

# TEGRION™ SLC21 Post-Quantum Edition

## Security Target Lite

### Release

### About this document

#### Scope and purpose

This document is the Security Target for the Infineon TEGRION™ SLC21 Post-Quantum Edition security controllers.

#### Intended audience

Composite product developers, Common Criteria Evaluators and Certifiers.

## Table of contents

<b>Table of contents</b> .....	<b>2</b>
<b>List of figures</b> .....	<b>3</b>
<b>List of tables</b> .....	<b>4</b>
<b>1 Introduction (ASE_INT)</b> .....	<b>6</b>
1.1 ST reference.....	6
1.2 TOE reference .....	6
1.3 TOE overview.....	6
1.3.1 TOE definition and usage.....	6
1.3.2 TOE major security features .....	6
1.4 TOE description .....	7
1.4.1 TOE components.....	7
1.4.1.1 TOE hardware.....	7
1.4.1.2 Firmware .....	9
1.4.1.3 Libraries.....	9
1.4.2 Physical scope .....	9
1.4.3 Logical scope.....	10
1.4.3.1 TSF.....	10
1.4.4 TOE delivery .....	11
1.4.5 Production sites .....	12
1.4.6 Configurations.....	12
1.4.7 Initialisation with embedded software .....	12
<b>2 Conformance (ASE_CCL)</b> .....	<b>13</b>
2.1 Conformance claims .....	13
2.1.1 PP claims .....	13
2.1.2 Package claims.....	13
2.2 Conformance rationale .....	13
<b>3 Security Problem Definition (ASE_SPD)</b> .....	<b>14</b>
3.1 Threats.....	14
3.1.1 Threats from PP0084.....	14
3.2 Organizational security policies .....	14
3.2.1 Organizational security policies from PP0084 .....	14
3.2.2 Organizational security policies defined in this ST .....	14
3.3 Assumptions .....	15
3.3.1 Assumptions defined in [PP0084].....	15
3.3.2 Assumptions defined in this ST .....	15
<b>4 Security Objectives (ASE_OBJ)</b> .....	<b>16</b>
4.1 Security objectives for the TOE.....	16
4.1.1 Security objectives for the TOE defined in PP0084.....	16
4.1.2 Security objectives for the TOE defined in this ST .....	16
4.2 Security objectives for the operational environment (OE).....	17
4.2.1 OEs defined in [PP0084] .....	17
4.2.2 OEs defined in this ST .....	17
4.3 Security objectives rationale .....	17
<b>5 Extended Components Definition (ASE_ECD)</b> .....	<b>18</b>
5.1 Extended components defined in [PP0084].....	18
5.2 Extended components defined in this ST .....	18

## List of figures

<b>6</b>	<b>Security Requirements (ASE_REQ)</b> .....	<b>19</b>
6.1	Security functional requirements.....	19
6.1.1	Hardware random number generators.....	19
6.1.2	Cryptographic services implemented in hardware .....	20
6.1.3	TSF testing.....	20
6.1.4	Malfunctions.....	21
6.1.5	Abuse of Functionality .....	21
6.1.6	Physical Manipulation and Probing.....	22
6.1.7	Leakage.....	23
6.1.8	Application Firewall .....	23
6.1.8.1	Policy definition .....	23
6.1.8.2	SFRs .....	25
6.1.9	Authentication of the Security IC.....	28
6.1.10	Flash loader .....	28
6.1.10.1	SFRs added in this ST .....	30
6.1.11	PQ Crypto Suite .....	31
6.1.11.1	Hash.....	32
6.1.11.2	MLKEM .....	32
6.1.11.3	Random .....	33
6.2	Security assurance requirements.....	37
6.2.1	Security Policy Model (SPM) details .....	38
6.3	Security requirements rationale.....	39
6.3.1	Rationale for the Security Functional Requirements .....	39
6.3.1.1	Additional SFRs related to O.Firewall .....	39
6.3.1.2	Additional SFRs related to O.Ctrl_Auth_Loader .....	40
6.3.1.3	Additional SFRs related to O.Phys-Manipulation .....	40
6.3.1.4	Additional SFRs related to O.Hash .....	40
6.3.1.5	Additional SFRs related to O.MLKEM.....	40
6.3.1.6	Additional SFRs related to O.RND .....	40
6.3.2	Dependencies of Security Functional Requirements .....	40
6.3.3	Rationale of the Assurance Requirements.....	41
<b>7</b>	<b>TOE Summary Specification (ASE_TSS)</b> .....	<b>43</b>
7.1	SF_DPM: Device Phase Management .....	43
7.2	SF_PS: Protection against Snooping.....	44
7.3	SF_PMA: Protection against Modifying Attacks .....	44
7.4	SF_PLA: Protection against Logical Attacks.....	45
7.5	SF_HC: Hardware provided cryptography .....	45
7.6	SF_CS: PQ Crypto Suite Services .....	46
<b>8</b>	<b>Hash values of libraries</b> .....	<b>47</b>
<b>9</b>	<b>Cryptographic Table</b> .....	<b>48</b>
<b>Acronyms</b> .....		<b>49</b>
<b>References</b> .....		<b>49</b>
<b>Revision history</b> .....		<b>51</b>

## List of figures

Figure 1	TOE hardware .....	7
----------	--------------------	---

## List of tables

## List of tables

Table 1	Hardware/Firmware components .....	9
Table 2	Libraries .....	9
Table 3	User guidance .....	9
Table 4	Forms of delivery .....	11
Table 5	TOE configuration options .....	12
Table 6	Order Options to initialize the TOE with customer software .....	12
Table 7	Threats from [PP0084] .....	14
Table 8	Organisational Security Policies from [PP0084] .....	14
Table 9	Organisational Security Policies defined in this ST .....	14
Table 10	Assumptions from [PP0084] .....	15
Table 11	Security objectives for the TOE from [PP0084] .....	16
Table 12	Security Objectives for the TOE .....	16
Table 13	Security objectives for the operational environment from [PP0084] .....	17
Table 14	Additional Security objectives for the environment .....	17
Table 15	FCS_RNG.1/TRNG .....	19
Table 16	FCS_COP.1/AES .....	20
Table 17	FCS_CKM.4 .....	20
Table 18	TSF testing .....	21
Table 19	FAU_SAS.1 .....	21
Table 20	FDP_SDC.1 .....	22
Table 21	FDP_SDI.2 .....	22
Table 22	FDP_ACC.2/AF .....	25
Table 23	FDP_ACF.1/AF .....	26
Table 24	FMT_MSA.3/AF .....	26
Table 25	FMT_MSA.1/AF/S .....	27
Table 26	FMT_MSA.1/AF/NS .....	27
Table 27	FMT_SMF.1/AF .....	27
Table 28	FMT_SMR.1/AF .....	28
Table 29	FIA_API.1 .....	28
Table 30	FMT_LIM.1/Loader .....	28
Table 31	FMT_LIM.2/Loader .....	29
Table 32	FTP_ITC.1 .....	29
Table 33	FDP_ACC.1/Loader .....	29
Table 34	FDP_ACF.1/Loader .....	30
Table 35	FMT_MTD.1/Loader .....	30
Table 36	FMT_SMR.1/Loader .....	31
Table 37	FMT_SMF.1/Loader .....	31
Table 38	FIA_UID.2/Loader .....	31
Table 39	FCS_COP.1/CS/Hash/<iter> .....	32
Table 40	Cryptographic table for FCS_COP.1/CS/Hash/<iter> .....	32
Table 41	FCS_COP.1/CS/MLKEