or a subset of the order entry options available in the electronic Order Entry Form [44].

Table 5 identifies these evaluated logical configuration options. These options are detailed in [17].

Table 5. Evaluated logical configuration options

Name of order entry option	Evaluated values
SNSE_HWOPT_ENABLE_ISORESET	YES/NO
SNSE_SWOPT_ENABLE_CHMODE	YES/NO
SNSE_SWOPT_ENABLE_APPDISABLE	YES/NO
SNSE_SWOPT_SELECT_MODE	AAP
SNSE_HWOPT_SELECT_RAM_HS_START	[0..0xFF]
SNSE_HWOPT_SELECT_RAM_HS_END	[0..0xFF]

The logical configuration options given in Table 5 are complemented with additional evaluated logical configuration options. These are not selectable by the customer via electronic Order Entry Form, but are exclusively under control of NXP.

The TOE as integral part of SN220x IC is delivered as a packaged device. The security of the TOE does not rely on the way the pads are connected to the package. Therefore the security functionality of SN220_SE is not affected by the delivered package type.

The only available package type is "Wafer Level Chip Scale Package" (WLCSP). This package is a thin fine-pitch ball grid array package.

The commercial type name of the SN220x IC reflects package type in the name. It is assigned according to the following format:

SN220 b pp(*p*) / *x y zz ff*

The commercial type name of a physical configuration is built by replacing the symbols in the above format with the values identified in Table 6.

Table 6. Values of symbols in commercial type name

Symbol	Value	Description
srs	SN220	Series in NXP product family
b	x	Basic type in the series of NXP product family, defining e.g. the NFC host interface
pp(*p*)	UK	Package type UK = Wafer Level Chip Scale Package (WLCSP)
x	B	Base layer identifier

Table 6. Values of symbols in commercial type name*...continued*

Symbol	Value	Description
y	1	Fixed metal masks identifier
zz	1	ROM Mask reference
ff	Two characters (each either a letter or a number)	FabKey Number (FKN), which identifies the contents in AP-Flash at TOE Delivery, and the selection of logical configuration options, processed by Order Entry Form Tool individually for each OEF

Information on how to order SN220x and how to identify the logical configuration options of the SN220_SE after TOE Delivery is described in [17].

The TOE is integral part of the SN220x IC. Note that SN220x without any Security IC Embedded Software for the TOE is available for NXP internal use only.

The manufacturing process of SN220x allows options that can be selected by NXP in the electronic Order Entry Form [44]. The evaluated options are given in Table 7

Table 7. Evaluated options of manufacturing process (NXP internal only)

Name of order entry option	Evaluated values
Diffusion fab	GF1 / SMIC

The delivery method used for SN220x is described in [19].

1.4.3 Logical Scope of TOE

1.4.3.1 Hardware Description

The hardware of SN220_SE facilitates seven types of software components, which are depicted in Figure 3.

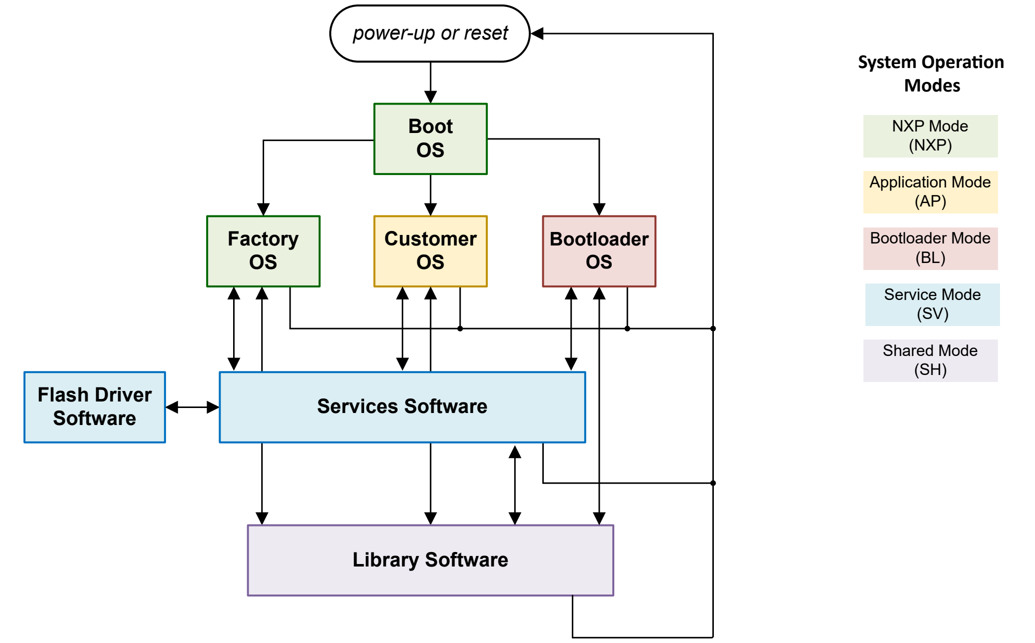


Figure 3. Types of software components and system operation modes facilitated by the hardware

The hardware always starts-up with executing the Boot OS. The Boot OS finally jumps to a start address in either Factory OS, Customer OS or Bootloader OS. The hardware provides no other way to start these operating systems but via power-up or reset of the device. Not more than one operating system out of Factory OS, Customer OS and Bootloader OS can be executed per start-up cycle. Each of the operating systems may interact with and with Library Software according to the programming interface they respectively provide.

The Factory OS implements security functionality against unauthorized access in the field. Startup into Bootloader OS is blocked by the TOE with order entry option SWOPT_SELECT_MODE = AAP (*Always Application*, i.e., mode that shall be entered after Boot Mode completes is Application mode) until Customer OS explicitly unblocks this with next startup by changing the logical configuration to SWOPT_SELECT_MODE = BOR (*BootLoader On Request*). Then Bootloader OS can reactivate this blockage with changing back to SWOPT_SELECT_MODE = AAP. Instead, order entry option SWOPT_SELECT_MODE = BOR causes the TOE to start-up into Bootloader OS when a special sequence is applied to a pad. Please refer to [17] for more information.

Jumps between types of software components imply transformations in system operation modes, which are under control of the hardware. The hardware distinguishes among five such system operation modes. These are named NXP Mode (NXP), Application Mode (AP), Bootloader Mode (BL), Service Mode (SV) and Shared Mode (SH). Figure 3 gives the basic assignment of system operation modes to the seven types of software components.

Transformations among NXP Mode, Bootloader Mode, Application Mode and Service Mode are usually transitions from one to another system operation mode. Exceptions are with logical configurations EN_SV_AP=YES, EN_SV_BL=YES and/or EN_BL_FOR_AP=YES. Logical configuration EN_SV_AP=YES resp. EN_SV_BL=YES enable Bootloader OS to also activate Application Mode resp. Bootloader Mode when it jumps to Services Software. These configurations fit to the needs of update functionality in a Bootloader OS provided by NXP for secure updates of Security IC

Embedded Software. ***Such Bootloader OS itself is not in scope of this TOE.*** In logical configuration EN_BL_FOR_AP=YES the TOE always sets both, Application Mode and Bootloader Mode when jumping to Customer OS. This configuration is appropriate for NXP operating systems with integrated update functionality in the field. Such NXP operating systems themselves are not in scope of this TOE.

Shared Mode is always activated in addition to the system operation mode(s) of the software component type that jumps to Library Software. This allows to share Library Software among different types of software components.

System operation modes are used by the hardware to control access to memories and hardware components. The software component types are stored to different areas in the Flash memory, which are assigned with access rights that fit to their related software component type.

Furthermore, the ARM SC300 processor supports two CPU modes named "thread" and "handler", and also two CPU privilege levels named privileged and unprivileged (of which the latter one is also called "user" by ARM). These choices are combined to three valid CPU operation modes, which are privileged thread, unprivileged thread and privileged handler. The SC300 processor implements these CPU operation modes to control access to some of its configuration registers and instructions. Use of the two modes thread and handler is limited to the SC300 processor whereas the privilege levels are also used in the system to control access to memories and hardware components.

SN220_SE implements 640 Kbytes ROM, 2 Mbytes Flash, 80 Kbytes System RAM, 5 Kbytes PKC RAM and a Buffer RAM for Flash erase/programming and for Flash read caching. All these memories are accessible over the bus system on data/address busses, and the PKC RAM can also be directly accessed by the PKC coprocessor on a separate data/address bus. PKC RAM accesses are arbitrated in the RAM Controller. The hardware controls access to the memories over the bus system. Direct access to the PKC RAM is controlled by way of access control to the hardware component PKC coprocessor. Access to the PKC RAM by the CPU and the PKC coprocessor over the bus system is adjusted accordingly.

The hardware controls write, read and execute access to the memories over the bus system against system operation modes. This is done based on segments in the logical address space. In this context the whole ROM address space is reserved for NXP.

The Flash address space is sectioned into an AP-Flash segment, a BL-Flash segment, an SV-Flash segment, an SH-Flash segment and a CFG-Flash segment. The AP-Flash segment is accessible in Application Mode without restrictions and blocked in Bootloader Mode for read and execute. The BL-Flash segment is accessible in Bootloader Mode without restrictions and completely blocked in Application Mode. Both segments are also blocked in Service Mode. The SV-Flash segment is accessible to Service Mode without restrictions. It is blocked in Application Mode and blocked in Bootloader Mode for read and execute. The SH-Flash segment is accessible for execute in Application Mode, Bootloader Mode and Service Mode, but blocked in all these system operation modes for read and write, except Bootloader Mode, which has write access when Shared Mode isn't also active. Library Software always has the same access rights like the software component from which it is executed and on top of that also has read access to the SH-Flash segment.

The CFG-Flash segment consists of several NXP areas, three System Pages and an area of the Buffer RAM for Flash erase/programming (PBRAM area), which are all under specific access control. The NXP areas are reserved for NXP. The three System Pages are combined of a System Page Application, which is blocked in Bootloader Mode and can be read in Application Mode, a System Page Bootloader, which is blocked in

Application Mode and can be read in Bootloader Mode, and a System Page Common, which can be read in both, Bootloader Mode and Application Mode. All three pages are accessible in read and write in Service Mode so that write access to these in Application Mode and Bootloader Mode can be put under the control of service software. The PBRAM area isn't accessible in Application Mode, Bootloader Mode and Service Mode as long as it is unlocked. In this state, any allowed write access to an address in the Flash address space outside the PBRAM area immediately locks the PBRAM area to the accessing mode. In this context, Application Mode and Bootloader Mode are not distinguished, and they are overruled by Service Mode in case it is active together with one of these. In case the PBRAM area is locked to Application Mode and Bootloader Mode, and Service Mode is active together with one of these, the locking state is updated to Service Mode with any allowed write access to an address in the Flash address area inside or outside the PBRAM area.

The System RAM address space is composed of an AP-RAM segment, an SV-RAM segment and a PUF-RAM segment. The AP-RAM segment is available for use in Application Mode and in Bootloader Mode whereas SV-RAM segment and PUF-RAM segment are reserved for NXP.

The above described restrictions are valid by default for memory access over the bus system by the CPU. Such access by the PKC coprocessor and DMA controller is blocked completely by default, except for PKC coprocessor access to the PUF-RAM segment, which is reserved for NXP, and to the PKC RAM, which is accessible like for the CPU.

The Memory Management Unit can be utilized by software running in privileged level to open access windows over the bus system for PKC coprocessor and DMA controller to areas, which are blocked by default. Such windows for the PKC coprocessor are restricted to AP-Flash segment, BL-Flash segment, SV-Flash segment, SH-Flash segment and AP-RAM segment and for the DMA controller to the AP-RAM segment. The Memory Management Unit can also be utilized by software running in privileged level to open access windows over the bus system for the CPU. Such windows must be inside segments that are accessible to the software which then can block access to the underlying segments and by this restrict access beyond its default. Access rights to all windows can be defined for system operation modes and CPU privilege levels. The Memory Management Unit therewith allows the software to protect its operating system and to implement an access control policy among its different applications.

SN220_SE implements a wide range of hardware components. It embeds the Fast Accelerator for Modular Exponentiation of 3rd Generation (Fame3), which can be utilized by the software to accelerate computations required for public-key cryptography like such related to RSA, Elliptic Curve Cryptography (ECC), and Secure Hash Function (SHA) .

Hardware component Symmetric Block Cipher (SBC) serves the IC Security Embedded Software with interfacing to a DES coprocessor, an AES coprocessor and a GCM coprocessor . The DES coprocessor provides Triple-DES calculation in 2-key or 3-key operation with a length of 56 bits for each key. The AES coprocessor performs AES encryption and decryption calculations with key lengths of 128, 192 or 256 bits. The GCM coprocessor implements a Galois Field Multiplier to support Galois/Counter Mode (GCM) of operation and GMAC performed by the Crypto Library. Besides ECB mode the SBC hardware supports Cipher Block Chaining Mode (CBC), Cipher Feedback Mode, (CFB) Output Feedback Mode (OFB) and Counter Mode (CTR).

Two CRC coprocessors each serve with checksum computation based on CRC generation polynomials CRC-8, CRC-16 and CRC-32. The Random Number Generator generates true random numbers, which are compliant to AIS31 and FIPS 140-2.

SN220_SE also implements a watchdog counter with time-out mechanism that can be utilized by the software to abort irregular program executions, and provides a CPU Guard with several security functionality, which can be utilized by the software to secure its execution.

Hardware components of SN220_SE can be controlled by the IC Security Embedded Software via Special Function Registers, which are accessible over the bus system on two separate busses. The peripheral control bus is provided for communication and thus gives access to the Special Function Registers of the DMA controller, the communication interfaces, the I/O switch matrix and a component for checksum computations over data streams of the communication interfaces. The Special Function Registers of all other hardware components are accessed on the control bus.

The hardware controls write and read access to its Special Function Registers to the point of single bits and this against Application Mode/Bootloader Mode, Service Mode, NXP Mode and against both privilege levels. This control does not distinguish between Bootloader Mode and Application Mode since these are separated via different start-up cycles in which Special Function Registers are reset to their default values. Also, this control does not consider Shared Mode since it is never stand-alone active. This is valid for accesses from the SC300 processor, whereas accesses from the PKC coprocessor are completely blocked for both busses and accesses from the DMA controller are completely blocked for the control bus.

Based on the above conditions the bus system can be utilized by software running in privileged level to further manage access to hardware components among Application Mode/Bootloader Mode and Service Mode as well as among the CPU privilege levels, which is then enforced by the hardware.

SN220_SE implements complex security functionality to protect code and data during processing and while stored to the device. This includes appropriate memory encryptions and masking schemes to preserve confidentiality. This also includes error detection codes, the Flash Secure Fetch Plus, ROM Secure Fetch and manifold light sensing to protect integrity. Active and passive shielding is present and operating conditions are monitored by sensors on temperature, power supplies and frequencies.

The TOE hardware operates with an power supply provided by the shared Power Management Unit ("SN220_PMU"). Normal operation is done in power mode ACTIVE, in which all hardware components are in operative condition. The device can be set into power modes SLEEP, DEEP SLEEP and DEEP POWER DOWN, which have different levels of reduced availability of hardware components with appropriately reduced power consumption.

1.4.3.2 IC Dedicated Support Software Description

The IC Dedicated Support Software of SN220_SE consists of the Factory OS, the Boot OS and the Flash Driver Software.

Boot OS, Factory OS and Flash Driver Software are stored to ROM. Patches to the Boot OS are stored to Flash.

The Factory OS provides controlled access to different levels of testing capabilities of SN220_SE. Full testing capabilities are under restricted access to NXP for production testing of SN220_SE and also for in-depth analysis of field returns from particular utilizations of SN220_SE with Customer OS. In addition, limited testing capabilities are accessible to NXP for basic analysis of field returns, which target to preserve the composite product in its original condition. Beyond that, the Factory OS provides the Composite Product Manufacturer with some basic functional testing of SN220_SE and also with a readout of the identification flags of SN220_SE from System Page Common. The Factory OS implements security functionality to protect from unauthorized access and measures that also authorized access cannot compromise confidentiality of content stored to AP-Flash, BL-Flash, SH-Flash and SV-Flash windows as well as System Page Application, System Page Bootloader and System Page Common.

The Boot OS is executed during start-up after power-on or reset of SN220_SE. It sets up the device and its configuration, and finally jumps to Customer OS, Bootloader OS or Factory OS.

The Flash Driver Software consists of the part of the Services HAL (Hardware Abstraction Layer) that is stored to ROM. The Services HAL provides an interface for the Services Software to the hardware that controls the Flash memory.

1.4.3.3 Services Software Description

The Services Software comprises the Flash Services Software, the Services Framework Software and the part of the Services HAL (Hardware Abstraction Layer) that is also stored to ROM with the exception of a small amount of code kept in Flash for backward compatibility purpose..

Flash Services Software

- The Flash Services Software manages technical demands of the Flash memory and serves the Security IC Embedded Software with an interface for Flash erase and/or programming.
- The Flash Services Software maintains the Flash with re-freshing, tearing-safe updates of Flash contents and wear leveling techniques to ensure integrity and consistency of its content and optimize its endurance.
- For more details, see [11].

Services Framework Software

- The Services Framework Software provides the utility functionality and interface for actual services. This comprises the control of services related functionality such as the resource management, patch handling, service and system configurations functionality.
- For more details, see [11].

1.4.3.4 Crypto Library Description

The Crypto Library (or parts thereof[3]) comprises a set of cryptographic functions.

AES

- The AES algorithm is intended to provide encryption and decryption functionality.
- The Crypto Library implements AES algorithm with different security configurations. For more details on those different configurations please refer the user guidance documentation of the Crypto Library [15].
- The following modes of operation are supported for AES: ECB, CBC, CFB, CTR, GCM, CBC-MAC, CCM, OFB and CMAC.

TDES

[3] Crypto functions are supplied as a library rather than as a monolithic program, and hence a user of the library may include only those functions that are actually required – it is not necessary to include all cryptographic functions of the library in every Security IC Embedded Software. For example, it is possible to omit the RSA or the SHA-1 components. However, some dependencies exist; details are described in the User Manual [10].

- The Triple-DES (TDES) algorithm is intended to provide encryption and decryption functionality.
- The Crypto Library implements Triple-DES algorithm with different security configurations. For more details on those different configurations please refer the user guidance documentation of the Crypto Library [15].
- The following modes of operation are supported for Triple-DES: ECB, CBC, CFB, CTR, CBC-MAC, RetailMAC, OFB and CMAC.
- To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards).

RSA

- The RSA algorithm can be used for encryption and decryption as well as for signature generation, signature verification, message encoding and signature encoding.
- The RSA key generation can be used to generate RSA key pairs.
- The RSA public key generation computation can be used to compute the public key that belongs to a given private CRT key.

The TOE supports various key sizes for RSA from 512 to 4096 bits.

ECC over GF(p)

- The ECDSA (ECC over GF(p)) algorithm can be used for signature generation and signature verification.
- The ECC over GF(p) key generation algorithm can be used to generate key pairs for ECDSA and ECDH.
- The ECDH (ECC Diffie-Hellman) key exchange algorithm can be used to establish cryptographic keys. It can be also used as secure point multiplication.
- Provide secure point addition for Elliptic Curves over GF(p).

The TOE supports various key sizes for ECC over GF(p) from 128 to 640 bits for signature generation, key pair generation and key exchange. For signature verification the TOE supports key sizes up to a limit of 640 bits.

ECDAA

- The ECDAA library component can be used for ECDAA signature generation as specified in the TPM 2.0 [77] specification.

EdDSA & MontDH

- The EdDSA and MontDH over GF(p) library component implements the EdDSA and MontDH over GF(p) related functions:
  - EdDSA key generation, signature generation and signature verification (generalization of Ed25519 and Ed448), support for filling of EdDSA domain parameters
  - MontDH key generation and key exchange for the DH key exchange scheme MontDH (generalization of Curve25519 and Curve448).

eUICC

- The eUICC library component implements the following MILENAGE and TUAK USIM modes: 3G authentication mode, 3G resynchronization mode, Virtual 2G mode, 3G + Kc mode, Anonymity keys for the 3G modes
- Support of the following CAVE operations: SSD generation, Authentication signature generation, CMEA Key and VPM generation

SHA

- The SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE-128 and SHAKE-256 algorithms can be used for different purposes such as computing hash values in the course of digital signature creation or key derivation.
- The Crypto Library implements two versions of each SHA algorithm with different security level: standard and high. The difference between the standard and high security level of the SHA implementations is that the high security level SHA is protected against more side-channel attacks.

To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards). In particular this means that SHA-1 shall not be used.

HMAC

- The HMAC algorithm can be used to calculate Keyed-Hash Authentication code. The TOE supports the calculation of HMAC authentication code with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384 or SHA-3/512 hash algorithms. The HMAC algorithm can use either the high security level or standard security level version of SHA, depending on required security level.

To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards). In particular this means that HMAC with SHA-1 shall not be used.

The TOE supports various key sizes for HMAC. To fend off attackers with high attack potential an adequate key length must be used (references can be found in national and international documents and standards).

Key Derivation Function (KDF)

- The Kdf algorithm can be used to perform hash-based key derivation according to ANSI X9.63 standard [80].

Multi-precision Arithmetic

- The libray provides functions to implemenent various arithmetic operations including exact division, secure modular addition, secure modular subtraction, secure modular multiplication, secure modular inversion, secure arithmetic comparison and secure exact addition.

Resistance of cryptographic algorithms against attacks

The cryptographic algorithms are resistant against attacks as described in JIL [9], which include Side Channel Attacks, Perturbation attacks, Differential Fault Analysis (DFA) and timing attacks, except for standard/high security level SHA and HMAC, which are only resistant against Side Channel Attacks and timing attacks.

More details about conditions and restrictions for resistance against attacks are given in the user documentation of the Crypto Library [15].

Random number generation

- Library component to access random numbers generated by a software (pseudo) random number generator and to perform a test of the hardware (true) random number generator at initialisation.

Further security functionality of the Crypto Library

- Internal security measures for residual information protection
- Secure Memory Copy routine

- Secure Memory Boolean Compare routine
- CRC16 & CRC32 routines for cyclic redundancy check calculation

Note that the TOE does not restrict access to the functions provided by the hardware: these functions are still directly accessible to the Security IC Embedded Software.

1.4.4 Interfaces of the TOE

Electrical interface

The electrical interface of SN220_SE are the 11 lines between the I/O subsystem and the communication pads, that are exclusively used by the SN220_SE subsystem. The interface can be configured to establish communication with the TOE via the following interfaces:

- Serial Peripheral Interface (SPI)
- I^{2}C interface
- ISO/IEC 7816 compliant interface by use of ISO/IEC 7816 UART
- GPIO interface by use of Special Function Registers

The TOE also provides an electrical interface to the SN220_PMU subsystem, which connects power supply voltage input and ground as reference voltage.

Additional dedicated control interface is:

- Control Interface between Analog Control Block and Power-Clock-Reset Module of the SN220_NFC subsystem

Logical interface

Figure 4 illustrates the logical interface to the Security IC Embedded Software (internal interfaces not drawn, refer to Figure 2 instead).

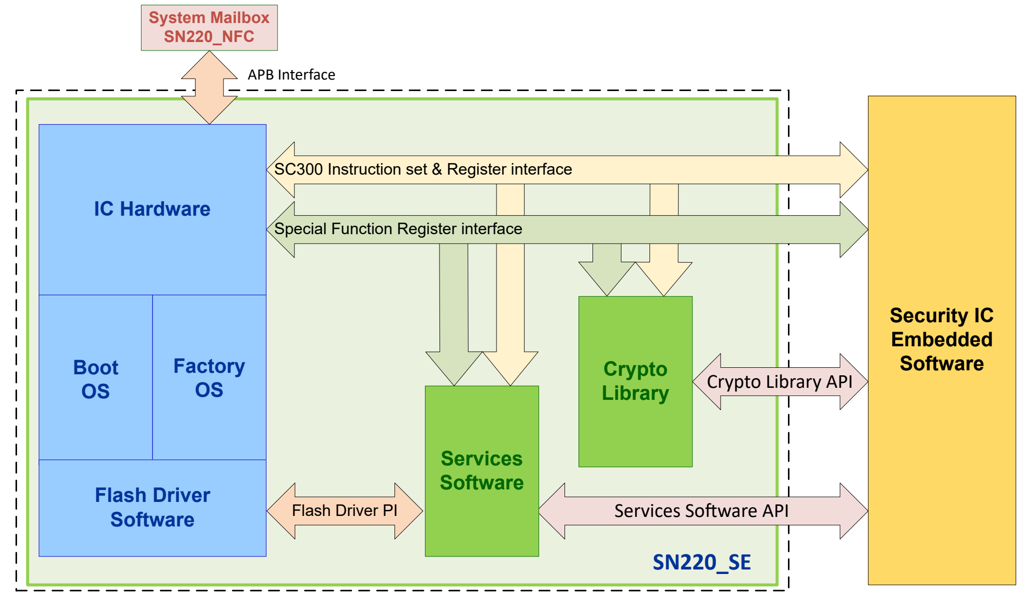


Figure 4. Logical interface of SN220_SE

The logical interface of SN220_SE is composed of the following:

- SC300 Instruction set and Register interface acc. to [21], which is accessible to Security IC Embedded Software as well as Library Software and Services Software running on SN220_SE
- Special Function Registers interface acc. to [17], which is accessible to Security IC Embedded Software as well as Library Software
- Special Function Registers interface acc. to [17], which is accessible to Security IC Embedded Software as well as Services Software
- Crypto Library API, which is accessible to Security IC Embedded Software
- Services Software API, which is accessible to Security IC Embedded Software
- Secure System Mailbox interface for data exhange with SN220_NFC subsystem, which is accessible to Security IC Embedded Software

All logical interfaces other than the Secure System Mailbox interface are accessible via the electrical interfaces SPI, I^2C, UART and GPIO.

Physical interface

The chip surface must be considered as an interface of the TOE as well. This interface could be exposed to environmental stress or physically manipulated by an attacker.

1.4.5 TOE Documentation Overview

- The documentation of the components of the SN220_SE is identified in Table 2 and Table 3 of Section 1.4.2.
- The interfaces of SN220_SE are linked to the documentation in Section 1.4.4.
- Proper use and operation of the hardware is described in [17], with details on some particular hardware components in [20] and [22].
- Particular information on secure use and operation of SN220_SE is provided for the hardware in [10] and for the Services Software in [11].
- The Crypto Library has a separate user guidance documentation [15] and associated user manuals per library component. The user guidance document contains guidelines on the secure usage of the Crypto Library, including the requirements on the environment (the Security IC Embedded Software calling the Crypto Library is considered to be part of the environment). The user manuals contain the specification of the functions provided by the Crypto Library and details of the parameters and options required to call the Crypto Library by the Security IC Embedded Software
- Information on packaging and delivery of the TOE is given in [19].

2 Conformance Claims

2.1 Conformance Claim

This Security Target and SN220_SE claim conformance to version 3.1 of Common Criteria for Information Technology Security Evaluation, which comprises

- "Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017, CCMB-2017-04-001" [1]
- "Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-002" [2]
- "Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-003" [3]

SN220_SE is evaluated against this Security Target in consideration of the methodology in

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017, CCMB-2017-04-004 [4]

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. Section 5 of this Security Target defines the security functional components, which are extended beyond CC Part 2, and also demonstrates that they are consistent with the above conformance claim.

This Security Target also claims strict conformance to Protection Profile

- "Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0084-2014" [5].

This claim includes conformance to packages defined in the Protection Profile [5]:

- Package "TDES" (with augmentations)
- Package "AES" (with augmentations)

The minimum assurance level for the Protection Profile [5] is EAL4 augmented with AVA_VAN.5 and ALC_DVS.2.

This Security Target claims conformance to assurance package **EAL6 augmented** with ALC_FLR.1 and ASE_TSS.2.

This claim includes and exceeds the minimum assurance level for the Protection Profile [5] as demonstrated in Section 6.2 of this Security Target.

2.2 Conformance Claim Rationale

SN220_SE is the type of TOE defined in Section 1.3.2 of this Security Target. Its components are detailed in Section 1.4.1 of this Security Target. These descriptions are consistent with the TOE definition in section 1.2.2 of the Protection Profile [5].

The security problem definition in Section 3 of this Security Target includes all threats, organizational security policies and assumptions, which are identified in the Protection Profile [5], and this without any restrictions or modifications. In addition, this Security Target contains new threats, organizational security policies and assumptions. The new assumptions neither mitigate any threat (or a part of it) nor fulfil any organizational security policy (or part of it). This is demonstrated in Section 3.4 of this Security Target.

This Security Target claim package-augmented conformance to the packages for Cryptographic Services "TDES" and "AES", as the selections in the Security Functional Requirements FCS_COP.1 are augmented by additional list of standards.

3 Security Problem Definition

3.1 Description of Assets

The assets and emanating high-level security concerns in section 3.1 of the Protection Profile [5] entirely apply to this Security Target. In compliance with Application Note 8 in the Protection Profile [5] this Security Target identifies the access restrictions of the TOE to its memories and hardware as a further asset. The high-level security concerns of this Security Target are summarized below.

- SC1 Integrity of user data of the Composite TOE and of Security IC Embedded Software, while being executed/processed and while being stored in the TOE’s protected memories
- SC2 Confidentiality of user data of the Composite TOE and of Security IC Embedded Software, while being executed/processed and while being stored in the TOE’s protected memories
- SC3 Correct operation of the security services provided by the TOE for Security IC Embedded Software
- SC4 Deficiency of Random Numbers
- SC5 Correct operation of access restrictions to memories and hardware as provided by the TOE for Security IC Embedded Software

To be able to protect the assets the TOE shall protect its TOE security functionality. Critical information about the TOE security functionality shall be protected by the development environment and the operational environment. Critical information includes the following.

- Logical design data
- Physical design data
- IC Dedicated Software
- Configuration data
- Initialization data and pre-personalization data
- Specific development aids
- Test and characterization related data
- Material for software development support
- Photomasks

3.2 Threats

The threats defined in section 3.2 of the Protection Profile [5] are listed in Table 8. They entirely apply to this Security Target.

Table 8. Threats defined in the Protection Profile

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

Table 8. Threats defined in the Protection Profile *...continued*

Name	Title
T.RND	Deficiency of Random Numbers

The threat T.RND explicitly includes both deficiencies of hardware (true) random numbers as well as deficiency of software (pseudo) random numbers provided by the Crypto Library.

In compliance with Application Note 4 of the Protection Profile [5] the TOE provides security functionality that protects against the additional threat listed in Table 9.

Table 9. Threats added in this Security Target

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

The threat in Table 9 is defined below.

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

Adverse action: An attacker may try to read, modify or execute code or data stored to restricted memory areas. An attacker may try to access or operate restricted hardware components by executing code that accidentally or deliberately accesses these restricted hardware components.

- Any code executed or data used in a system operation mode, with and without Shared Mode, may accidentally or deliberately access code or data or hardware components restricted to other system operation modes.
- Any code executed or data used in unprivileged level may accidentally or deliberately access code or data or hardware components restricted to privileged level.
- Any code executed or data used in unprivileged level, which is assigned to a certain application, may accidentally or deliberately access code or data or hardware components restricted to unprivileged level of the same system operation mode but assigned to another application.

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

Asset: Code and data belonging to Security IC Embedded Software as well as code and data belonging to IC Dedicated Software.

The TOE provides security functionality for control of access to its memories and hardware components. This control targets to prevent

- Boot OS and Factory OS from being compromised by other software component types,
- Flash Driver Software and Services Software from being compromised by other Security IC Embedded Software - and vice versa,
- Customer OS from being compromised by Bootloader OS - and vice versa,
- Security IC Embedded Software assigned to privileged level from being compromised by Security IC Embedded Software assigned to unprivileged level,

- separate applications of Security IC Embedded Software, which are assigned to unprivileged level of the same system operation mode, from being compromised by each other.

3.3 Organizational Security Policies

The organizational security policies defined in section 3.3 and section 7.4 of the Protection Profile [5] are listed in Table 10. They entirely apply to this Security Target.

Table 10. Organizational security policies defined in the Protection Profile

Name	Title
P.Process-TOE	Identification during TOE Development and Production
P.Crypto-Service	Cryptographic services of the TOE

In compliance with Application Note 5 of the Protection Profile [5] the TOE provides security components and security functionality, which require additional organizational security policies that are listed in Table 11.

Table 11. Organizational security policies added in this Security Target

Name	Title
P.Add-Components	Additional Specific Hardware Security Components
P.Add-Func	Additional Specific Security Functionality of Crypto Library

The organizational security policies in Table 11 are defined as follows.

P.Add-Components

Additional Specific Hardware Security Components

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

- Integrity support of content stored to Flash memory
- Computation of Cyclic Redundancy Checks
- Support for Galois/Counter Mode (GCM) and GMAC

The security policies of the TOE include specific security policies for the Crypto Library. The Crypto Library part of the TOE uses the AES co-processor hardware to provide AES security functionality, and the DES co-processor hardware to provide Triple-DES security functionality. The following security functionality is provided by the Crypto Library for use by the Security IC Embedded Software:

P.Add-Func

Additional Specific Security Functionality of Crypto Library

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

- AES encryption and decryption
- Triple-DES encryption and decryption
- RSA encryption, decryption, signature generation, signature verification, message encoding and signature encoding
- RSA public key computation
- RSA key generation

- ECDSA (ECC over GF(p)) signature generation and verification
- ECC over GF(p) key generation
- ECDH (ECC Diffie-Hellman) key exchange
- ECC over GF(p) point addition
- TPM 2.0 ECDAA (ECC-based Direct Anonymous Attestation) signature generation
- MontDH (Montgomery Curves over GF(p)) key generation
- MontDH (Diffie Hellman on Montgomery Curves) key exchange
- EdDSA (Edwards-curve Digital Signature Algorithm) signature generation and verification
- EdDSA (Edwards-curve Digital Signature Algorithm) key generation
- eUICC authentication functions
- SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE-128 and SHAKE-256 Hash Algorithms
- HMAC algorithm
- Arithmetic operations
- access to the RNG (implementation of a software RNG)
- secure copy routine
- secure compare routine
- Cyclic Redundancy Checks routine

In addition, for this functionality the TOE shall provide protection of residual information, and resistance against attacks as described in Note 4 and in Section 7.2.2.

3.4 Assumptions

The assumptions defined in section 3.4 of the Protection Profile [5] are listed in Table 12. They entirely apply to this Security Target.

Table 12. Assumptions defined in the Protection Profile

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 of the Protection Profile [5] the TOE provides security functionality, which requires an additional assumption that is listed in Table 13.

Table 13. Assumptions added in this Security Target

Name	Title
A.Check-Init	Check of TOE identification data

The assumption in Table 13 is defined below.

A.Check-Init	**Check of TOE identification data** It is assumed that either the Security IC Embedded Software implements a function, which checks the TOE identification data, or the Composite Product Manufacturer uses the command interface to the Factory OS of the TOE to check the TOE identification data. The TOE identification data are part of the initialization data. They are defined with the order entry of the TOE from the Composite Product Manufacturer and are injected by the TOE Manufacturer into the Flash memory of the TOE. TOE identification data can be used to identify and to trace a certain instantiation of the TOE.

4 Security Objectives

4.1 Security Objectives for the TOE

The security objectives for the TOE defined in section 4.1, section 7.2.1 and section 7.4 of the Protection Profile [5] are listed in Table 14. They entirely apply to this Security Target.

Table 14. Security objectives for the TOE defined in the Protection Profile

Name	Title
O.Malfunction	Protection against Malfunctions
O.Abuse-Func	Protection against Abuse of Functionality
O.Phys-Probing	Protection against Physical Probing
O.Phys-Manipulation	Protection against Physical Manipulation
O.Leak-Inherent	Protection against Inherent Information Leakage
O.Leak-Forced	Protection against Forced Information Leakage
O.RND	Random Numbers
O.Identification	TOE Identification
O.TDES	Cryptographic service Triple-DES
O.AES	Cryptographic service AES

In compliance with Application Note 9 of the Protection Profile [5] the TOE provides security functionality that results in the additional security objectives for the TOE listed in Table 15.

Table 15. Security Objectives for the TOE added in this Security Target

Name	Title
O.MEM-ACCESS	Memory Access Control
O.SFR-ACCESS	Special Function Register Access Control
O.FLASH-INTEGRITY	Integrity support of data stored to Flash memory
O.GCM-SUPPORT	Support for NIST Galois/Counter Mode and GMAC
O.CRC	Cyclic Redundancy Checks

In addition, the Crypto Library provides security functionality that results in the additional security objectives for the TOE listed in Table 16.

Table 16. Security Objectives for the TOE related to Crypto Library added in this Security Target

Name	Title
O.SW_AES	Software AES
O.SW_DES	Software DES
O.RSA	RSA
O.RSA_PubExp	RSA public key computation
O.RSA_KeyGen	RSA key pairs generation

Table 16. Security Objectives for the TOE related to Crypto Library added in this Security Target *...continued*

Name	Title
O.ECDSA	ECCSA signature generation & verification
O.ECC_DHKE	ECC Diffie-Hellman key exchange
O.ECC_KeyGen	ECC key pairs generation
O.ECC_Add	ECC point addition
O.ECDAA	TPM 2.0 functions
O.SHA	SHA algorithms
O.HMAC	HMAC algorithm
O.EDDSA	EdDSA signature generation & verification
O.EDDSA_KeyGen	EdDSA key generation
O.MONT_KeyGen	MontDH key generation
O.MONT_DHKE	MontDH Diffie-Hellman key exchange
O.EUICC	eUICC
O.KDF	Hash-based key derivation
O.SW_CRC	Software CRC
O.COPY	Memory copy
O.COMPARE	Memory compare
O.ARITH_OP	Arithmetic operations
O.REUSE	Memory clear for reuse

The security objectives in Table 15 and Table 16 are defined as follows:

O.MEM-ACCESS

Memory Access Control

The TOE controls access of the SC300 processor, the DMA Controller and the PKC coprocessor over the bus system to ROM, Flash address space, System RAM and PKC RAM. The TOE also controls access of the PKC coprocessor over its Direct Memory Access (DMA) channel to PKC RAM. Control of access is enforced on these ports by generic limitations as well as restrictions based on system operation modes and CPU privilege levels.

O.SFR-ACCESS

Special Function Register Access Control

The TOE controls access of the SC300 processor, the DMA Controller and the PKC coprocessor over the bus system to the Special Function Registers of the hardware components. Control of access is enforced on these ports by generic limitations as well as restrictions based on system operation modes and CPU privilege levels.

O.FLASH-INTEGRITY

Integrity support of data stored to Flash memory

The TOE preserves integrity of content stored to its Flash memory with wearout detection capabilities.

O.GCM-SUPPORT

Support for Galois/Counter Mode and GMAC

The TOE provides secure hardware based multiplication operation on blocks and incrementing function for the Galois/Counter Mode (GCM) and GMAC.

O.CRC

Cyclic Redundancy Checks

The TOE provides secure hardware based computation of Cyclic Redundancy Checks (CRC).

O.SW_AES

Software AES

The TOE includes functionality to provide encryption and decryption facilities of the AES algorithm, see Note 4

O.SW_DES

Software DES

The TOE includes functionality to provide encryption and decryption facilities of the Triple-DES algorithm, see Note 4

O.RSA

RSA

The TOE includes functionality to provide encryption, decryption, signature creation, signature verification, message encoding and signature encoding using the RSA algorithm, see Note 4.

O.RSA_PubExp

RSA public key computation

The TOE includes functionality to compute an RSA public key from an RSA private key, see Note 4.

O.RSA_KeyGen

RSA key pairs generation

The TOE includes functionality to generate RSA key pairs, see Note 4.

O.ECDSA

ECCSA signature generation & verification

The TOE includes functionality to provide signature generation and signature verification using the ECC over GF(p) algorithm, see Note 4.

O.ECC_DHKE

ECC Diffie-Hellman key exchange

The TOE includes functionality to provide Diffie-Hellman key exchange based on ECC over GF(p), see Note 4.

O.ECC_KeyGen

ECC key pairs generation

The TOE includes functionality to generate ECC over GF(p) key pairs, see Note 4.

O.ECC_Add

ECC point addition

The TOE includes functionality to provide a point addition based on ECC over GF(p), see Note 4.

O.ECDAA

ECDAA TPM 2.0 functions

The TOE includes functionality to support the TPM 2.0 ECDAA signature generation, see Note 4.

O.SHA

SHA algorithms

The TOE includes functionality to provide electronic hashing facilities using the SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE-128 and SHAKE-256 algorithms.

O.HMAC

HMAC algorithm

The TOE includes the functionality to provide keyed-hash message authentication facilities using the HMAC algorithm.

O.KDF

Hash-based key derivation

The TOE includes the functionality to provide hash-based key derivation algorithm.

O.SW_CRC

Software CRC

The TOE includes functionality to provide Cyclic Redundancy Checks.

O.EDDSA

EdDSA signature generation & verification

The TOE includes functionality to provide signature generation and signature verification using the EdDSA algorithm, see Note 4.

O.EDDSA_KeyGen

EdDSA key generation

The TOE includes functionality to generate EdDSA key pairs, see Note 4.

O.MONT_KeyGen

MontDH key generation

The TOE includes functionality to generate key pairs for the DH key exchange scheme MontDH which generalizes Curve25519 to a wider class of Montgomery curves over GF(p), see Note 4.

O.MONT_DHKE

MontDH Diffie-Hellman key exchange

The TOE includes functionality to provide Diffie-Hellman key exchange for the DH key exchange scheme MontDH which generalizes Curve25519 to a wider class of Montgomery curves over GF(p), see Note 4.

O.EUICC

eUICC

The TOE includes functionality to perform eUICC authentication functions using the cryptographic algorithms MILENAGE, CAVE and TUAK, see Note 4.

O.COPY

Memory copy

The TOE includes functionality to copy memory content, see Note 4.

O.COMPARE

Memory compare

The TOE includes functionality to compare memory content, see Note 4.

O.ARITH_OP

Arithmetic Operations

The TOE includes functionality for secure modular addition, modular subtraction, modular multiplication, modular inversion, arithmetic comparison and exact addition, see Note 4.

O.REUSE

Memory flush for reuse

The TOE includes measures to ensure that the memory resources being used by the TOE cannot be disclosed to subsequent users of the same memory resource.

Note 4. All introduced security objectives claiming cryptographic functionality and the security objectives for copy and compare are protected against attacks as described in the JIL, Attack Methods for smartcards and Similar Devices [9], which include Side Channel Attacks, Perturbation attacks, Differential Fault Analysis (DFA) and timing attack. The following exceptions apply:

1. SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE-128 and SHAKE-256 are provided by the TOE with two implementations with different level of security:
   - One implementation that protects against non-differential side channel attacks (e.g. non-differential template attacks)
   - The second implementation protects against differential and non-differential side channel attacks
2. HMAC implementation do not contain protective measures against DFA.
3. The security objectives for copy, compare and arithmetic operations are secured against fault attacks and non-differential side channel attacks. Please note that non-differential side channel attacks also include profiled non-differential attacks like standard template attacks.

More details about conditions and restrictions for resistance against attacks are given in the user documentation of the Crypto Library.

To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards).

4.2 Security Objectives for the Security IC Embedded Software

The security objectives for the Security IC Embedded Software defined in section 4.2 of the Protection Profile [5] are listed in Table 17. They entirely apply to this Security Target.

Table 17. Security objectives for the Security IC Embedded Software defined in the Protection Profile

Name	Title
OE.Resp-Appl	Treatment of user data of the Composite TOE

This Security Target does not add security objectives for the Security IC Embedded Software.

4.3 Security Objectives for the Operational Environment

The security objectives for the operational environment in section 4.3 of the Protection Profile [5] are listed in Table 18. They entirely apply to this Security Target.

Table 18. Security objectives for the operational environment defined in the Protection Profile

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

This Security Target adds the security objectives for the operational environment listed in Table 19.

Table 19. Security Objectives for the operational environment added in this Security Target

Name	Title
OE.Check-Init	Check of TOE identification data

The security objectives in Table 19 are defined below.

OE.Check-Init

Check of TOE identification data

To ensure the receipt of the correct TOE, the Security IC Embedded Software or the Composite Product Manufacturer shall check the TOE identification data. The TOE identification data are stored to System Page Common of the Flash. They can be used to identify and to trace a certain instantiation of the TOE.

4.4 Security Objectives Rationale

Table 20 traces the security objectives for the TOE in Section 4.1 back to the threats countered by them and the organisational security policies enforced by them. The table also traces the security objectives for the Security IC Embedded Software and for the operational environment in Section 4.2 and Section 4.3 back to the assumptions they uphold.

Table 20. Tracing of security objectives

Name of threat, org. security policy or assumption	Name of security objective	Applied to life cycle phases
T.Malfunction	O.Malfunction	
T.Abuse-Func	O.Abuse-Func	
T.Phys-Probing	O.Phys-Probing	
T.Phys-Manipulation	O.Phys-Manipulation	
T.Leak-Inherent	O.Leak-Inherent	
T.Leak-Forced	O.Leak-Forced	
T.RND	O.RND	
T.Unauthorized-Access	O.MEM-ACCESS	
	O.SFR-ACCESS	
P.Process-TOE	O.Identification	phases 2 to phase 4
P.Crypto-Service	O.TDES	
	O.AES	
P.Add-Components	O.FLASH-INTEGRITY	

Table 20. Tracing of security objectives*...continued*

Name of threat, org. security policy or assumption	Name of security objective	Applied to life cycle phases
	O.GCM-SUPPORT	
	O.CRC	
P.Add-Func	O.SW_AES	
	O.SW_DES	
	O.RSA	
	O.RSA_PubExp	
	O.RSA_KeyGen	
	O.ECDSA	
	O.ECC_DHKE	
	O.ECC_KeyGen	
	O.ECC_Add	
	O.ECDAA	
	O.SHA	
	O.HMAC	
	O.KDF	
	O.EDDSA	
	O.EDDSA_KeyGen	
	O.MONT_KeyGen	
	O.MONT_DHKE	
	O.EUICC	
	O.COPY	
	O.COMPARE	
	O.ARITH_OP	
	O.REUSE	
	O.SW_CRC	
	O.RND	
A.Process-Sec-IC	OE.Process-Sec-IC	phases 5 to 6
A.Resp-Appl	OE.Resp-Appl	
A.Check-Init	OE.Check-Init	phase 1 and phases 5 to 6

The green and blue colored cells in Table 17 show how the Protection Profile [5] traces its security objectives back to its threats, organizational security policies and assumptions, see section 4.4. Green marks this for the mandatory security requirements of the protection profile, blue marks this for the augmentations including package "TDES" and "AES". Section 4.4 of the Protection Profile [5] also gives the security objective rationale for the tracings colored in green.

Security objectives O.TDES and O.AES together enforce organizational security policy P.Crypto-Service since they target such kind of cryptographic services defined in P.Crypto-Service.

O.MEM-ACCESS and O.SFR-ACCESS counter threat T.Unauthorized-Access for two reasons. First, O.MEM-ACCESS targets to control all access ports available in the TOE to its memories and O.SFR-ACCESS targets to control all access ports available in the TOE to the Special Function Registers of its hardware components. Secondly, both objectives target to control accesses via these ports based on system operation modes, which are used to separate software component types from each other and based on CPU privilege levels, which can be used by a software component type to separate its operating system from the applications it may implement and also to separate its applications from each other.

Security objectives O.FLASH-INTEGRITY, O.GCM-SUPPORT and O.CRC together enforce organizational security policy P.Add-Components since they target at the components defined in P.Add-Components.

The objectives O.SW_AES, O.SW_DES, O.RSA, O.RSA_PubExp, O.RSA_KeyGen, O.ECDSA, O.ECC_DHKE, O.ECC_KeyGen, O.ECC_Add, O.ECDAA, O.SHA, O.HMAC, O.EDDSA, O.EDDSA_KeyGen, O.MONT_KeyGen, O.MONT_DHKE, O.EUICC, O.KDF, O.COPY, O.COMPARE, O.ARITH_OP, O.SW_CRC and O.REUSE require the TOE to implement exactly the same specific security functionality as required by P.Add-Func, therefore the organizational security policy P.Add-Func is covered by the security objectives.

The security objective for the operational environment OE.Check-Init derives from assumption A.Check-Init. It requires the operational environment to implement the measure assumed in assumption A.Check-Init.

5 Extended Components Definition

The extended components defined in chapter 5 of the Protection Profile [5] are listed in Table 21. They entirely apply to this Security Target.

Table 21. Extended components defined in the Protection Profile

Name	Title
FCS_RNG	Generation of random numbers
FMT_LIM	Limited capabilities and availability
FAU_SAS	FAU_SAS Audit data storage
FDP_SDC	Stored data confidentiality
FCS_CKM	Cryptographic key derivation

To define the IT Security Functional Requirements of the TOE an additional family (FDP_SOP) of the Class FDP (user data protection) is defined here. This family describes the functional requirements for basic operations on data in the TOE.

As defined in CC Part 2, FDP class addresses user data protection. Secure basic operations (FDP_SOP) address protection of user data when it is processed by Copy or Compare function, respectively. Therefore, it is judged that FDP class is suitable for FDP_SOP family.

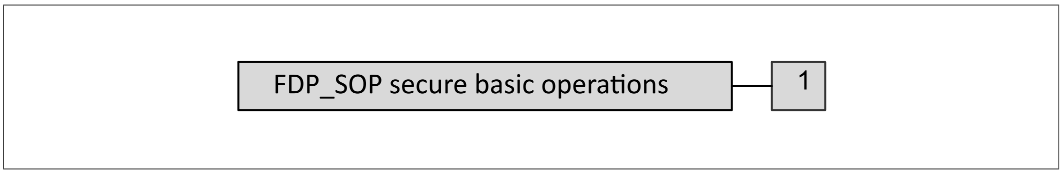
The reason for adding an extra family to FDP class is that existing families do not address protection of user data against all relevant attacks.

5.1 Secure basic operations (FDP_SOP)

Family Behaviour

This family defines requirements addressing the protection of data during security relevant basic operations inside the TSF. The data can comprise user data as well as TSF data. Appropriate separation between user data or TSF data shall be ensured by sequential, atomic processing of either TSF data or user data. The integrity and confidentiality of the data shall be protected during the processing of the basic operation against attacks. Each influence or interaction of the TOE that is not intended and/or specified is considered as attack.

Component levelling



FDP_SOP.1 requires the TOE to provide the possibility to perform basic secure operations on data

Management: FDP_SOP.1

There are no management activities foreseen.

Audit: FDP_SOP.1

There are no actions defined to be auditable.

FDP_SOP.1	**Secure Basic Operations**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SOP.1.1	The TSF shall provide basic operations [selection: *Copy, Move, Compare*] on objects stored in the TOE. The basic operation is applied between objects stored in [assignment: *list of memory locations*] and [assignment: *list of memory locations*].
FDP_SOP.1.2	The TSF shall protect the data against attacks from [selection: *disclosure, modification*] that can be inherently applied during the processing of the basic operations.
Application Notes:	The different memories are seen as possible objects.

The attacks addressed by disclosure and modification comprise side-channel attacks including timing attacks, fault injection attacks including manipulation of the basic operation result and attacks trying to violate the data separation based on the sequential operation.

5.2 Cryptographic Key Derivation (FCS_CKM.5)

FCS_CKM.5 Cryptographic Key Derivation requires the TOE to provide key derivation which can be based on an assigned standard.

There are no actions defined to be auditable.

FCS_CKM.5	**Cryptographic Key Derivation**
Hierarchical to:	No other components
Dependencies:	[FCS_CKM.2 Cryptographic Key Distribution, or FCS_COP.1 Cryptographic Operation] FCS_CKM.4 Cryptographic Key Destruction
FCS_CKM.5.1	The TSF shall derive cryptographic keys [assignment: *key type*] from [assignment: *input parameters*] in accordance with a specified cryptographic key derivation algorithm [assignment: *cryptographic key derivation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].
Application Notes:	None

6 Security Requirements

6.1 Security Functional Requirements for the TOE

6.1.1 General

Security functional requirements from the Protection Profile [5] are applied to this Security Target as described in Section 6.1.2. In compliance with Application Note 12 in the Protection Profile [5] this Security Target adds security functional requirements as detailed in Security Functional Requirements added in this Security Target.

6.1.2 Security Functional Requirements from Protection Profile

Table 22 lists the security functional requirements for the TOE, which are defined in section 6.1 and in sections 7.4.1 and 7.4.2 of the Protection Profile [5]. They entirely apply to this Security Target.

Table 22. Security Functional Requirements from the Protection Profile

Name	Title
FRU_FLT.2	Limited fault tolerance
FPT_FLS.1	Failure with preservation of secure state
FMT_LIM.1	Limited capabilities
FMT_LIM.2	Limited availability
FPT_PHP.3	Resistance to physical attack
FDP_ITT.1	Basic internal transfer protection
FPT_ITT.1	Basic internal TSF data transfer protection
FDP_IFC.1	Subset information flow control

On some further Security Functional Requirements from the Protection Profile [5] operations are made. Table 22 gives an overview on the Security Functional Requirement that were subject to refinement, selection, assignment and/or iteration operations in this Security Target.

Table 23. Security Functional Requirements from the Protection Profile with operations done in this Security Target

Name	Title
FAU_SAS.1	Audit storage
FDP_SDC.1	Stored data confidentiality
FDP_SDI.2: • FDP_SDI.2/AGE • FDP_SDI.2/FLT	Stored data integrity monitoring and action
FCS_RNG.1: • FCS_RNG.1/PTG.2	Random number generation
FCS_COP.1: • FCS_COP.1/TDES • FCS_COP.1/AES	Cryptographic operation

Table 23. Security Functional Requirements from the Protection Profile with operations done in this Security Target *...continued*

Name	Title
FCS_CKM.4: • FCS_CKM.4/TDES • FCS_CKM.4/AES	Cryptographic key destruction

FPT_FLS.1 requests the TSF to preserve a secure state when the TOE is exposed to operating conditions which may not be tolerated according to FRU_FLT.2. The TOE detects such operating conditions and forces itself into a secure state as long as these conditions are valid. This secure state is enforced by security feature SF.OPC as described in Section 7.1.3. This addresses Application Note 14 in the Protection Profile [5].

The TOE does not generate audit data for FRU_FLT.2 and/or FPT_FLS.1. This addresses Application Note 15 in the Protection Profile [5].

FPT_PHP.3 requests the TSF to resist physical manipulation and physical probing by responding automatically such that the security functional requirements are always enforced. The TOE implements two types of such automatic responses. One type of response is permanent and implicitly hampers exploitability or already incidence of physical attacks. The other type of response is conditional upon a failed check and explicitly detects physical attacks. Such type of response stops operation of the TOE or the attacked parts of it. These responses are enforced by security feature SF.PHY as described in Section 7.1.3. This addresses Application Note 19 in the Protection Profile [5].

Refinement, selection, assignment and iteration operations on the security functional requirements in Table 22 are performed in this Security Target as detailed below. Iteration operations are notified by a slash, which is appended to the name of the security functional requirement and followed by an identifier. Selection and assignment operations are denoted in italics. Refinements are denoted just as described in the Protection Profile [5].

This Security Target performs one selection and two assignment operations on FAU_SAS.1 according to Application Note 17 in the Protection Profile [5].

FAU_SAS.1	**Audit storage**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FAU_SAS.1.1	The TSF shall provide the test process before TOE Delivery with the capability to store *the Initialisation Data, Pre-personalisation Data and other user data*[4] in the *Flash memory* [5].

This Security Target performs one assignment operation on FDP_SDC.1 according to Application Note 18 in the Protection Profile [5].

FDP_SDC.1	**Stored data confidentiality**
Hierarchical to:	No other components.
Dependencies:	No dependencies.

[4] [selection: *the Initialisation Data, Pre-personalisation Data, [assignment: other data]*]

[5] [assignment: *type of persistent memory*]

FDP_SDC.1.1	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the *Flash memory, the System RAM, the PKC RAM and the Buffer RAM* [6].

This Security Target performs two iteration operations on FDP_SDI.2, which comply with section 8.1 in CC Part 1 [1], and also performs two assignment operations on each iteration according to Application Note 18 in the Protection Profile [5].

FDP_SDI.2/AGE	**Stored data integrity monitoring and action - Ageing**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1/AGE	The TSF shall monitor user data stored in containers controlled by the TSF for *integrity violations due to ageing* [7] on all objects, based on the following attributes: *ageing check information associated with the data including code stored to the Flash memory* [8].
FDP_SDI.2.2/AGE	Upon detection of a data integrity error, the TSF shall *raise a wearout failure* [9].

FDP_SDI.2/FLT	**Stored data integrity monitoring and action - Faults**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1/FLT	The TSF shall monitor user data stored in containers controlled by the TSF for *modification, deletion, repetition or loss of data* [10] on all objects, based on the following attributes: *integrity check information associated with the data including code stored to the Flash memory, the ROM, the System RAM, the PKC RAM and the Buffer RAM* [11].
FDP_SDI.2.2/FLT	Upon detection of a data integrity error, the TSF shall *correct the error or trigger a security reset or raise a non-maskable interrupt* [12].

This Security Target performs an iteration operation on FCS_RNG.1, which complies with section 8.1 in CC Part 1 [1]. It also performs two assignment operations on each iteration of FCS_RNG.1 according to Application Note 21 in the Protection Profile [5]. The operations follow the example and its Application Note 44 in section 7.5.1 of the Protection Profile [5] in consideration of the updated documents [7] and [6].

FCS_RNG.1/PTG.2	**Random number generation - PTG.2**
Hierarchical to:	No other components.
Dependencies:	No dependencies.

[6] [assignment: *memory area*]
[7] [assignment: *integrity errors*]
[8] [assignment: *memory area*]
[9] [assignment: *action to be taken*]
[10] [assignment: *integrity errors*]
[11] [assignment: *memory area*]
[12] [assignment: *action to be taken*]

Note:	This security functional requirement complies with PTG.2 in [6]
FCS_RNG.1.1/PTG.2	The TSF shall provide a *physical* [13] random number generator that implements: *(PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.* *(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.* *(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.* *(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.* *(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered continuously. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.* [14]
FCS_RNG.1.2/PTG.2	The TSF shall provide *octets of bits or packages of 32 bits*[15] that meet *(PTG.2.6) Test procedure A does not distinguish the internal random numbers from output sequences of an ideal RNG.* *(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.* [16]

Sections 7.4.1 and 7.4.2 of the Protection Profile [5] perform operations on two iterations of each, FCS_COP.1 and FCS_CKM.4 for package "TDES" and for package "AES". This Security Target completes these operations in compliance with section 8.1 in CC Part 1 [1]: On the iterations of FCS_COP.1 selection and refinement operations are performed. On the iterations of FCS_CKM.4 assignments are made.

FCS_COP.1/TDES	**Cryptographic operation - TDES**
Hierarchical to:	No other components.

[13] [selection: *physical, hybrid physical, hybrid deterministic*]

[14] [assignment: *list of security capabilities*]

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

[16] [assignment: *a defined quality metric*]

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

FCS_COP.1.1/TDES The TSF shall perform *encryption and decryption* [17] in accordance with a specified cryptographic algorithm *TDES in ECB mode and with support for CBC mode, CFB mode, OFB mode, CTR mode*[18] and cryptographic key sizes *112 bit, 168 bit* [19] that meet the following: *NIST SP 800-67* [79] , *NIST SP 800-38A* [47] [48] [20].

FCS_CKM.4/TDES **Cryptographic key destruction - TDES**

Hierarchical to: No other components.

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

FCS_CKM.4.1/TDES The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *overwriting the internally stored key* [21] that meets the following: *none* [22] .

FCS_COP.1/AES **Cryptographic operation - AES**

Hierarchical to: No other components.

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

FCS_COP.1.1/AES The TSF shall perform *encryption and decryption* [23] in accordance with a specified cryptographic algorithm *AES in ECB mode and with support for CBC mode, CFB mode, OFB mode, CTR mode*[24] and cryptographic key sizes *128 bit, 192 bit, 256 bit* [25] that meet the following: *FIBS 197* [58] , *NIST SP 800-38A* [47] [48] [26].

FCS_CKM.4/AES **Cryptographic key destruction - AES**

Hierarchical to: No other components.

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

[17] [assignment: *list of cryptographic operations*]

[18] [assignment: *list of cryptographic algorithm*]

[19] [assignment: *cryptographic key sizes*]

[20] [assignment: *list of standards*]

[21] [assignment: *cryptographic key destruction method*]

[22] [assignment: *list of standards*]

[23] [assignment: *list of cryptographic operations*]

[24] [assignment: *list of cryptographic algorithm*]

[25] [assignment: *cryptographic key sizes*]

[26] [assignment: *list of standards*]

security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1/AES The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *overwriting the internally stored key* [27] that meets the following: *none* [28].

6.1.3 Security Functional Requirements added in this Security Target

Table 24 lists the Security Functional Requirements for the TOE, which are added in this Security Target. These Security Functional Requirements are taken from CC Part 2 [2]. They are subject to refinement, selection, assignment and/or iteration operations in this Security Target. The Security Functional Requirement based on the definition in Section 5.1 is listed here as well.

Table 24. Security functional requirements added in this Security Target

Name	Title
FDP_ACC.1: • FDP_ACC.1/MEM • FDP_ACC.1/SFR	Subset access control
FDP_ACF.1: • FDP_ACF.1/MEM • FDP_ACF.1/SFR	Security attribute based access control
FMT_MSA.1: • FMT_MSA.1/MEM • FMT_MSA.1/SFR	Management of security attributes
FMT_MSA.3: • FMT_MSA.3/MEM • FMT_MSA.3/SFR	Static attribute initialisation
FMT_SMF.1	Management of TSF data

[27] [assignment: *list of cryptographic key destruction method*]
[28] [assignment: *list of standards*]

Table 24. Security functional requirements added in this Security Target *...continued*

Name	Title
FCS_COP.1: • FCS_COP.1/GCM • FCS_COP.1/CRC • FCS_COP.1/SW_AES • FCS_COP.1/SW_DES • FCS_COP.1/RSA • FCS_COP.1/RSA_PAD • FCS_COP.1/RSA_PubExp • FCS_COP.1/ECDSA • FCS_COP.1/ECC_DHKE • FCS_COP.1/ECC_Add • FCS_COP.1/ECDAA • FCS_COP.1/SHA • FCS_COP.1/HMAC • FCS_COP.1/EDDSA • FCS_COP.1/MONT_DHKE • FCS_COP.1/EUICC • FCS_COP.1/SW_CRC	Cryptographic operation
FCS_CKM.1: • FCS_CKM.1/RSA • FCS_CKM.1/ECC • FCS_CKM.1/EDDSA • FCS_CKM.1/MONT	Cryptographic key generation
FCS_CKM.4: • FCS_CKM.4/CL	Cryptographic Key Destruction
FCS_CKM.5: • FCS_CKM.5/KDF	Cryptographic Key Derivation
FDP_RIP.1	Subset Residual Information Protection
FCS_RNG.1 • FCS_RNG.1/HYB-DET • FCS_RNG.1/HYB-PHY	Random number generation
FDP_SOP.1 • FDP_SOP.1/Copy • FDP_SOP.1/Compare • FDP_SOP.1/Arith_op	Secure basic operations

The security functional requirements in Table 24 address the Access Control Policy of the TOE. This Access Control Policy is applied to the memories and hardware components. It is enforced on the following access ports, which are:

- *CPU_ovBSY*: CPU access over the bus system
- *DMA_ovBSY*: DMA controller access over the bus system
- *PKC_ovBSY*: PKC coprocessor access over the bus system
- *PKC_ovDMA*: PKC coprocessor access over the DMA channel

by generic limitations as well as restrictions based on the following system operation modes (*SOMs*):

- *AP*: Application Mode
- *BL*: Bootloader Mode
- *SV*: Service Mode
- *SH*: Shared Mode
- *NXP*: NXP Mode

and the following CPU privilege levels:

- *P*: privileged
- *U*: unprivileged

The Access Control Policy controls access to two groups of objects, which are *objects for access control to memories* and *objects for access control to hardware components*. The objects of each group are detailed below.

Objects for access control to memories are as follows.

- *DWINs*: default address windows, which do not overlap in their address ranges with each other. All address ranges are fixed in hardware, except for one window, which can be configured by software as an option.
- *SWWINs*: software-controlled address windows, which must overlap with *DWINs*. They can be configured by software as an option.

Objects for access control to hardware components are as follows.

- *GSFR_ALL*: The Special Function Registers (SFRs) of all hardware components as composed of
  - *GSFR_PCBUS*: All SFRs of the hardware components connected to the peripheral control bus as composed of
    - *GSFR_DMAC*: SFRs of the DMA controller
    - *GSFR_IOSM*: SFRs of the IO Switch Matrix
    - *GSFR_IOCC*: SFRs of the IO-CRC/LRC
    - *GSFR_SMB*: SFRs of the Secure Mailbox communication interface
    - *GSFR_GPIO*: SFRs of the Port IO communication interface
    - *GSFR_UART*: SFRs of the ISO7816 UART communication interface
    - *GSFR_I2C*: SFRs of the I2C communication interface
    - *GSFR_SPI0*: SFRs of the SPI communication interface
  - *GSFR_CBUS*: All SFRs of the hardware components connected to the control bus as composed of
    - *GSFR_GRD*: SFRs of the CPU Guard
    - *GSFR_PKC*: SFRs of the PKC coprocessor
    - *GSFR_SBC*: SFRs of the SBC interface to AES and DES coprocessors
    - *GSFR_PCR*: SFRs of the PCR
    - *GSFR_CRCi*: SFRs of CRC coprocessor i=0,1
    - *GSFR_TMRi*: SFRs of Timer i=0,1,2,3
    - *GSFR_WDG*: SFRs of the Watchdog Timer
    - *GSFR_PUF*: SFRs of the PUF
    - *GSFR_RNG*: SFRs of the Random Number Generator
    - *GSFR_OTHERS*: SFRs of all other hardware components on the control bus

The *objects for access control to memories* are controlled against access rights in read (*r*) and write (*w*) and for *CPU_ovBSY* access also against access rights in execute (*x*). The *objects for access control to hardware components* are controlled against access rights in read (*r*) and write (*w*).

The Access Control Policy is applied to the following *subjects of access control to memories and hardware components*.

Subjects of access control to memories and hardware components are these:

- *CPU_ovBSY*: accesses via 7 types of software component types as follows
  - *BOS_ovBSY*: Boot OS, stored to *DWIN_ROM*, executed in *NXP*
  - *FOS_ovBSY*: Factory OS , stored to *DWIN_ROM*, executed in *NXP*
    COS_ovBSY: Customer OS, stored to *DWIN_AP-FLH*, executed in AP or (AP and BL)
  - *BLOS_ovBSY*: Bootloader OS, stored to *DWIN_BL-FLH*, executed in *BL*
  - *FDSW_ovBSY*: Flash Driver Software, stored to *DWIN_ROM*, executed like *ssw_ovBSYS*
  - *ssw_ovBSY*: services software, stored to *DWIN_SV-FLH*, executed in
    - *SV* when called by software in *AP*, in *BL* or in *AP* + *BL*
    - *SV* + *AP* when called by software in *BL* or in *AP* + *BL*
    - *SV* + *BL* when called by software in *BL* or in *AP* + *BL*
    - *SV* + *AP* + *BL* when called by software in *AP* + *BL*
  - *LSW_ovBSY*: Library Software, stored to *DWIN_SH-FLH*, executed in *SH* and the *SOM(s)* of the software from which it is executed
- *DMA_ovBSY*: accesses in the *SOM(s)* in which the actual *CPU_ovBSY* access runs, but *SV* and *SH* are masked out
- *PKC_ovBSY*: accesses in the *SOM(s)* in which the PKC coprocessor was recently started
- *PKC_ovDMA*: accesses w/o *SOM(s)*

The Access Control Policy of the above subjects to the above objects is defined in the following security functional requirements. The rules there are given for each port accessing in a single *SOM*. In case a port accesses in more than one *SOM* the access rights of this port are the sum of its granted access rights in each single *SOM* except in cases where rules are explicitly stated for combinations of *SOM(s)*. This occurs to some extent for combinations with *SV+AP*, *SV+BL* and *BL+SH*.

This Security Target performs two iteration operations on FDP_ACC.1 and also two assignment operations on each iteration, which comply with section 8.1 of CC Part 1 1.

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 *Access Control Policy* [29] on *all subjects, all objects for access control to memories and all operations on the objects for access control to memories* [30].

FDP_ACC.1/SFR	**Subset access control - Hardware components**
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control

[29] [assignment: *access control SFP*]

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

FDP_ACC.1.1/SFR	The TSF shall enforce the *Access Control Policy* [31] on *all subjects, all objects for access control to Special Function Registers and all operations on the objects for access control to Special Function Registers* [32].

This Security Target performs two iteration operations on FDP_ACF.1 and also five assignment operations on each iteration, which comply with section 8.1 in CC Part 1 1.

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 *Access Control Policy* [33] to objects based on the following: *all subjects and all objects for access control to memories and security attributes for memories.* [34].
Application Note:	List of all subjects and all objects for access control to memories and security attributes for memories is given in the full Security Target.
FDP_ACF1.2/MEM	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed *for CPU_ovBSY access to DWINs, DMA_ovBSY access to DWINs, PKC_ovBSY access to DWINs and PKC_ovDMA access to DWINs* [35].
Application Note:	List of rules to determine if an operation among controlled subjects and controlled objects is allowed for CPU_ovBSY access to DWINs, DMA_ovBSY access to DWINs, PKC_ovBSY access to DWINs and PKC_ovDMA access to DWINs is given in the full Security Target.
FDP_ACF1.3/MEM	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *rules to explicitly authorise DMA_ovBSY access to DMAWIN, PKC_ovBSY access to PKCWIN0 and PKC_ovBSY access to PKCWIN1* [36].
Application Note:	List of rules to explicitly authorise DMA_ovBSY access to DMAWIN, PKC_ovBSY access to PKCWIN0 and PKC_ovBSY access to PKCWIN1 is given in the full Security Target.

[31] [assignment: *access control SFP*]

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

[33] [assignment: *access control SFP*]

[34] [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*]

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

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

FDP_ACF1.4/MEM	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *CPU_ovBSY access to DWINs, CPU_ovBSY access to IDWINn and CPU_ovBSY access to SWINn* [37].
Application Note:	List of rules to explicitly deny CPU_ovBSY access to DWINs, CPU_ovBSY access to IDWINn and CPU_ovBSY access to SWINn is given in the full Security Target.

FDP_ACF.1/SFR	**Security attribute based access control - Hardware components**
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 *Access Control Policy* [38] to objects based on the following: *all subjects and all objects for access control to Special Function Registers and security attributes S2A_APACTRL, S2A_SVACTRL, S2S_APACTRL, S2S_SVACTRL* [39].
FDP_ACF1.2/SFR	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *rules for CPU_ovBSY, DMA_ovBSY, PKC_ovBSY and PKC_ovDMA access to SFR_PCBUS and SFR_CBUS.* [40].
Application Note:	List of rules for CPU_ovBSY, DMA_ovBSY, PKC_ovBSY and PKC_ovDMA access to SFR_PCBUS and SFR_CBUS is given in the full Security Target.
FDP_ACF1.3/SFR	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

- *CPU_ovBSY, DMA_ovBSY access to each group out of GSFR_DMAC, GSFR_IOSM, GSFR_IOCC, GSFR_SMB, GSFR_GPIO, GSFR_UART, GSFR_I2C, GSFR_SPI0 can be:*
  - *in AP for U: allowed in rw acc. to the rules for each bit in GSFR_ALL for a group by setting its corresponding bit in S2A_APUCTRL*
  - *in SV for U: allowed in rw acc. to the rules for each bit in GSFR_ALL for a group by setting its corresponding bit in S2A_SVUCTRL*

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

[38] [assignment: *access control SFP*]

[39] [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant securtiy attributes, or named groups of SFP-relevant security attributes*]

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

- *CPU_ovBSY access to each group out of GSFR_GRD, GSFR_PKC, GSFR_SBC, GSFR_PCR, GSFR_CRCi, GSFR_TMRi, GSFR_WDG, GSFR_PUF, GSFR_RNG can be:*
  - *in AP for U: allowed acc. to the rules for each bit in GSFR_ALL in for a group by setting its corresponding bit in S2S_APUCTRL*
  - *in SV for U: allowed acc. to the rules for each bit in GSFR_ALL in for a group by setting its corresponding bit in S2S_SVUCTRL*[41].

FDP_ACF1.4/SFR The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

- *CPU_ovBSY, DMA_ovBSY access to each group out of GSFR_DMAC, GSFR_IOSM, GSFR_IOCC, GSFR_SMB, GSFR_GPIO, GSFR_UART, GSFR_I2C, GSFR_SPI0 can be:*
  - *in AP for P: denied in rw for a group by clearing its corresponding bit in S2A_APACTRL*
  - *in SV for P: denied in rw for a group by clearing its corresponding bit in S2A_SVACTRL*
- *CPU_ovBSY access to each group out of GSFR_GRD, GSFR_PKC, GSFR_SBC, GSFR_PCR, GSFR_CRCi, GSFR_TMRi, GSFR_WDG, GSFR_PUF, GSFR_RNG can be:*
  - *in AP for P: denied in rw for a group by clearing its corresponding bit in S2S_APACTRL*
  - *in SV for P: denied in rw for a group by clearing its corresponding bit in S2S_SVACTRL* [42].

This Security Target performs two iteration operations on FMT_MSA.1 and also one selection and four assignment operations on each iteration, which comply with section 8.1 of CC Part 1.

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 *Access Control Policy* [43] to restrict the ability to *modify* [44] *security attributes for memories*[45] to the authorised identified roles. [46]

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

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

[43] [assignment: *access control SFP(s), information flow control SFP(s)*]

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

[45] [assignment: *list of security attributes*]

[46] [assignment: *the authorised identified roles*]

Application Note:	List of security attributes for memories and authorised identified roles is given in the full Security Target.

FMT_MSA.1/SFR	**Management of security attributes - Hardware components**
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 *Access control Policy* [47] to restrict the ability to *modify* [48] *the security attributes S2S_APACTRL, S2S_APUCTRL, S2S_SVACTRL, S2S_SVUCTRL*[49] to the authorised identified roles.[50]

This Security Target performs two iteration operations on FMT_MSA.3 and also one selection and two assignment operations on each iteration, which comply with section 8.1 of CC Part 1.

FMT_MSA.3/MEM	**Static attribute initialisation - 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 *Access Control Policy* [51] to provide *restrictive* [52] default values for security attributes that are used to enforce the SFP.
Application Note:	Restrictive default values of security attributes for memories are given in the full Security Target.
FMT_MSA.3.2/MEM	The TSF shall allow the *no subject* [53] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3/SFR	**Static attribute initialisation - Hardware components**
Hierarchical to:	No other components
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/SFR	The TSF shall enforce the *Access Control Policy* [54] to provide *restrictive* [55] default values for security attributes that are used to enforce the SFP.

[47] [assignment: *access control SFP(s), information flow control SFP(s)*]
[48] [selection: *change_default, query, modify, delete [assignment: other operations]*]
[49] [assignment: *list of security attributes*]
[50] [assignment: *the authorised identified roles*]
[51] [assignment: *access control SFP, information flow control SFP*]
[52] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]
[53] [assignment: *the authorised identified roles*]
[54] [assignment: *access control SFP, information flow control SFP*]
[55] [selection, choose one of: *restrictive, permissive, [assignment: other property]*]

Application Note: Restrictive default values of security attributes S2S_APACTRL, S2S_APUCTRL, S2S_SVACTRL, S2S_SVUCTRL are given in the full Security Target.

FMT_MSA.3.2/SFR The TSF shall allow the *no subject* [56] to specify alternative initial values to override the default values when an object or information is created.

This Security Target performs two assignment operations on FMT_SMF.1, which comply with section 8.1 of CC Part 1 1.

FMT_SMF.1 **Specification of Management Functions**

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- *Transformations in system operation modes for subjects CPU_ovBSY, DMA_ovBSY and PKC_ovBSY*
- *Change in the CPU privilege level for subject(s) CPU_ovBSY* [57]

Application Note: The conditions for the management functions are given in the full Security Target.

Additional SFR regarding cryptographic functionality

This Security Target performs the following iterations on FCS_COP.1, which are in addition to those already done in sections 7.4.1 and 7.4.2 of the Protection Profile [5].

FCS_COP.1/GCM **Cryptographic operation - GCM support**

Hierarchical to: No other components.

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

FCS_COP.1.1/GCM The TSF shall perform *multiplication operation on blocks and incrementing function* [58] in accordance with a specified cryptographic algorithm *Galois/Counter Mode (GCM) and GMAC* [59] and cryptographic key sizes *none* [60] that meet the following: *NIST SP 800-38D* [51] [61].

FCS_COP.1/CRC **Cryptographic operation - CRC**

Hierarchical to: No other components.

[56] [assignment: *the authorised identified roles*]
[57] [assignment: *list of management functions to be provided by the TSF*]
[58] [assignment: *list of cryptographic operations*]
[59] [assignment: *list of cryptographic algorithm*]
[60] [assignment: *cryptographic key sizes*]
[61] [assignment: *list of standards*]

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

FCS_COP.1.1/CRC The TSF shall perform *calculation of cyclic redundancy checks* [62] in accordance with a specified cryptographic algorithm *CRC-8, CRC-16 and CRC-32* [63] and cryptographic key sizes *none* [64] that meet the following:

- *ITU-T I.432.1* [64] *(for CRC-8)*
- *ITU-T V.42* [65] *and ITU-T X.25* [66] *(for CRC-16)*
- *ITU-T V.42* [65] *and IEEE 802.3* [67] *(for CRC-32)*

[65].

The TSF provides cryptographic functionality to help satisfy several high-level security objectives. In order for a cryptographic operation to function correctly, the operation must be performed in accordance with a specified algorithm and with a cryptographic key of a specified size. The following Functional Requirements to the TOE can be derived from this CC component:

FCS_COP.1/SW_AES **Cryptographic operation - AES**

Hierarchical to: No other components.

FCS_COP.1.1/SW_AES The TSF shall perform *decryption and encryption* [66]in accordance with a specified cryptographic algorithm *AES in ECB, CBC, CFB, CTR, GCM, CBC-MAC, CCM, OFB and CMAC* [67]and cryptographic key sizes *128, 192 and 256 bit* [68]that meet the following *FIPS 197* [58], *NIST SP 800-38A (ECB, CBC, CFB, CTR and OFB modes)* [47], *NIST SP 800-38B (CMAC mode)* [49], *NIST SP 800-38C (CCM mode)* [50], *NIST SP 800-38D (GCM mode), and* [51],*ISO 9797-1, Algorithm 1 (CBC-MAC mode)* [53] [69]

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

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

FCS_COP.1/SW_DES **Cryptographic operation - TDES**

Hierarchical to: No other components.

[62] [assignment: *list of cryptographic operations*]

[63] [assignment: *list of cryptographic algorithm*]

[64] [assignment: *cryptographic key sizes*]

[65] [assignment: *list of standards*]

[66] [assignment: *list of cryptographic operations*]

[67] [assignment: *cryptographic algorithm*]

[68] [assignment: *cryptographic key sizes*]

[69] [assignment: *list of standards*]

FCS_COP.1.1/SW_DES The TSF shall perform *encryption and decryption* [70]in accordance with a specified cryptographic algorithm *Triple-DES in ECB, CBC, CFB, CTR, CBC-MAC, RetailMAC, OFB and CMAC* [71]and cryptographic key sizes *2-key TDES (112 bit) and 3-key TDES (168 bit)* [72]that meet the following *NIST SP 800-67* [79], *NIST Special Publication 800-38A, 2001 (ECB, CBC, CFB and CTR mode)* [47], *ISO 9797-1, Algorithm 1 (CBC-MAC mode)* [53], , *ISO 9797-1 Algorithm 3 (RetailMAC)* [53], *and NIST Special Publication 800-38B (CMAC mode)* [49] [73].

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9]. To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards).

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

FCS_COP.1/RSA **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/RSA The TSF shall perform *encryption, decryption, signature and verification* [74]in accordance with the specified cryptographic algorithm *RSA* [75]and cryptographic key sizes *512 bits to 4096 bits* [76]that meet the following: *PKCS #1, v2.2: RSAEP, RSADP, RSASP1, RSAVP1 and FIPS PUB 186-4-2013* [57] [77].

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

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

FCS_COP.1/RSA_PAD **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/RSA_PAD The TSF shall perform *message and signature encoding methods*[78] in accordance with the specified

[70] [assignment: *list of cryptographic operations*]
[71] [assignment: *cryptographic algorithm*]
[72] [assignment: *cryptographic key sizes*]
[73] [assignment: *list of standards*]
[74] [assignment: *list of cryptographic operations*]
[75] [assignment: *cryptographic algorithm*]
[76] [assignment: *cryptographic key sizes*]
[77] [assignment: *list of standards*]
[78] [assignment: *list of cryptographic operations*]

cryptographic algorithm *EME-OAEP and EMSA-PSS*[79] and cryptographic key sizes *512 bits to 4096 bits*[80] that meet the following: *PKCS #1, v2.2: EME-OAEP, EMSA-PSS* [69] *and PKCS #1 Padding v1.5:* [70] [81].

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

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

FCS_COP.1/RSA_PubExp **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/RSA_PubExp The TSF shall perform *public key computation from a RSA CRT private key* [82]in accordance with the specified cryptographic algorithm *RSA* [83]and cryptographic key sizes *512 bits to 4096 bits* [84]that meet the following: *PKCS #1, v2.2* [69]*and FIPS PUB 186-4-2013* [57] [85].

Application Notes:

(1) The security functionality is resistant against side channel analysis and other attacks described in [9].

(2) The computation will result in the generation of a public RSA key from the private key (in CRT format). As this key is implied by the private key, this is not true key generation, and, to prevent duplication in this ST, this has not been included as a separate FCS_CKM.1 SFR.

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

FCS_COP.1/ECDSA **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/ECDSA The TSF shall perform *signature generation and verification*[86] in accordance with the specified cryptographic algorithm *ECDSA and ECC over GF(p)*[87] and cryptographic key sizes *128 to 640 bits*[88] that

[79] [assignment: *cryptographic algorithm*]
[80] [assignment: *cryptographic key sizes*]
[81] [assignment: *list of standards*]
[82] [assignment: *list of cryptographic operations*]
[83] [assignment: *cryptographic algorithm*]
[84] [assignment: *cryptographic key sizes*]
[85] [assignment: *list of standards*]
[86] [assignment: *list of cryptographic operations*]
[87] [assignment: *cryptographic algorithm*]
[88] [assignment: *cryptographic key sizes*]

meet the following: *ISO/IEC 14888-3-2015* [61] , *ANSI X9.62-1998* [68] , *and FIPS PUB 186-4-2013* [57] [89].

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

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

FCS_COP.1/ECC_DHKE **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/ECC_DHKE The TSF shall perform *Diffie-Hellman Key Exchange*[90] in accordance with the specified cryptographic algorithm *ECC over GF(p)*[91] and cryptographic key sizes *128 to 640 bits*[92] that meet the following: *ISO/IEC 11770-3-2015* [62] , *and ANSI X9.63* [80] [93].

Application Notes:

(1) The security functionality is resistant against side channel analysis and other attacks described in [9].

(2) The security functionality does not provide the complete key exchange procedure, but only the point multiplication which is used for the multiplication of the private key with the communication partner’s public key. Therefore this function can be used as part of a Diffie-Hellman key exchange as well pure point multiplication.

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

FCS_COP.1/ECC_Add **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/ECC_Add The TSF shall perform *a full point addition*[94] in accordance with a specified cryptographic algorithm *ECC over GF(p)*[95] and cryptographic key sizes *128 to 640 bits*[96] that meet the following: *ISO/IEC 15946-1-2008* [63] [97]

[89] [assignment: *list of standards*]
[90] [assignment: *list of cryptographic operations*]
[91] [assignment: *cryptographic algorithm*]
[92] [assignment: *cryptographic key sizes*]
[93] [assignment: *list of standards*]
[94] [assignment: *list of cryptographic operations*]
[95] [assignment: *cryptographic algorithm*]
[96] [assignment: *cryptographic key sizes*]
[97] [assignment: *list of standards*]

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

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

FCS_COP.1/ECDAA **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/ECDAA The TSF shall perform *the TPM 2.0 ECDAA signature function* [98] in accordance with the specified cryptographic algorithm *ECC over GF(p)*[99] and cryptographic key sizes *128 to 640 bits*[100] that meet the following: *TPM Rev. 2.0* [77]

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

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

FCS_COP.1/SHA **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/SHA The TSF shall perform *hashing*[101] in accordance with a specified cryptographic algorithm *SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512, SHAKE-128 and SHAKE-256* [102] and cryptographic key sizes *none*[103] that meet the following: *FIPS PUB 180-4-2011* [56] *and FIPS PUB 202-2015* [60][104].

Application Notes: 1) The security functionality is resistant against side channel analysis and timing attacks as described in [9].

(2) The length of the data to hash has to be a multiple of one byte. Arbitrary bit lengths are not supported.

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

[98] [assignment: *list of cryptographic operations*]
[99] [assignment: *cryptographic algorithm*]
[100] [assignment: *cryptographic key sizes*]
[101] [assignment: *list of cryptographic operations*]
[102] [assignment: *cryptographic algorithm*]
[103] [assignment: *cryptographic key sizes*]
[104] [assignment: *list of standards*]

FCS_COP.1/HMAC	**Cryptographic operation**
Hierarchical to:	No other components.
FCS_COP.1.1/HMAC	The TSF shall perform *keyed-hash message authentication code calculation*[105] in accordance with a specified cryptographic algorithm *SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384 or SHA-3/512*[106] and cryptographic key sizes *none*[107] that meet the following: *FIPS PUB 198-1-2008* [59] *and FIPS PUB 202-2015* [60][108]
Application Notes:	The security functionality is resistant against side channel analysis and other attacks described in [9].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction.

FCS_COP.1/EDDSA	**Cryptographic operation**
Hierarchical to:	No other components.
FCS_COP.1.1/EDDSA	The TSF shall perform *signature generation and verification*[109] in accordance with the specified cryptographic algorithm *EdDSA for twisted Edwards curves over GF(p) including the cryptographic algorithms Ed25519 and Ed448*[110] and cryptographic key sizes *128 to 640 bits*[111] that meet the following: *IETF RFC 8032* [71] *and IETF RFC 8032* [71] *except for the scalar calculation*[112].
Application Notes:	(1) The cryptographic key size refers to the bit length of the prime p. (2) The security functionality is resistant against side channel analysis and other attacks described in [9]. (3) For signature generation according to IETF RFC 8032 except for the scalar calculation, the scalar used for the point multiplication is chosen randomly and not computed from the private key. The generated signature is non-deterministic. The verification of the generated signature will pass successfully.

[105] [assignment: *list of cryptographic operations*]
[106] [assignment: *cryptographic algorithm*]
[107] [assignment: *cryptographic key sizes*]
[108] [assignment: *list of standards*]
[109] [assignment: *list of cryptographic operations*]
[110] [assignment: *cryptographic algorithm*]
[111] [assignment: *cryptographic key sizes*]
[112] [assignment: *list of standards*]

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

FCS_COP.1/MONT_DHKE **Cryptographic operation**

Hierarchical to: No other components.

FCS_COP.1.1/MONT_DHKE The TSF shall perform *Diffie-Hellman Key Exchange*[113] in accordance with the specified cryptographic algorithms *Curve25519 and Curve448 and generalizations thereof for a wider class Montgomery curves over GF(p)*[114] and cryptographic key sizes *128 to 640 bits*[115] that meet the following: *IETF RFC 7748* [72] [116].

Application Notes:

(1) The cryptographic key size refers to the bit length of the prime p..

(2) The security functionality is resistant against side channel analysis and other attacks described in [9].

(3) The security functionality does not provide the complete key exchange procedure, but only an x-only point multiplication of a secret scalar derived from the passed private key with the communication partner's passed public key. Therefore this function can be used as part of a Diffie-Hellman key exchange as well pure point multiplication.

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

FCS_CKM.5/KDF **Cryptographic key derivation**

Hierarchical to: No other components.

FCS_CKM.5/KDF The TSF shall derive cryptographic keys *session key* [117] from *a shared secret*[118] in accordance with a specified cryptographic key derivation algorithm *ANSI X9.63 Key Derivation Function* [119] and cryptographic key sizes *128 bits to 256 bits* [120] that meet the following: *ANSI X9.63* [80] [121].

[113] [assignment: *list of cryptographic operations*]
[114] [assignment: *cryptographic algorithm*]
[115] [assignment: *cryptographic key sizes*]
[116] [assignment: *list of standards*]
[117] [assignment: *key type*]
[118] [assignment: *input parameters*]
[119] [assignment: *cryptographic key derivation algorithm*]
[120] [assignment: *cryptographic key sizes*]
[121] [assignment: *list of standards*]

Application Notes:	The security functionality is resistant against side channel analysis and other attacks described in [9].
Dependencies:	[FCS_CKM.2 Cryptographic Key Distribution, or FCS_COP.1 Cryptographic Operation] FCS_CKM.4 Cryptographic Key Destruction.

FCS_COP.1/EUICC	**Cryptographic operation**
Hierarchical to:	No other components.
FCS_COP.1.1/EUICC	The TSF shall perform *eUICC authentication functions*[122] in accordance with the specified cryptographic algorithms *MILENAGE, TUAK and CAVE*[123] and cryptographic key sizes *none*[124] that meet the following: *3GPP TS 35.205/35.206 (MILENAGE)* [73] [74], *3GPP TS 35.231 (TUAK)* [75], *3GPP2 S.S0053-0 (CAVE)* [76] [125].
Application Notes:	The security functionality is resistant against side channel analysis and other attacks described in [9].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction.

FCS_COP.1/SW_CRC	**Cryptographic operation - CRC**
Hierarchical to:	No other components.
FCS_COP.1.1/SW_CRC	The TSF shall perform *calculation of cyclic redundancy checks* [126] in accordance with a specified cryptographic algorithm *CRC-16 and CRC-32* [127] and cryptographic key sizes *none* [128] that meet the following: *ITU-T X.25* [66] *(for CRC-16) and IEEE 802.3* [67] *(for CRC-32)* [129].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction.

The TSF provides functionality to generate a variety of key pairs. In order for the key generation to function correctly, the operation must be performed in accordance with a specified standard and with cryptographic key sizes out of a specified range. The

[122] [assignment: *list of cryptographic operations*]
[123] [assignment: *cryptographic algorithm*]
[124] [assignment: *cryptographic key sizes*]
[125] [assignment:*list of standards*]
[126] [assignment: *list of cryptographic operations*]
[127] [assignment: *cryptographic algorithm*]
[128] [assignment: *cryptographic key sizes*]
[129] [assignment: *list of standards*]

following Security Functional Requirements to the TOE can be derived from this CC component:

FCS_CKM.1/RSA **Cryptographic Key Generation**

Hierarchical to: No other components.

FCS_CKM.1.1/RSA The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *RSA* [130]and specified cryptographic key sizes *512-4096 bits* [131]that meet the following: *PKCS #1, v2.2, FIPS PUB 186-4-2013* [57] [132].

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1/ECC **Cryptographic Key Generation**

Hierarchical to: No other components.

FCS_CKM.1.1/ECC The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *ECDSA (ECC over GF(p))* [133]and specified cryptographic key sizes *128 to 640bits* [134]that meet the following: *ISO/IEC 15946-1-2008* [63], *ANSI X9.62-1998* [68] *and FIPS PUB 186-4-2013* [57] . [135]

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1/EDDSA **Cryptographic Key Generation**

Hierarchical to: No other components.

FCS_CKM.1.1/EDDSA The TSF shall generate keys pairs in accordance with a specified cryptographic algorithm *EdDSA for twisted Edwards curves over GF(p), including the cryptographic algorithms Ed25519 and Ed448*[136] and specified cryptographic key sizes *128 to 640 bits*[137] that meet the following: *IETF RFC 8032* [71] [138].

[130] [assignment: *cryptographic key generation algorithm*]
[131] [assignment: *cryptographic key sizes*]
[132] [assignment: *list of standards*]
[133] [assignment: *cryptographic algorithm*]
[134] [assignment: *cryptographic key sizes*]
[135] [assignment: *list of standards*]
[136] [assignment: *cryptographic key generation algorithm*]
[137] [assignment: *cryptographic key sizes*]
[138] [assignment: *list of standards*]

Application Notes:

(1) The cryptographic key size refers to the bit length of the prime p. .

(2) The security functionality is resistant against side channel analysis and other attacks described in [9].

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1/MONT **Cryptographic Key Generation**

Hierarchical to: No other components.

FCS_CKM.1.1/MONT The TSF shall generate key pairs for Diffie-Hellman Key Exchange in accordance with a specified cryptographic algorithm *Curve25519 and Curve448 and generalizations thereof for a wider class Montgomery curves over GF(p)* [139] and specified cryptographic key sizes *128 to 640 bits*[140] that meet the following: *IETF RFC 7748* [71] [141].

Application Notes: The security functionality is resistant against side channel analysis and other attacks described in [9].

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

The Residual information protection addresses the need to ensure that information in a resource is no longer accessible when the resource is deallocated, and that therefore newly created objects do not contain information that was accidentally left behind in the resources used to create the objects. The following Functional Requirement to the TOE can be derived from the CC component FDP_RIP.1:

FDP_RIP.1 **Subset Residual Information Protection**

Hierarchical to: No other components.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from*[142] the following objects: *any memory resources used by the Crypto Library that contained temporary or secret values* [143].

Dependencies: No dependencies.

Note 6. The TSF ensures that, upon exit from each function, with the exception of input parameters, return values or locations where it is explicitly documented that values remain at specific addresses, any memory resources used by that function that contained temporary or secret values are cleared.

[139] [assignment: *cryptographic key generation algorithm*]

[140] [assignment: *cryptographic key sizes*]

[141] [assignment: *list of standards*]

[142] [selection: *allocation of the resource to, deallocation of the resource from*]

[143] [assignment: *list of objects*]

FCS_CKM.4/CL **Cryptographic Key Destruction - Crypto Library**

Hierarchical to: No other components.

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *o verwrite*[144] that meets the following: *ISO 11568-4-2007* [54] [145]

Application Notes: The TOE provides the Security IC Embedded Software with library calls to perform various cryptographic algorithms that involve keys (e.g., AES, DES, RSA, etc.). Through the parameters of the library calls the Security IC Embedded Software provides keys for the cryptographic algorithms. To perform its cryptographic algorithms the library copies these keys, or a transformation thereof, to the working-buffer (supplied by the Security IC Embedded Software) and/or the memory/special function registers of the TOE. Depending upon the algorithm the library either overwrites these keys before returning control to the Security IC Embedded Software or provides a library call to through which the Security IC Embedded Software can clear these keys. In the case of a separate library call to clear keys the guidance instructs the Security IC Embedded Software when/how this call should be used.

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

Note: Clearing of keys that are provided by the Security IC Embedded Software to the Crypto Library is the responsibility of the Security IC Embedded Software.

The TOE shall meet the requirements “Random number generation” as specified below.

The hardware part of the TOE provides a physical random number generator (RNG) that fulfils FCS_RNG.1/PTG.2 as defined in Section 6.1.2. The additional software part of the TOE (Crypto Library) implements a software (pseudo) RNG that fulfils FCS_RNG.1/HYB-DET (see below). This software RNG obtains its seed from the hardware RNG, after the TOE (Crypto Library) has performed a self test of the hardware RNG.

FCS_RNG.1/HYB-DET **Random number generation**

Hierarchical to: No other components.

FCS_RNG.1.1/HYB-DET The TSF shall provide a *hybrid deterministic*[146] random number generator that implements:

(K.4.1) a chi-squared test on the seed generator.

[144] [assignment: *cryptographic key destruction method*]
[145] [assignment: *list of standards*]
[146] [selection: *physical, hybrid physical, hybrid deterministic*]

(DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 (as defined in [6]*) as random source.*

(DRG.4.2) The RNG provides forward secrecy (as defined in [6]*).*

(DRG.4.3) The RNG provides backward secrecy even if the current internal state is known (as defined in [6]).

(DRG.4.4) The RNG provides enhanced forward secrecy on demand (as defined in [6]).

(DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2 (as defined in [6]). [147]

FCS_RNG.1.2/HYB-DET

The TSF shall provide *random numbers*[148] that meet:

(K.4.2) class K.4 of AIS20 [8].

(DRG.4.6) The RNG generates output for which 2^{48} *strings of bit length 128 are mutually different with probability at least* $1 – 2^{-24}$*.*

(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [6]).[149]

Application Notes:

(1) The security functionality is resistant against side channel analysis and similar techniques.

(2) The TOE provides the Security IC Embedded Software with separate library calls to initialise the random number generator (which includes the chi-squared test) and to generate random data. The user can call an initialisation function upon use of the random number generator.

Dependencies:

No dependencies.

Note:

Only if the chi-squared test succeeds the hardware RNG seeds the software RNG implemented as part of the Crypto Library (as part of security functionality SS.SW_ RNG).

The Crypto Library does not prevent the operating system from accessing the hardware RNG. If the hardware RNG is used by the operating system directly, it has to be decided based on the Security IC Embedded Software's security needs, what kind of test has to be performed and what requirements will have to be applied for this test. In this case the developer of the Security IC Embedded Software must ensure that the conditions

[147] [assignment: *list of security capabilities*]

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

[149] [assignment: *assignment: a defined quality metric*]

prescribed in the Guidance, Delivery and Operation Manual for the TOE are met.

The software (pseudo) RNG, which is implemented in the software part of the TOE (Crypto Library), fulfils FCS_RNG.1/HYB-PHY (see below) with a certain limitation. This limitation can be given by the Security IC Embedded Software. For details on the limitation please refer the user guidance documentation of the Crypto Library [15].

FCS_RNG.1/HYB-PHY **Random number generation**

Hierarchical to: No other components.

FCS_RNG.1.1/HYB-PHY The TSF shall provide a *hybrid physical*[150] random number generator that implements:

(PTG.3.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure has been detected no random numbers will be output.

(PTG.3.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source.

(PTG.3.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG is started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test and the seeding of the DRG.3 postprocessing algorithm have been finished successfully or when a defect has been detected.

(PTG.3.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.

(PTG.3.5) The online test procedure checks the raw random number sequence. It is triggered continuously [151]*. The online test is suitable for detecting nontolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.*

(PTG.3.6) The algorithmic post-processing algorithm belongs to Class DRG.3 with cryptographic state transition function and cryptographic output function, and the output data rate of the post-processing algorithm shall not exceed its input data rate.

FCS_RNG.1.2/HYB-PHY The TSF shall provide *numbers* [152] that meet:

[150] [selection: *physical, hybrid physical, hybrid deterministic*]

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

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

(PTG.3.7) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [6]).

(PTG.3.8) The internal random numbers shall use PTRNG of class PTG.2 as random source for the post-processing[153] ,[154]

FDP_SOP.1/Copy	**Secure Basic Operations**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SOP.1.1/Copy	The TSF shall provide basic operations *Copy*[155] on objects stored in the TOE. The basic operation is applied between objects stored in *ROM, RAM and Flash*[156] and *RAM*[157].
FDP_SOP.1.2/Copy	The TSF shall protect the data against attacks from *disclosure and modification*[158] that can be inherently applied during the processing of the basic operations.
Application Notes:	The security functionality is secured against fault attacks and non-differential side channel attacks. Please note that non-differential side channel attacks also include profiled non-differential attacks like standard template attacks.

FDP_SOP.1/Compare	**Secure Basic Operations**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SOP.1.1/Compare	The TSF shall provide basic operations *Compare*[159] on objects stored in the TOE. The basic operation is applied between objects stored in *ROM, RAM and Flash*[160] and *ROM, RAM and Flash*[161].
FDP_SOP.1.2/Compare	The TSF shall protect the data against attacks from *disclosure and modification*[162] that can be inherently applied during the processing of the basic operations.
Application Notes:	The security functionality is secured against fault attacks and non-differential side channel attacks. Please note that non-differential side channel attacks also include

[153] [selection: *use PTRNG of class PTG.2 as random source for the post-processing, have [assignment: work factor], require [assignment: guess work]*]

[154] [assignment: *a defined quality metric*]

[155] [selection: *Copy, Move, Compare*]

[156] [assignment: *list of memory locations*]

[157] [assignment: *list of memory locations*]

[158] [selection: *disclosure, modification*]

[159] [selection: *Copy, Move, Compare*]

[160] [assignment: *list of memory locations*]

[161] [assignment: *list of memory locations*]

[162] [selection: *disclosure, modification*]

profiled non-differential attacks like standard template attacks.

FDP_SOP.1/Arith_op **Secure Basic Operations**

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SOP.1.1/Arith_op The TSF shall provide basic operations *Arithmetic operations (secure modular addition, modular subtraction, modular multiplication, modular inversion, arithmetic comparison and exact addition)*[163] on objects stored in the TOE. The basic operation is applied between objects stored in *ROM, RAM and Flash*[164] and *ROM, RAM and Flash*[165].

FDP_SOP.1.2/Arith_op The TSF shall protect the data against attacks from *disclosure and modification*[166] that can be inherently applied during the processing of the basic operations.

Application Notes: The security functionality is secured against fault attacks and non-differential side channel attacks. Please note that non-differential side channel attacks also include profiled non-differential attacks like standard template attacks.

6.2 Security Assurance Requirements for the TOE

Table 25 lists the security assurance requirements for the TOE. These security functional requirements are either copied from the Protection Profile [5] without modifications, or augmented from there, or newly added in this Security Target as indicated in column three of the table. This partly addresses Application Note 22.

Table 25. Security assurance requirements for the TOE

Name	Title	compared to PP
ADV_ARC.1	Security architectural description	as in PP
ADV_FSP.5	Complete semi-formal functional specification with additional error information	augmented from PP to EAL6
ADV_IMP.2	Complete mapping of the implementation representation of the TSF	augmented from PP to EAL6
ADV_INT.3	Minimally complex internals	added for EAL6
ADV_SPM.1	Formal TOE security policy model	added for EAL6
ADV_TDS.5	Complete semiformal modular design	augmented from PP to EAL6
AGD_OPE.1	Operational user guidance	as in PP
AGD_PRE.1	Preparative procedures	as in PP

[163] [selection: *Copy, Move, Compare*]
[164] [assignment: *list of memory locations*]
[165] [assignment: *list of memory locations*]
[166] [selection: *disclosure, modification*]

Table 25. Security assurance requirements for the TOE *...continued*

Name	Title	compared to PP
ALC_CMC.5	Advanced support	augmented from PP to EAL6
ALC_CMS.5	Development tools CM coverage	augmented from PP to EAL6
ALC_DEL.1	Delivery procedures	as in PP
ALC_DVS.2	Sufficiency of security measures	as in PP
ALC_FLR.1	Basic flaw remediation	not in PP, added for EAL6+
ALC_LCD.1	Developer defined life-cycle model	as in PP
ALC_TAT.3	Compliance with implementation standards - all parts	augmented from PP to EAL6
ASE_CCL.1	Conformance claims	as in PP
ASE_ECD.1	Extended components definition	as in PP
ASE_INT.1	ST introduction	as in PP
ASE_OBJ.2	Security objectives	as in PP
ASE_REQ.2	Derived security requirements	as in PP
ASE_SPD.1	Security problem definition	as in PP
ASE_TSS.2	TOE summary specification with architectural design summary	augmented from PP to EAL6+
ATE_COV.3	Rigorous analysis of coverage	augmented from PP to EAL6
ATE_DPT.3	Testing: modular design	augmented from PP to EAL6
ATE_FUN.2	Ordered functional testing	augmented from PP to EAL6
ATE_IND.2	Independent testing - sample	as in PP
AVA_VAN.5	Advanced methodical vulnerability analysis	as in PP

This Security Target performs an assignment operation on ADV_SPM.1 as follows.

ADV_SPM.1 **Formal TOE security policy model**

ADV_SPM.1.1D: The developer shall provide a formal security policy model for the

- *Access Control Policy of the TOE according to FDP_ACC.1/MEM, FDP_ACF.1/MEM, FMT_MSA.1/MEM, FMT_MSA.3/MEM as well as FDP_ACC.1/SFR, FDP_ACF.1/SFR, FMT_MSA.1/SFR, FMT_MSA.3/SFR and also FMT_SMF.1*[167]

All refinements in section 6.2.1 of the Protection Profile [5] to security assurance requirements in Table 25, which are copied from the Protection Profile without modifications, entirely apply to this Security Target.

[167] [assignment: list of policies that are formally modelled]

All refinements in section 6.2.1 of the Protection Profile [5] to security assurance requirements in Table 25, which are augmented from the Protection Profile, are discussed below in their applicability to this Security Target. This addresses Application Note 23 in the Protection Profile [5].

Refinements regarding ADV_FSP

Refinement no. 215 to ADV_FSP.4 in the Protection Profile [5] is not relevant for this Security Target since the TOE does not embed IC Dedicated Test Software.

The Factory OS is not considered as IC Dedicated Test Software but instead as IC Dedicated Support Software since it is **not** only used to support testing of the TOE during production and **does** provide security functionality to be used after TOE delivery, which both contradicts to abstract 12 on page 8 of the Protection Profile [5]. However, the Factory OS provides testing capabilities for production testing and analysis of field returns, which is under restricted access to NXP and not for usage by the Composite Product Manufacturer. Therefore, these testing capabilities are considered as "test tool", which don't have to be described in the Functional Specification, but only be evaluated against their abuse after TOE delivery. Apart from that the Factory OS provides the Composite Product Manufacturer with some basic functional testing of SN220_SE and also with a readout of the identification flags of SN220_SE from System Page Common, which must be described in the Functional Specification.

Refinements no. 216, no. 217 and no. 218 to ADV_FSP.4 in the Protection Profile [5] are entirely applicable to ADV_FSP.5 since the refinements clarify the scope of the functional specification, and ADV_FSP.5 adds to this scope in accordance with the refinements.

Refinements regarding ADV_IMP

Refinement no. 223 to ADV_IMP.1 in the Protection Profile [5] is redundant since it is implicitly covered by the augmentation to ADV_IMP.2. First, ADV_IMP.2 requires the developer to provide the mapping between the TOE design description and the entire implementation representation instead of a sample of it only as in ADV_IMP.1. Second, ADV_IMP.2 requires the evaluator to confirm that, for the entire implementation representation and not only for a sample of it as in ADV_IMP.1, the information provided meets all requirements for content and presentation of evidence.

Refinements regarding ALC_CMC

Refinement no. 205 to ALC_CMC.4 in the Protection Profile [5] is entirely applicable to ALC_CMC.5 since the refinement clarifies the scope of configuration items in ALC_CMC.4, and ALC_CMC.5 does not touch this scope.

Refinement no. 206 to ALC_CMC.4 in the Protection Profile [5] is entirely applicable to ADV_CMC.5 since the refinement details requirements on configuration management of the TOE for ALC_CMC.4, which are not subverted in ADV_CMC.5.

Refinements regarding ALC_CMS

Refinement no. 199 to ALC_CMS.4 in the Protection Profile [5] is a clarification of the configuration item "TOE implementation representation". Although NXP as the TOE manufacturer is providing the Security IC Embedded Software, this item is not relevant for the configuration list, as the Security IC Embedded Software is developed independently from the TOE.

Compared to ALC_CMS.4 component ALC_CMS.5 only adds the requirement for a new configuration items to be included in the configuration list. (ALC_CMS.5.1C). Therefore the refinement in the PP regarding ADV_CMS.4 can be applied without changes and is valid for ADV_CMS.5.

Refinements regarding ATE_COV

Refinements no. 226 and no. 227 to ATE_COV.2 in the Protection Profile [5] are entirely applicable to ATE_COV.3 since they define some particular requirements on the test coverage for ATE_COV.2, which are not subverted in ATE_COV.3.

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

Table 26 maps the Security Objectives for the TOE to the Security Functional Requirements for the TOE.

Table 26. Mapping of the Security Objectives for the TOE to the Security Functional Requirements for the TOE

Security Objective for the TOE	Security Functional Requirement of the TOE
O.Malfunction	FRU_FLT.2, FPT_FLS.1
O.Abuse-Func	FMT_LIM.1, FMT_LIM.2
	FRU_FLT.2, FTP_FLS.1
	FPT_PHP.3
	FDP_ITT.1, FPT_ITT.1, FDP_IFC.1
O.Phys-Probing	FPT_PHP.3
	FDP_SDC.1
O.Phys-Manipulation	FDP_SDI.2/FLT
	FPT_PHP.3
O.Leak-Inherent	FDP_ITT.1, FPT_ITT.1 , FDP_IFC.1
O.Leak-Forced	FRU_FLT.2, FPT_FLS.1
	FPT_PHP.3
	FDP_ITT.1, FPT_ITT.1, FDP_IFC.1
O.RND	FCS_RNG.1/PTG.2
	FRU_FLT.2, FPT_FLS.1
	FPT_PHP.3
	FDP_ITT.1, FPT_ITT.1 , FDP_IFC.1
O.Identification	FAU_SAS.1
O.TDES	FCS_COP.1/TDES
	FCS_CKM.4/TDES
O.AES	FCS_COP.1/AES
	FCS_CKM.4/AES
O.FLASH-INTEGRITY	FDP_SDI.2/AGE
O.GCM-SUPPORT	FCS_COP.1/GCM
O.CRC	FCS_COP.1/CRC
O.MEM-ACCESS	FDP_ACC.1/MEM

Table 26. Mapping of the Security Objectives for the TOE to the Security Functional Requirements for the TOE *...continued*

Security Objective for the TOE	Security Functional Requirement of the TOE
	FDP_ACF.1/MEM
	FMT_MSA.1/MEM
	FMT_MSA.3/MEM
	FMT_SMF.1
O.SFR-ACCESS	FDP_ACC.1/SFR
	FDP_ACF.1/SFR
	FMT_MSA.1/SFR
	FMT_MSA.3/SFR
	FMT_SMF.1

The green and blue colored cells in Table 26 show how the Protection Profile [5] maps its security objectives for the TOE to the Security Functional Requirements for the TOE, see section 6.3.1 and section 7.4.2. of the Protection Profile [5]. Green marks this for the mandatory security requirements of the protection profile, blue marks this for the augmentations. Section 6.3.1 of the Protection Profile [5] also gives the rationale for the mappings colored in green.

The justification related to security objective O.TDES is as follows:

O.TDES is met by FCS_COP.1/TDES and FCS_CKM.4/TDES since FCS_COP.1/TDES requests the TOE to implement the cryptographic service targeted in O.TDES according to approved public standards and FCS_CKM.4/TDES requests the TOE to implement a secure destruction method for its cryptographic key.

The justification related to security objective O.AES is as follows:

O.AES is met by FCS_COP.1/AES and FCS_CKM.4/AES since FCS_COP.1/AES requests the TOE to implement the cryptographic service targeted in O.AES according to approved public standards and FCS_CKM.4/AES requests the TOE to implement a secure destruction method for its cryptographic key.

The justification related to security objective O.MEM-ACCESS is as follows:

O.MEM-ACCESS is met by FDP_ACC.1/MEM, FDP_ACF.1/MEM, FMT_MSA.1/MEM, FMT_MSA.3/MEM and FMT_SMF.1 together.

FDP_ACC.1/MEM requests the TOE to enforce the Access Control Policy to its memories. FDP_ACF.1/MEM gives the rules for all access ports of the TOE versus system operation modes and CPU privilege levels, which must be applied to the objects, and also the dependencies of these rules on security attributes. FMT_MSA.1/MEM and FMT_MSA.3/MEM give the restrictions required on these security attributes. FMT_SMF.1 finally lists the rules for all access ports that make the TOE changing their system operation modes and CPU privilege levels.

The justification related to security objective O.SFR-ACCESS is as follows:

O.SFR-ACCESS is met by FDP_ACC.1/SFR, FDP_ACF.1/SFR, FMT_MSA.1/SFR, FMT_MSA.3/SFR and FMT_SMF.1 together.

FDP_ACC.1/SFR requests the TOE to enforce the Access Control Policy to its hardware components. FDP_ACF.1/SFR gives the rules for all access ports of the TOE versus system operation modes and CPU privilege levels, which must be applied to the objects, and also the dependencies of these rules on security attributes. FMT_MSA.1/MEM and FMT_MSA.3/MEM give the restrictions required on these security attributes. FMT_SMF.1 finally lists the rules for all access ports that make the TOE changing their system operation modes and CPU privilege levels.

The justification related to security objective O.FLASH-INTEGRITY is as follows:

O.FLASH-INTEGRITY is met by FDP_SDI.2/AGE for the following reason. O.FLASH-INTEGRITY targets to preserve integrity over life-time and FDP_SDI.2/AGE addresses this with a request to monitor integrity and either correct violations or indicate a wearout failure.

The justification related to security objective O.GCM-SUPPORT is as follows:

O.GCM-SUPPORT is met by FCS_COP.1/GCM since FCS_COP.1/GCM requests the TOE to implement the support for cryptographic services targeted in O.GCM-SUPPORT according to an approved public standard. No keys are used by the support for the cryptographic services.

The justification related to security objective O.CRC is as follows:

O.CRC is met by FCS_COP.1/CRC since FCS_COP.1/CRC requests the TOE to implement the cryptographic service targeted in O.CRC according approved public standards. No keys are used by the cryptographic service.

The rationale for the Security Functional Requirements for Crypto Library which are additional to the PP is described below.

Note 7. O.RND taken from the PP [5] is considered to be generic and therefore covers both hardware RNG and software RNG. For O.RND additional requirements (FCS_RNG.1/HYB-DET, and FCS_RNG.1/HYB-PHY) have been added. The explanation following Table 27 describes this in detail.

Table 27. Mapping of SFRs to Security Objectives for Crypto Library in this ST

Objective	TOE Security Functional Requirements
O.SW_AES	FCS_COP.1/SW AES
O.SW_DES	FCS_COP.1/SW DES
O.RSA	FCS_COP.1/RSA FCS_COP.1/RSA_Pad
O.RSA_PubExp	FCS_COP.1/RSA_PubExp
O.RSA_KeyGen	FCS_CKM.1/RSA
O.ECDSA	FCS_COP.1/ECDSA
O.ECC_DHKE	FCS_COP.1/ECC_DHKE
O.ECC_Add	FCS_COP.1/ECC_Add
O.ECC_KeyGen	FCS_CKM.1/ECC
O.ECDAA	FCS_COP.1/ECDAA
O.SHA	FCS_COP.1/SHA
O.HMAC	FCS_COP.1/HMAC
O.EDDSA	FCS_COP.1/EDDSA
O.EDDSA_KeyGen	FCS_CKM.1/EDDSA

Table 27. Mapping of SFRs to Security Objectives for Crypto Library in this ST*...continued*

Objective	TOE Security Functional Requirements
O.MONT_KeyGen	FCS_CKM.1/MONT
O.MONT_DHKE	FCS_COP.1/MONT_DHKE
O.EUICC	FCS_COP.1/EUICC
O.KDF	FCS_CKM.5/KDF
O.SW_CRC	FCS_COP.1/SW_CRC
O.COPY	FDP_SOP.1/Copy
O.COMPARE	FDP_SOP.1/Compare
O.ARITH_OP	FDP_SOP.1/Arith_op
O.REUSE	FDP_RIP.1 FCS_CKM.4/CL
O.RND	FCS_RNG.1/HYB-DET FCS_RNG.1/HYB-PHY

The justification of the security objectives O.SW_AES, O.SW_DES, O.RSA, O.RSA_PubExp, O.RSA_KeyGen, O.ECDSA, O.ECC_DHKE, O.ECC_Add, O.ECC_KeyGen, O.ECDAA, O.SHA, O.HMAC, O.EDDSA, O.EDDSA_KeyGen, O.MONT_KeyGen, O.MONT_DHKE, O.KDF, O.EUICC, O.COPY, O.COMPARE, O.ARITH_OP and O.SW_CRC are all as follows:

- Each objective is directly implemented by a single SFR specifying the (cryptographic) service that the objective wishes to achieve (see the above table for the mapping).
- The requirements and architectural measures that originally were taken from the Protection Profile [5] support the objective:
  - ADV.ARC.1 (and underlying platform SFRs) supports the objective by ensuring that the TOE works correctly (i.e., all of the TOE's capabilities are ensured) within the specified operating conditions and maintains a secure state when the TOE is outside the specified operating conditions. A secure state is also entered when perturbation or DFA attacks are detected.
  - ADV.ARC.1 (and underlying platform SFRs) ensures that no User Data (plain text data, keys) or TSF Data is disclosed when they are transmitted between different functional units of the TOE (i.e., the different memories, the CPU, cryptographic co-processors), thereby supporting the objective in keeping confidential data secret.
- ADV.ARC.1 (and underlying platform SFRs) by ensuring that User Data and TSF Data are not accessible from the TOE except when the Security IC Embedded Software decides to communicate them via an external interface.

The justification of the security objective O.REUSE is as follows:

- O.REUSE requires the TOE to provide procedural measures to prevent disclosure of memory contents that was used by the TOE. This applies to the SN220 Series - Secure Element with Crypto Library and is met by the SFR FDP_RIP.1 and FCS_CKM.4/CL, which requires the library to make unavailable all memory contents that has been used by it. Note that the requirement for residual information protection applies to all functionality of the Cryptographic Library.

The justification of the security objective O.RND is as follows:

- O.RND requires the TOE to generate random numbers with (a) ensured cryptographic quality (i.e. not predictable and with sufficient entropy) such that (b) information about the generated random numbers is not available to an attacker.
  1. Ensured cryptographic quality (sufficient entropy part) of generated random numbers is met by FCS_RNG.1.1/HYB-DET through the characteristic ‘hybrid deterministic’, by FCS_RNG.1.1/HYB-PHY through the characteristic ‘hybrid physical’, and by the random number generator meeting NIST SP 800-90A. Ensured cryptographic quality (not predictable part) of generated random numbers is met by FCS_RNG.1/HYB-DET through the characteristic ‘chi-squared test of the seed generator’, by FCS_RNG.1/HYB-PHY through the characteristic ‘cryptographic post-processing algorithm’, and FCS_RNG.1 from the certified hardware platform.
  2. Information about the generated random numbers is not available to an attacker is met through ADV.ARC.1, which prevent physical manipulation and malfunction of the TOE and support this objective because they prevent attackers from manipulating or otherwise affecting the random number generator.

6.3.2 Dependencies of Security Functional Requirements

The dependencies of the Security Functional Requirements for the TOE are given in Table 28.

Table 28. Dependencies of the Security Functional Requirements for the TOE

SFR of the TOE	Dependencies	Fullfilled by SFRs
FRU_FLT.2	FPT_FLS.1	FPT_FLS.1
FPT_FLS.1	none	N/A
FMT_LIM.1	FMT_LIM.2	FMT_LIM.2
FMT_LIM.2	FMT_LIM.1	FMT_LIM.1
FPT_PHP.3	none	N/A
FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1	FDP_IFC.1
FPT_ITT.1	none	N/A
FDP_IFC.1	FDP_IFF.1	N/A, see sec. 6.3.2 in PP [5]
FAU_SAS.1	none	N/A
FDP_SDC.1	none	N/A
FDP_SDI.2/AGE	none	N/A
FDP_SDI.2/FLT	none	N/A
FCS_RNG.1/PTG.2	none	N/A
FCS_COP.1/TDES	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/TDES
FCS_COP.1/AES	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/AES
FCS_COP.1/GCM	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	N/R, see item 2 below
FCS_COP.1/CRC	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	N/R, see item 2 below
FCS_CKM.4/TDES	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
FCS_CKM.4/AES	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below

Table 28. Dependencies of the Security Functional Requirements for the TOE *...continued*

SFR of the TOE	Dependencies	Fullfilled by SFRs
FDP_ACC.1/MEM	FDP_ACF.1	FDP_ACF.1/MEM
FDP_ACC.1/SFR	FDP_ACF.1	FDP_ACF.1/SFR
FDP_ACF.1/MEM	FDP_ACC.1	FDP_ACC.1/MEM
	FMT_MSA.3	FMT_MSA.3/MEM
FDP_ACF.1/SFR	FDP_ACC.1	FDP_ACC.1/SFR
	FMT_MSA.3	FMT_MSA.3/SFR
FMT_MSA.1/MEM	FDP_ACC.1 or FDP_IFC.1	FDP_ACC.1/MEM
	FMT_SMR.1	see item 3 below
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1/SFR	FDP_ACC.1 or FDP_IFC.1	FDP_ACC.1/SFR
	FMT_SMR.1	see item 3 below
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3/MEM	FMT_MSA.1	FMT_MSA.1/MEM
	FMT_SMR.1	see item 3 below
FMT_MSA.3/SFR	FMT_MSA.1	FMT_MSA.1/SFR
	FMT_SMR.1	see item 3 below
FMT_SMF.1	none	N/A
FCS_COP.1/SW_AES	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/SW_DES	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/RSA	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1/RSA (for keys if generated by this SFR) otherwise see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/RSA_PAD	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/RSA_PubExp	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/ECDSA	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1/ECC (for keys if generated by this SFR) otherwise see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/ECC_DHKE	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/ECC_Add	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/ECDAA	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/SHA	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	N/R, see item 2 below
FCS_COP.1/HMAC	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below

Table 28. Dependencies of the Security Functional Requirements for the TOE *...continued*

SFR of the TOE	Dependencies	Fullfilled by SFRs
	FCS_CKM.4	N/R, see item 2 below
FCS_COP.1/EDDSA	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1/EDDSA (for keys if generated by this SFR) otherwise see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/MONT_DHKE	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1/MONT (for keys if generated by this SFR) otherwise see item 1 below
	FCS_CKM.4	FCS_CKM.4/CL
FCS_COP.1/EUICC	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	N/R, see item 2 below
FCS_COP.1/SW_CRC	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	see item 1 below
	FCS_CKM.4	N/R, see item 2 below
FCS_CKM.1/RSA	FCS_CKM.2 or FCS_COP.1	FCS_COP.1/RSA
	FCS_CKM.4	FCS_CKM.4/CL
FCS_CKM.1/ECC	FCS_CKM.2 or FCS_COP.1	FCS_COP.1/ECDSA
	FCS_CKM.4	FCS_CKM.4/CL
FCS_CKM.1/EDDSA	FCS_CKM.2 or FCS_COP.1	FCS_COP.1/EDDSA
	FCS_CKM.4	FCS_CKM.4/CL
FCS_CKM.1/MONT	FCS_CKM.2 or FCS_COP.1	FCS_COP.1/MONT_DHKE
	FCS_CKM.4	FCS_CKM.4/CL
FCS_CKM.4/CL	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	FCS_CKM.1/RSA, FCS_CKM.1/ECC, FCS_CKM.1/EDDSA, FCS_CKM.1/MONT
FCS_CKM.5/KDF	FCS_CKM.2 or FCS_COP.1	N/R, see item 4 below
	FCS_CKM.4	FCS_CKM.4/CL
FDP_RIP.1	none	N/A
FCS_RNG.1/HYB-DET	none	N/A
FCS_RNG.1/HYB-PHY	none	N/A
FDP_SOP.1/Copy	none	N/A
FDP_SOP.1/Compare	none	N/A
FDP_SOP.1/Arith_op	none	N/A

1. The dependencies of Security Functional Requirements FCS_CKM.4/TDES, FCS_CKM.4/AES, FCS_COP.1/TDES, FCS_COP.1/AES, FCS_COP.1/CRC, FCS_COP.1/GCM, FCS_COP.1/SW_AES, FCS_COP.1/SW_DES, FCS_COP.1/RSA, FCS_COP.1/RSA_PAD, FCS_COP.1/RSA_PubExp, FCS_COP.1/ECDSA, FCS_COP.1/ECC_DHKE, FCS_COP.1.ECC_Add, FCS_COP.1/ECDAA, FCS_COP.1/SHA, FCS_COP.1/HMAC, FCS_COP.1/EDDSA, FCS_COP.1/MONT_DHKE, FCS_COP.1/EUICC and FCS_COP.1/SW_CRC on the Security Functional Requirements FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 are not considered in this Security Target if these SFRs are not fulfilled. This is because the decision on how to import user data and how to generate the keys shall be left to the Security IC Embedded Software.

2. The dependencies of Security Functional Requirements FCS_COP.1/CRC, FCS_COP.1/GCM, FCS_COP.1/SHA, FCS_COP.1/HMAC, FCS_COP.1/EUICC and FCS_COP.1/SW_CRC on Security Functional Requirements FCS_CKM.4 don't have to be considered in this Security Target since their operations do not need any cryptographic keys.
3. The dependencies of Security Functional Requirements FMT_MSA.1/MEM, FMT_MSA.3/MEM, FMT_MSA.1/SFR and FMT_MSA.3/SFR on FMT_SMR.1 are not considered in this Security Target. This is because the security attributes shall be managed by Security IC Embedded Software based on which the Security IC Embedded Software shall be capable to maintain roles and assign users to roles appropriate to its needs.
4. The dependencies of Security Functional Requirement FCS_CKM.5/KDF on FCS_CKM.2 or FCS_COP.1 are not considered in this Security Target if these SFRs are not fulfilled. This is because the decision on how a derived key is used shall be left to the Security IC Embedded Software.

6.3.3 Rationale for the Security Assurance Requirements

The Protection Profile [5] targets EAL4 augmented with ALC_DVS.2, and AVA_VAN.5 and also gives a rationale for this choice, which is entirely applicable to this Security Target.

This Security Target augments from EAL4 to EAL6 in order to meet increasing assurance expectations of digital signature applications and electronic payment systems on the resistance to attackers with high attack potential. The augmentations to EAL4 in the Protection Profile [5] are mandatory for EAL6.

This Security Target augments EAL6 with ALC_FLR.1 and ASE_TSS.2 for the following reasons.

ALC_FLR.1 is added to cover policies and procedures that are applied to track and correct flaws and to support surveillance of the TOE.

ASE_TSS.2 is chosen to give architectural information on the security functionality of the TOE, which enhances comprehensibility.

6.3.4 Dependencies of Security Assurance Requirements

The dependencies of the Security Assurance Requirements are given in Table 29. They are derived from Appendix C of CC [3] . The table indicates whether the SAR is directly or indirectly required. Only applicable dependencies from the highest level assurance components are considered.

Table 29. Dependencies of the Security assurance requirements

Name	Directly required	Indirectly required
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1	ADV_FSP.2
ADV_FSP.5	ADV_IMP.1, ADV_TDS.1	ADV_TDS.3, ALC_TAT.1
ADV_IMP.2	ADV_TDS.3, ALC_CMC.5, ALC_TAT.1	ADV_FSP.4, ALC_CMS.1, ALC_DVS.2, ALC_LCD.1,
ADV_INT.3	ADV_IMP.1, ADV_TDS.3, ALC_TAT.1	ADV_FSP.4,
ADV_SPM.1	ADV_FSP.4	ADV_TDS.1
ADV_TDS.5	ADV_FSP.5	ADV_IMP.1, ADV_TDS.3, ALC_TAT.1

Table 29. Dependencies of the Security assurance requirements *...continued*

Name	Directly required	Indirectly required
AGD_OPE.1	ADV_FSP.1	none
AGD_PRE.1	none	none
ALC_CMC.5	ALC_CMS.1, ALC_DVS.2, ALC_LCD.1	none
ALC_CMS.5	none	none
ALC_DEL.1	none	none
ALC_DVS.2	none	none
ALC_FLR.1	none	none
ALC_LCD.1	none	none
ALC_TAT.3	ADV_IMP.1	ADV_FSP.4, ADV_TDS.3
ASE_CCL.1	ASE_ECD.1, ASE_INT.1, ASE_REQ.1	none
ASE_ECD.1	none	none
ASE_INT.1	none	none
ASE_OBJ.2	ASE_SPD.1	none
ASE_REQ.2	ASE_ECD.1, ASE_OBJ.2	ASE_SPD.1
ASE_SPD.1	none	none
ASE_TSS.2	ADV_ARC.1, ASE_INT.1, ASE_REQ.1	ADV_FSP.2, ADV_TDS.1, ASE_ECD.1
ATE_COV.3	ADV_FSP.2, ATE_FUN.1	ADV_TDS.1, ATE_COV.1
ATE_DPT.3	ADV_ARC.1, ADV_TDS.4, ATE_FUN.1	ADV_FSP.5, ADV_IMP.1, ALC_TAT.1, ARE_COV.1
ATE_FUN.2	ATE_COV.1	ADV_FSP.2, ADV_TDS.1, ATE_FUN.1
ATE_IND.2	ADV_FSP.2, AGD_OPE.1, AGD_PRE.1, ATE_COV.1, ATE_FUN.1	ADV_TDS.1
AVA_VAN.5	ADV_ARC.1, ADV_FSP4, ADV_IMP.1, ADV_TDS.3, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1	ALC_TAT.1, ATE_COV.1, ATE_FUN.1

6.3.5 Security Requirements are Internally Consistent

The statement on internal consistency of security requirements in section 6.3.4 of the Protection Profile [5] entirely applies to this Security Target.

Security functional requirements FRU_FLT.2, FPT_FLS.1, FPT_PHP.3, FDP_SDC.1, FDP_SDI.2/FLT, FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, which meet security objectives O.Malfunction, O.Phys-Probing, O.Phys-Manipulation, O.Leak- Inherent and O.Leak-Forced, protect the whole security functionality of the TOE and with this also the cryptographic operations requested in all iterations on FCS_COP.1, related operations on keys as requested in the iterations on FCS_CKM.4 as well as the access control policy according to FMT_SMF.1 and both iterations on each of FDP_ACC.1, FDP_ACF.1, FMT_MAS.1 and FMT_MSA.3.

The iterations FDP_SDI.2/FLT and FDP_SDI.2/AGE on FCS_SDI.2 complement each other in protecting integrity since they both request security functionality that detects integrity violations. Therefore FDP_SDI.2/AGE also adds to O.Phys-Manipulation.

The iterations on FCS_COP and FCS_CKM do not conflict since they address different operations with different keys.

The two iterations on each FDP_ACC.1, FDP_ACF.1, FMT_MSA.1 and FMT_MSA.3 do not contradict as they are related to different objects and their requests on shared security attributes fit together.

7 TOE Summary Specification

7.1 Portions of the TOE Security Functionality

7.1.1 Security Functionality of the TOE

The TOE Security Functionality (TSF) is composed of Security Services (SS) and Security Features (SF). They together fulfill the security functional requirements for the TOE, which are identified in Section 6.1.

The Security Services of the TOE are summarized in Table 30 and described in Section 7.1.2 .

The Security Features of the TOE are summarized in Table 31 and described in Section 7.1.3..

The TOE also implements security functionality, which is not part of its Security Services and Security Features like the PKC coprocessor. Such security functionality isn't required to meet the security functional requirements for the TOE. Instead, it can be used by Security IC Embedded Software to implement further Security Services and Security Features.

Table 30. Security Services of the Hardware Security Target

Security Services	Name
SS.RNG	Random Number Generator
SS.TDES	Triple-DES coprocessor
SS.AES	AES coprocessor
SS.GCM	GCM coprocessor
SS.SBC	SBC interface functions
SS.CRC	CRC coprocessor
SS.SW_AES	Software AES
SS.SW_DES	Software DES
SS.RSA	RSA
SS.RSA_Pad	RSA Padding
SS.RSA_PublicExp	RSA Public Exponentiation
SS.KDF	KDF: Cryptographic Key Derivation
SS.ECDSA	Elliptic Curve Digital Signature Algorithm (ECDSA)
SS.ECC_DHKE	ECC Diffie Hellman Key Exchange
SS.ECC_Add	Full ECC Point Addition
SS.ECDAA	Elliptic Curve Direct Anonymous Attestation (ECDAA)
SS.RSA_KeyGen	RSA Key Generator
SS.ECC_KeyGen	ECC Key Generator
SS.SHA	Secure Hash Algorithms
SS.HMAC	HMAC

Table 30. Security Services of the Hardware Security Target...*continued*

Security Services	Name
SS.EDDSA	Edwards-curve Digital Signature Algorithm (EdDSA)
SS.EDDSA_KeyGen	EdDSA Key Generation
SS.MONT_KeyGen	MontDH Key Generation
SS.MONT_DHKE	MontDH Key Exchange
SS.EUICC	eUICC Authentication
SS.COPY	Secure Copy
SS.COMPARE	Secure Compare
SS.ARITH_OP	Secure arithmetic operations
SS.SW_CRC	Software CRC
SS.SW_RNG	Software RNG

Table 31. Security Features of the TOE

Security Features	Name
SF.OPC	Control of Operating Conditions
SF.PHY	Protection against Physical Manipulation
SF.LOG	Logical Protection
SF.FOS-USE	Factory OS use restrictions
SF.MEM-ACC	Memory Access Control
SF.SFR-ACC	Special Function Register Access Control
SF.FLASH-SVC	Flash Services
SF.Object_Reuse	Object Reuse

7.1.2 Security Services of the TOE

7.1.2.1 SS.RNG : Random Number Generator

SS.RNG serves Security IC Embedded Software with random numbers.

For this purpose SS.RNG implements a physical hardware Random Number Generator, which claims functionality class PTG2 of the pre-defined RNG classes in [6]. With this it is suitable for generation of signature key pairs, generation of session keys for symmetric encryption mechanisms, random padding bits, zero-knowledge proofs, generation of seeds for Digital Random Number Generation (DRNG).

The Random Number Generator fulfills the online test requirements defined in [6] and embeds hardware test functionality to detect hardware defects and quality issues of the random numbers.

This security functionality covers:

- FPT_PHP.3
- FCS_RNG.1/PTG.2

7.1.2.2 SS.TDES : Triple-DES coprocessor

SS.TDES serves Security IC Embedded Software with calculation of the Triple Data Encryption Algorithm (TDEA) based on the Data Encryption Standard (DES) as defined in [79].

For this purpose SS.TDES implements a Triple-DES coprocessor in hardware, which can be configured by the Security IC Embedded Software to calculate the Triple DES algorithm or the Triple DES inverse algorithm on blocks of 64 bits with selectable keying option 1 of two 56-bit keys or keying option 2 of three 56-bit keys according to [79]. The keys shall be provided by the Security IC Embedded Software.

This security functionality covers:

- FCS_COP.1/TDES

7.1.2.3 SS.AES : AES coprocessor

SS.AES serves Security IC Embedded Software with calculation of the Advanced Encryption Standard (AES) algorithm as defined in [58].

For this purpose SS.AES implements an AES coprocessor in hardware, which can be configured by the Security IC Embedded Software to calculate the AES algorithm or the inverse AES algorithm on blocks of 128 bits with a selectable key length of 128, 192 or 256 bits. The keys shall be provided by the Security IC Embedded Software.

This security functionality covers:

- FCS_COP.1/AES

7.1.2.4 SS.GCM : GCM coprocessor

SS.GCM serves Security IC Embedded Software with support of Galois/Counter Mode (GCM) of operation for symmetric-key cryptographic block ciphers and Galois Message Authentication Code (GMAC) as defined in [51].

For this purpose SS.GCM implements a GCM coprocessor in hardware, which can be configured by the Security IC Embedded Software to perform Galois field multiplication of two 128-bits input values according to section 6.3 of [51].

This security functionality covers:

- FCS_COP.1/GCM

7.1.2.5 SS.SBC : SBC interface functions

SS.SBC serves the Security IC Embedded Software with support of Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) and Counter (CTR) modes of operation for symmetric block ciphers as defined in *“NIST SP 800-38A”* [47], *“Addendum to NIST SP 800-38A”* [48] and with support of Galois/Counter Mode (GCM) of operation for symmetric block ciphers and Galois Message Authentication Code (GMAC) as defined in *“NIST SP 800-38D”* [51].

For this purpose SS.SBC implements XOR operations in hardware and also implements an increment function in hardware according to section 6.2 of *“NIST SP 800-38D”* [51] with $s = 32$. In addition, the TOE implements a register bank that handles input and output data of SS.TDES , SS.AES , SS.GCM as well as their pre- and post-processing with XOR operations and increment function.

This security functionality covers:

- FCS_COP.1/TDES
- FCS_CKM.4/TDES
- FCS_COP.1/AES
- FCS_CKM.4/AES
- FCS_COP.1/GCM

7.1.2.6 SS.CRC : CRC coprocessor

SS.CRC serves the Security IC Embedded Software with calculation of of cyclic redundancy checks as defined in [64] for 8 bits, in [66] for 16 bits and in [67] for 32 bits.

For this purpose SS.CRC implements two CRC coprocessors in hardware. Each CRC coprocessor can be configured by Security IC Embedded software to calculate a cyclic redundancy check over a data stream of selectable number of one, two, three or four input bytes. The Security IC embedded Software can choose the cyclic redundancy check out of an 8-bits value based on the polynomial in [64], a 16-bits value based on the polynomial in [66] and a 32-bits value based on the polynomial in [67].

This security functionality covers:

- FCS_COP.1/CRC

7.1.2.7 SS.SW_AES : Software AES

.

The TOE uses the AES coprocessor to provide AES encryption and decryption facility using 128, 192 and 256 bit keys.

The TOE implements additional countermeasures that are configurable at runtime and provides functionality for handling checksums over loaded keys.

The supported modes are ECB, CBC, CFB, CTR, GCM, CBC-MAC, CCM, OFB and CMAC (i.e. the CBC mode applied to the block cipher algorithm AES).

The CBC-MAC mode of operation is rather similar to the CBC mode of operation, but returns only the last cipher text (see also *ISO/IEC 9797-1* [53], Algorithm 1).

SS.SW_AES is a basic cryptographic function which provides the AES algorithm as defined by the standard *“FIPS PUB 197-2001”*[58].

The interface to SS.SW_AES allows AES operations independent from prior key loading. The user has to take care that adequate keys of the correct size are loaded before the cryptographic operation is performed. Details are described in the user guidance [15] and the user manual [39].

.

This security functionality covers:

- FCS_COP.1/SW_AES

7.1.2.8 SS.SW_DES : Software DES

.

The TOE uses the Triple-DES hardware coprocessor to provide a Triple-DES encryption and decryption. The Triple-DES function uses double-length or triple-length keys with sizes of 112 or 168 bits respectively.

The TOE implements additional counter measures that are configurable at runtime and provides functionality for handling checksums over loaded keys.

The supported modes are ECB, CBC, CFB, CTR, CBC-MAC, RetailMAC, OFB and CMAC (i.e. the CBC mode applied to the block cipher algorithm TDES).

The CBC-MAC mode of operation is rather similar to the CBC mode of operation, but returns only the last cipher text (see also “ ISO/IEC 9797-1” [53], Algorithm 1, or “ FIPS PUB 81-1980” [55], Appendix F).

To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards).In particular this means that Single-DES shall not be used.

SS.SW_DES is a modular basic cryptographic function which provides the Triple-DES algorithm (with two and three keys) as defined by the standard NIST SP 800-67 [79].

The interface to SS.SW_DES allows performing 2-key and 3-key Triple-DES operations independent from prior key loading. The user has to take care that adequate keys of the correct size are loaded before the cryptographic operation is performed. Details are described in the user manual [39]. All modes of operation (ECB, CBC, CTR, CBC MAC) can be applied to two-key TDES and three-key TDES.

.

This security functionality covers:

- FCS_COP.1/SW_DES

7.1.2.9 SS.RSA : RSA

.

The TOE provides functions that implement the RSA algorithm and the RSA-CRT algorithm for data encryption, decryption, signature and verification. All algorithms are defined in PKCS #1, v2.2 (RSAEP, RSADP, RSAP1, RSAVP1) and FIPS PUB 186-4-2013 [57]

This routine supports various key lengths from 512 bits to 4096 bits. To fend off attackers with high attack potential an adequate key length must be used (references can be found in national and international documents and standards).

The TOE contains modular exponentiation functions, which, together with other functions in the TOE, perform the operations required for RSA encryption or decryption. Two different RSA algorithms are supported by the TOE, namely the "Simple Straight Forward Method" (called RSA "straight forward", the key consists of the pair n and d) and RSA using the "Chinese Remainder Theorem" (RSA CRT, the key consists of the quintuple p, q, dp, dq, qInv).

.

This security functionality covers:

- FCS_COP.1/RSA

7.1.2.10 SS.RSA_Pad : RSA Padding

.

The TOE provides functions that implement the RSA algorithm and the RSA-CRT algorithm for message and signature encoding. This IT security functionality supports the EME-OAEP and EMSA-PSS signature scheme. All algorithms are defined in PKCS#1 v2.2 (EME-OAEP, EMSA-PSS) [69] and PKCS#1 v1.5 [70].

This routine supports various key lengths from 512 bits to 4096 bits.

.

This security functionality covers:

- FCS_COP.1/RSA_PAD

7.1.2.11 SS.RSA_PublicExp : RSA Public Exponentiation

.

The TOE provides functions that implement computation of an RSA plain public key from a private CRT key. All algorithms are defined in PKCS #1, v2.2. and FIPS PUB 186-4-2013 [57] .

This routine supports various key lengths from 512 bits to 4096 bits (CRT).

.

This security functionality covers:

- FCS_COP.1/RSA_PubExp

7.1.2.12 SS.ECDSA : Elliptic Curve Digital Signature Algorithm (ECDSA)

.

The TOE provides functions to perform ECDSA Signature Generation and Signature Verification according to ISO/IEC 14888-3 [61], ANSI X9.62-1998 [68] and FIPS PUB 186-4-2013 [57] .

Note that hashing of the message must be done beforehand and is not provided by this security functionality, but could be provided by SS.SHA .

The supported key length is 128 to 640 bits for both signature generation and for signature verification.

.

This security functionality covers:

- FCS_COP.1/ECDSA

7.1.2.13 SS.ECC_DHKE : ECC Diffie Hellman Key Exchange

.

The TOE provides functions to perform Diffie-Hellman Key Exchange according to ISO/IEC 11770-3-2015 [62] and ANSI X9.63 [80]. It can also be used as secure point multiplication.

The supported key length is 128 to 640bits.

.

This security functionality covers:

- FCS_COP.1/ECC_DHKE

7.1.2.14 SS.ECC_Add : Full ECC Point Addition

.

The TOE provides functions to perform a full ECC point addition according to ISO/IEC 15946-1-2008 [63].

The supported key length is 128 to 640 bits.

.

This security functionality covers:

- FCS_COP.1/ECC_Add

7.1.2.15 SS.ECDAA : Elliptic Curve Direct Anonymous Attestation (ECDAA)

The TOE provides the ECDAA related function to perform an ECDAA signature calculation as specified in the TPM 2.0 [77].

This security functionality covers:

- FCS_COP.1/ECDAA

7.1.2.16 SS.RSA_KeyGen : RSA Key Generator

.

The TOE provides functions to generate RSA key pairs as described in PKCS #1, v2.2, FIPS PUB 186-4-2013 [57]. With this the TOE complies to the content of SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms [78].

It supports various key lengths from 512 bits to 4096 bits.

Two different output formats for the key parameters are supported by the TOE, namely the "Simple Straight Forward Method" (RSA "straight forward") and RSA using the "Chinese Remainder Theorem" (RSA CRT).

.

This security functionality covers:

- FCS_CKM.1/RSA

7.1.2.17 SS.ECC_KeyGen : ECC Key Generator

.

The TOE provides functions to perform ECC over GF(p) Key Generation according to ISO/IEC 15946-1 [63], ANSI X9.62-1998 [68] and FIPS PUB 186-4-2013 [57]. With this the TOE complies to the content of SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms [78].

It supports key length from 128 to 640bits.

.

This security functionality covers:

- FCS_CKM.1/ECC

7.1.2.18 SS.SHA : Secure Hash Algorithms

.

The TOE implements functions to compute the Secure Hash Algorithms SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 according to the standard FIPS PUB 180-4-2011 [56] and SHA-3/224, SHA-3/256, SHA-3/384, SHA-3/512 and SHAKE-128 and SHAKE-256 according to the standard and FIPS PUB 180-4-2011 [56] and FIPS PUB 202-2015 [60].

To fend off attackers with high attack potential an adequate security level must be used (references can be found in national and international documents and standards). In particular this means that SHA-1 shall not be used.

.

This security functionality covers:

- FCS_COP.1/SHA

7.1.2.19 SS.HMAC : HMAC

.

The TOE provides functions to perform HMAC Keyed-hash Message Authentication algorithm according to FIPS 198-1 [59] and FIPS 202 [60].

The TOE supports the calculation of HMAC authentication code with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384 or SHA-3/512 hash algorithms. The HMAC algorithm can use either the high security level or standard security level version of SHA, depending on required security level.

There is no limitation on the supported key length except that it must be a multiple of 8 bits.

To fend off attackers with high attack potential an adequate key length must be used (references can be found in national and international documents and standards). In particular this means that HMAC with SHA-1 shall not be used.

.

This security functionality covers:

- FCS_COP.1/HMAC

7.1.2.20 SS.EDDSA : Edwards-curve Digital Signature Algorithm (EdDSA)

.

This TSF performs EdDSA signature generation and verification for cryptographic key sizes 128 to 640 bits. It meets IETF RFC 8032 [71] as well as IETF RFC 8032 [71] except for the scalar calculation (where the scalar used for the point multiplication is chosen randomly and not computed from the private key).

.

This security functionality covers:

- FCS_COP.1/EDDSA

7.1.2.21 SS.EDDSA_KeyGen : EdDSA Key Generation

.

The TSF generates private/public key pairs for the EdDSA signature scheme for key sizes from 128 to 640 bits that meets the IETF RFC 8032 [71].

.

This security functionality covers:

- FCS_CKM.1/EDDSA

7.1.2.22 SS.MONT_KeyGen : MontDH Key Generation

.

The TSF generates key pairs for Diffie-Hellman Key Exchange in accordance with the Curve25519 and Curve448 and generalizations thereof for a wider class Montgomery curves over GF(p) with key sizes from 128 to 640 bits. It meets IETF RFC 7748 [72].

.

This security functionality covers:

- FCS_CKM.1/MONT

7.1.2.23 SS.MONT_DHKE : MontDH Key Exchange

.

The TSF performs Diffie-Hellman Key Exchange in accordance with Curve25519 and Curve448 and generalizations thereof for a wider class Montgomery curves over GF(p) with key sizes from 128 to 640 bits. It meets IETF RFC 7748 [72].

.

This security functionality covers:

- FCS_COP.1/MONT_DHKE

7.1.2.24 SS.KDF : KDF: Cryptographic Key Derivation

.

The TOE provides a function to perform hash-based key derivation according to ANSI X9.63 [80]. The TOE supports the calculation of KDF with SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3/224, SHA-3/256, SHA-3/384 or SHA-3/512 hash algorithms. The KDF algorithm can use either the high security level or standard security level version of SHA, depending on required security level. To fend off attackers with high attack potential an adequate key length must be used (references can be found in national and international documents and standards). In particular this means that HMAC with SHA-1 shall not be used.

.

This security functionality covers:

- FCS_CKM.5/KDF

7.1.2.25 SS.EUICC : eUICC Authentication

.

The TSF provides services for authentication for eUICC applications. It implements the cryptographic algorithms MILENAGE, TUAK and CAVE.

.

This security functionality covers:

- FCS_COP.1/EUICC

7.1.2.26 SS.COPY : Secure Copy

.

The security service SS.COPY implements functionality to copy memory content in a secure manner protected against attacks.

.

This security functionality covers:

- FDP_SOP.1/Copy

7.1.2.27 SS.COMPARE : Secure Compare

.

The security service SS.COMPARE implements functionality to compare different blocks of memory content in a manner protected against attacks.

.

This security functionality covers:

- FDP_SOP.1/Compare

7.1.2.28 SS.ARITH_OP : Secure arithmetic operations

.

The security service SS.ARITH_OP provides functions to do arithmetic operations (modular addition, modular subtraction, modular multiplication, modular inversion, arithmetic comparison and exact addition) in a side channel resistant way.

This resistance against attacks is described in Section 7.2.2.

.

This security functionality covers:

- FDP_SOP.1/Arith_op

7.1.2.29 SS.SW_CRC : Software CRC

.

SS.SW_CRC serves the Security IC Embedded Software with calculation of of cyclic redundancy checks as defined in [66] for 16 bits and in [67] for 32 bits.

.

This security functionality covers:

- FCS_COP.1/SW_CRC

7.1.2.30 SS.SW_RNG : Software RNG

.

SS.SW_RNG consists of a software (pseudo) RNG that can be used as a general purpose random source, and of appropriate online tests for the hardware RNG (SS.RNG):

- The Crypto Library implements a software RNG that complies to class PTG.3 for Hybrid-physical RNG and class DRG.4 for Hybrid-deterministic RNG according to [6]. This software RNG is seeded by random numbers taken from the hardware RNG. The implementation of the software RNG is based on the standard NIST SP 800-90A as described in [52].
- The Crypto Library implements appropriate online tests according to the Hardware User Guidance Manual [10] for the hardware RNG, to complete PTG.2 as required by FCS_RNG.1/PTG.2. The interface of SS.SW_RNG allows to test the hardware RNG and to seed the software RNG after successful testing.

.

This security functionality covers:

- FCS_RNG.1/HYB-DET
- FCS_RNG.1/HYB-PHY

7.1.3 Security Features of the TOE

7.1.3.1 SF.OPC : Control of Operating Conditions

SF.OPC controls operating conditions of the TOE. These are explicitly controlled by security functionality that simply hampers feeding certain electrical stimulations into the device. Such security functionality is composed of frequency filters and voltage limiters. Operating conditions of the device are explicitly controlled also by security functionality that actively monitors certain electrical parameters. These parameters are voltage levels of external supply from pad and internal supplies, frequencies of internal clocks and on-chip temperature. Such security functionality raises an error message whenever a monitored parameter drops out of its valid range. In addition, exposure of the device to light is explicitly controlled by security functionality that senses abnormal light over its whole surface, raising an error message when detected.

SF.OPC also controls operating conditions implicitly. This is done by security functionality that detects faults in code and data stored to memories and while processed in the device. Such faults might be inserted by electrical stimulations or by exposure of the device to energy or particles. Error detection codes are used to protect the memories as well as the access channels over the bus system to memories and to hardware peripherals on the control bus, the direct access channel to PKC RAM and security-relevant storage and processing in CPU coprocessor and hardware peripherals on the control bus including SBC interface with Triple-DES, AES, GCM coprocessors and CRC coprocessor. Watchdogs on error detection codes run over code and data stored to System RAM and PKC RAM, and the Secure Fetch Plus on code and data read from Flash memory and Secure Fetch on code and data read from ROM memory can be configured and enabled by Security IC Embedded Software.

Further on, Security IC Embedded Software can configure and enable a Secure Fetch on CPU code and/or data accesses over the bus system and also range checks on values in general purpose, stack pointers and link registers of the CPU as well as checks on predefined CPU instructions for zero values in their operands or in the addresses of their resulting data accesses to memory. In addition, Security IC Embedded Software may protect its program flow by use of a signature watchdog on CPU code accesses over the bus system, by use of a secure counter and by use of a watchdog timer.

SF.OPC also provides the Security IC Embedded Software with multiple calculation modes for the Triple-DES, AES and GCM coprocessors. Triple-DES and AES coprocessors each is equipped with a fault detection mechanism on its key schedule that must be enabled by Security IC Embedded Software.

In case an error message is raised the TOE either (i) aborts code execution and forces a reset or (ii) raises an exception, which interrupts code execution and jumps to an exception vector on which the Security IC Embedded Software can react with an appropriate exception handler. In case of reset the TOE returns to its initial state and provides information on the reset source to the Security IC Embedded Software. In case of an exception the TOE provides information on the exception source to the Security IC Embedded Software.

SF.OPC also implements security functionality that corrects errors in Flash memory.

This security functionality covers:

- FRU_FLT.2
- FPT_FLS.1
- FPT_PHP.3
- FDP_SDI.2/FLT

7.1.3.2 SF.PHY : Protection against Physical Manipulation

SF.PHY protects the TOE from physical probing and physical manipulation of its hardware, its IC Dedicated Software, its TSF data and Security IC Embedded Software stored to its Flash memory including user data of the Composite TOE. This is achieved by appropriate shielding techniques for all elements in the physical design of the TOE, as well as redundant routing of sensitive signals and layout constraints on particular placements and routings.

Selected security functionality in analog design parts of the TOE is additionally checked for its basic operability by a built-in selftests that run during startup of the device.

Memories and their interfaces are additionally protected against probing by appropriate encryption of stored content and address scrambling mechanisms.

This security functionality covers:

- FRU_FLT.2
- FPT_FLS.1
- FPT_PHP.3
- FDP_SDC.1
- FDP_ITT.1
- FPT_ITT.1
- FDP_IFC.1

7.1.3.3 SF.LOG : Logical Protection

SF.LOG provides logical protection of the TOE that fights disclosure of confidential data stored to and processed in the TOE through tracing of power consumption or emanation and subsequent complex signal analysis.

Triple-DES, AES, GCM and CRC coprocessors each implements security functionality that eliminates exploitable side channel leakage. Such security functionality in Triple-DES and AES coprocessors uses masking techniques in data processing, inserts diverse dummy activity that can partly be configured by Security Embedded Software, and randomizations. GCM coprocessor and CRC coprocessor implement masking schemes on their data processing.

Input and output data of Triple-DES, AES and GCM coprocessors are handled via the register bank in the SBC interface that implements masking. XOR operations in the SBC interface are embedded in this masking.

The PKC coprocessor implements security functionality that effectively reduces side channel leakage by adding noise, inserting dummy activity and randomizations.

Code and data are masked on their transfer via the access channels over the bus system to memories and hardware peripherals on the control bus like SBC interface, CRC coprocessor and PKC coprocessor. The CPU embeds masking schemes for storage and processing of data and code.

SF.LOG also serves the Security IC Embedded Software with security functionality for additional protection for loading of data into the register bank of the SBC interface and into the input register of the CRC coprocessor.

This security functionality covers:

- FDP_ITT.1
- FPT_ITT.1
- FDP_IFC.1

7.1.3.4 SF.FOS-USE : Factory OS use restrictions

SF.FOS-USE restricts use of the Factory OS among three levels of testing capabilities of the TOE. Access to the lower level of testing capabilities is not blocked. Instead, its testing capabilities are very limited so that they cannot be exploited. The medium level of testing capabilities is blocked by an authentication procedure. After successful authentication to this level the TOE serves with testing capabilities to the extent that confidentiality of content stored to its memories cannot be compromised.

The upper level of testing capabilities is blocked by two authentication checks, of which the latter one also forces an erase of AP-Flash, BL-Flash, SH-Flash and SV-Flash windows as well as System Page Application, System Page Bootloader and System Page Common before full testing capabilities are provided.

Commands of the Factory OS are conditionally installed in stages and commands with test functionality are cut to tests of basic functionality only.

This security functionality covers:

- FMT_LIM.1
- FMT_LIM.2
- FAU_SAS.1

7.1.3.5 SF.MEM-ACC : Memory Access Control

SF.MEM-ACC controls access to the memories of the TOE. This is done based on physical restrictions in the bus system that block certain access ports for particular memories, and also direct memory access to PKC RAM is physically restricted to the PKC coprocessor.

In addition, security functionality is implemented that checks every single access over the bus system to the memories against predefined and/or configurable access rights for each of the following combinations of system operations modes and CPU privilege levels:

- NXP Mode
- Service Mode privileged
- Service Mode unprivileged
- Shared Mode
- Bootloader Mode privileged
- Bootloader Mode unprivileged
- Application Mode privileged
- Application Mode unprivileged

Every access over the bus system to a memory address is checked against access rights in read, write and execute. Access rights are set for predefined default address windows in ROM, Flash memory, System RAM and PKC RAM and also for configurable software-controlled address windows within these default address windows. Configurations are accessible to Security IC Embedded Software.

System operation modes and CPU privilege levels are assigned to each access port on the bus system and are transmitted over the bus system into the memory controllers. System operation modes and CPU privilege levels are also transmitted into the mode controller, which implements appropriate rules for transformations in system operation modes that dynamically update those assigned to CPU and DMA controller access ports, whereat Service Mode is permanently masked out for the DMA controller access port. CPU privilege levels are updated by the CPU. The PKC controller access port is assigned with system operation modes that are dynamically updated to the access rights actually valid for direct memory access to PKC RAM.

This security functionality covers:

- FMT_LIM.2
- FDP_ACC.1/MEM
- FDP_ACF.1/MEM
- FMT_MSA.1/MEM
- FMT_MSA.3/MEM
- FMT_SMF.1

7.1.3.6 SF.SFR-ACC : Special Function Register Access Control

SF.SFR-ACC controls access to the Special Function Registers of the TOE. This is done based on physical restrictions in the bus system that block DMA controller access to hardware components on the control bus and also PKC coprocessor access to hardware components on both, control bus and peripheral control bus.

In addition, security functionality is implemented that checks every single access over the bus system on the control bus and on the peripheral control bus to a Special

Function Register against predefined and/or configurable access rights for the following combinations of system operation modes and CPU privilege levels:

- NXP Mode
- Service Mode privileged
- Service Mode unprivileged
- Bootloader Mode privileged or Application Mode privileged
- Bootloader Mode unprivileged or Application Mode unprivileged

Control of access to the Special Function Registers is done in two layers of security functionality. The first layer of security functionality is implemented in every hardware component that is connected to the control bus or to the peripheral control bus and checks each access to a Special Function Register per bit against predefined and partly configurable access rights in read and write. Such access rights cannot be enlarged in the second layer but only be further restricted. The second layer of security functionality is implemented in the bus system and can be configured to either completely block access to the group of Special Function Registers belonging to a hardware component or completely unblock it to the extent provided in the first layer. Configurations of access rights are accessible to Security IC Embedded Software.

Relevant combinations of system operation modes and CPU privilege levels are assigned to every access port on the bus system and are transmitted over the bus system into the hardware components on the control bus and on the peripheral control bus. System operation modes and CPU privilege levels are also transmitted into the mode controller, which implements appropriate rules for transformations in system operation modes that dynamically update those assigned to CPU and DMA controller access ports, whereat Service Mode is permanently masked out for the DMA controller access port. CPU privilege levels are updated by the CPU.

This security functionality covers:

- FMT_LIM.2
- FDP_ACC.1/SFR
- FDP_ACF.1/SFR
- FMT_MSA.1/SFR
- FMT_MSA.3/SFR
- FMT_SMF.1

7.1.3.7 SF.FLASH-SVC : Flash Services

SF.FLASH-SVC provides a Flash Services Software application programming interface (API), which serves Security IC Embedded Software with operations that update content in Flash memory (Flash erase and/or programming). These operations are tearing-save and verify the updated content in Flash memory.

SF.FLASH-SVC implements dynamic wear-leveling for Flash memory and serves a Flash Services Software application progamming interface (API) that provides Security IC Embedded Software with functionality for static wear-leveling and Flash memory refreshing, and with wearout indication for Flash memory.

In addition, the Flash Services Software application programming interface (API) provides Security IC Embedded Software with write access control to certain System Pages depending on the System Operation Mode.

This security functionality covers:

- FRU_FLT.2
- FPT_FLS.1
- FDP_SDI.2/AGE
- FDP_ACC.1/MEM
- FDP_ACF.1/MEM

7.1.3.8 SF.Object_Reuse

.

The TOE provides internal security measures which clear memory areas used by the Crypto Library after usage. This functionality is required by the security functional component FDP_RIP.1 and FCS_CKM.4, taken from the Common Criteria Part 2 [2].

These measures ensure that a subsequent process may not gain access to cryptographic assets stored temporarily in memory used by the TOE.

This security functionality covers:

- FDP_RIP.1
- FCS_CKM.4

7.2 TOE Summary Specification Rationale

7.2.1 Rationale for the Portions of the TOE Security Functionality

Deleted here, only available in the full version of the Security Target.

7.2.2 Security Architectural Information

Deleted here, only available in the full version of the Security Target.

8 Bibliography

8.1 Evaluation documents

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

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

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

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

[5] Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0084-2014

[6] A proposal for: Functionality classes for random number generators, Version 2.0, 18 September 2011, Bundesamt für Sicherheit in der Informationstechnik/T-Systems GEI GmbH

[7] Evaluation of random number generators, Bundesamt für Sicherheit in der Informationstechnik, Version 0.10

[8] AIS20: Anwendungshinweise und Interpretationen zum Schema (AIS), Funktionalitätsklassen und Evaluationsmethodologie für deterministische Zufallszahlengeneratoren, Version 3, 15.05.2013, Bundesamt für Sicherheit in der Informationstechnik (BSI),

[9] JIL-ATT-SC: Attack Methods for Smartcards and Similar Devices, Joint Interpretation Library, Version 2.2, January 2013

8.2 Developer documents

[10] SN220_SE Information on Guidance and Operation, Version 1.0, 12.07.2021, NXP Semiconductors

[11] SN220 Services User Manual - API and Operational Guidance, Version 1.0, 01.10.2020, NXP Semiconductors

[12] SN220 Services User Manual - API and Operational Guidance, Version 1.1, 05.05.2022, NXP Semiconductors

[13] SN220 Services Addendum - Additional API and Operational Guidance, Version 1.0, 01.10.2020, NXP Semiconductors

[14] SN220 Services Addendum - Additional API and Operational Guidance, Version 1.1, 05.05.2022, NXP Semiconductors

[15] SN220x Crypto Library Information on Guidance and Operation, Version 1.1, 12.07.2021, NXP Semiconductors

[16] SN220x Crypto Library Information on Guidance and Operation, Version 1.3, 31.08.2022, NXP Semiconductors

[17] SN220x_SE High-performance secure element subsystem, Product data sheet, DocID 633012, Rev. 1.2, 12.07.2022, NXP Semiconductors

[18] SN220x_SE - SFR Tables for Coburg core, Version 0.1, 04.09.2020, NXP Semiconductors

[19] SN220x Wafer and Delivery Specification, Product data sheet addendum, DocID 632913, Rev. 1.3, 24 August 2022, NXP Semiconductors

[20] P73 family SC300 User Manual, Product Data sheet addendum, DocID: 341410, NXP Semiconductors

[21] ARM®v7-M Architecture Reference Manual, DDI 0403E.b (ID120114), ARM

[22] P73 family DMA Controller PL080 User manual, Product data sheet addendum, DocID: 341510, NXP Semiconductors

[23] P73 Family Code Signature Watchdog, Application note, Version 1.1, 12.11.2018, NXP Semiconductors

[24] SN220x Crypto Library: User Manual – RNG Library, Version 1.0, 10.06.2020

[25] SN220x Crypto Library: User Manual – RNG Library, Version 1.1, 18.12.2020

[26] SN220x Crypto Library: User Manual – HASH Library, Version 1.0, 19.06.2020

[27] SN220x Crypto Library: User Manual – SHA Library, Version 1.0, 19.06.2020

[28] SN220x Crypto Library: User Manual – Secure SHA Library, Version 1.0, 19.06.2020

[29] SN220x Crypto Library: User Manual – SHA-3 Library, Version 1.0, 10.08.2020

[30] SN220x Crypto Library: User Manual – Secure SHA-3 Library, Version 1.0, 10.08.2020

[31] SN220x Crypto Library: User Manual – HMAC Library, Version 1.0, 19.06.2020

[32] SN220x Crypto Library: User Manual – RSA Library, Version 1.0, 21.07.2020

[33] SN220x Crypto Library: User Manual – RSA Library, Version 1.1, 15.12.2020

[34] SN220x Crypto Library: User Manual – RSA Key Generation Library (RsaKg), Version 1.0, 30.07.2020

[35] SN220x Crypto Library: User Manual – ECC over GF(p) Library, Version 1.0, 17.06.2020

[36] SN220x Crypto Library: User Manual – ECDAA, Version 1.0, 17.06.2020

[37] SN220x Crypto Library: User Manual – TwdEdMontGfp Library, Version 1.0, 23.06.2020

[38] SN220x Crypto Library: User Manual – eUICC Library, Version 1.0, 17.06.2020

[39] SN220x Crypto Library: User Manual – Symmetric Cipher Library (SymCfg), Version 1.0, 23.06.2020

[40] SN220x Crypto Library: User Manual – Symmetric Cipher Library (SymCfg), Version 1.1, 25.02.2022

[41] SN220x Crypto Library: User Manual – Utils Library, Version 1.0, 23.06.2020

[42] SN220x Crypto Library: User Manual – Utils Math Library, Version 1.0, 23.06.2020

[43] SN220x Crypto Library: User Manual – Kdf Library, Version 1.0, 19.06.2020

[44] Order Entry Form, online document, NXP Semiconductors

[45] Trust Provisioning for Secure ICs - Trust Provisioning System Architecture, Version 1.08, 07.03.2017, NXP Semiconductors

[46] P73 Family Chip Health Mode, Application note, DocID: 411910, Rev. 1.0, 09.08.2018, NXP Semiconductors

8.3 Standards

[47] NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation, National Nationale Institute of Standards and Technology, Edition 2001

[48] Addendum to NIST SP 800-38A: Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode, National Nationale Institute of Standards and Technology, October 2010

[49] NIST SP 800-38B: Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, May 2005, Morris Dworkin, National Institute of Standards and Technology

[50] NIST SP 800-38C: Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, July 2007, Morris Dworkin, National Institute of Standards and Technology

[51] NIST SP 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/ Counter Mode (GCM) and GMAC, November 2007, Morris Dworkin, National Institute of Standards and Technology

[52] NIST SP 800-90A, Revision 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015, Elaine Barker and John Kelsey, National Institute of Standards and Technology

[53] ISO/IEC 9797-1: 2011 Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher

[54] ISO 11568-4-2007: Banking – Key management (retail) – Part 4: Asymmetric cryptosystems – Key management and life cycle, 2007

[55] FIPS PUB 81-1980: DES modes of operation, Federal Information Processing Standards Publication, December 2nd, 1980, US Department of Commerce/ National Institute of Standards and Technology

[56] FIPS PUB 180-4-2011: Secure Hash Standard, Federal Information Processing Standards Publication, February 2011, US Department of Commerce/National Institute of Standards and Technology

[57] FIPS PUB 186-4-2013: Digital Signature Standard, Federal Information Processing Standards Publication, 2013, July, National Institute of Standards and Technology

[58] FIPS PUB 197-2001: ADVANCED ENCRYPTION STANDARD (AES), Federal Information Processing Standards Publication 197, November 26, 2001, U.S. Department of Commerce/National Institute of Standards and Technology

[59] FIPS PUB 198-1-2008: The Keyed-Hash Message Authentication Code (HMAC), Federal Information Processing Standards Publication, July 2008, US Department of Commerce/National Institute of Standards and Technology

[60] FIPS PUB 202-2015: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, Federal Information Processing Standards Publication, August 2015, US Department of Commerce/National Institute of Standards and Technology

[61] ISO/IEC 14888-3-2015: Information technology – Security techniques – Digital signatures with appendix -- Part 3: Discrete logarithm based mechanisms, 2016

[62] ISO/IEC 11770-3-2015: Information technology – Security techniques – Key management -- Part 3: Mechanisms using asymmetric techniques, 2015

[63] ISO/IEC 15946-1-2008: Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General, 2008

[64] "SERIES I: INTEGRATED SERVICES DIGITAL NETWORK, ISDN user-network interfaces – Layer 1 Recommendations", International Telecommunication Union, ITU-T Recommendation I.432.1, Februar 1999

[65] "SERIES V: DATA COMMUNICATION OVER THE TELEPHONE NETWORK, Error control", International Telecommunication Union, ITU-T Recommendation V.42, March 2002

[66] "SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATION, Public data networks – Interfaces", International Telecommunication Union, ITU-T Recommendation X.25, October 1996

[67] "IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/ CD) access method and physical layer specifications", IEEE Computer Society, IEEE Std 802.3™-2005, Dec-12, 2005

[68] ANSI X9.62-1998: Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), September 1998, American National Standards Institute

[69] PKCS#1 v2.2: RSA Cryptography Standard, October 2012, RSA Laboratories

[70] PKCS#1 v1.5: RSA Encryption, March 1998, RSA Laboratories

[71] IETF RFC 8032: Edwards-Curve Digital Signature Algorithm (EdDSA), January 2017, Internet Research Task Force (IRTF)

[72] IETF RFC 7748: Elliptic Curves for Security, January 2016, Internet Research Task Force (IRTF)

[73] 3GPP TS 35.205: 3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: General, v14.0.0, March 2017

[74] 3GPP TS 35.206: 3G Security; Specification of the MILENAGE algorithm set: An example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 2: Algorithm specification, v14.0.0, March 2017

[75] 3GPP TS 35.231: 3G Security; Specification of the TUAK algorithm set: A second example algorithm set for the 3GPP authentication and key generation functions f1, f1*, f2, f3, f4, f5 and f5*; Document 1: Algorithm specification, v 13.0.0, January 2016

[76] 3GPP2 S.S0053-0: Common Cryptographic Algorithms, v2.0, May 2009

[77] TPM Rev. 2.0: Trusted Platform Module Library Specification, Family "2.0", Level 00, Revision 01.38 – September 2016 and ISO/IEC 11889:2015, Parts 1-4

[78] SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, SOG-IS Crypto Working Group, Version 1.0, May 2016

[79] NIST SP 800-67, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher, National Nationale Institute of Standards and Technology, Revised January 2012

[80] ANSI X9.63: Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography, December 2011, American National Standards Institute

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.

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.

9.3 Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

Tables

Figures

Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

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

Date of release: 29 September 2022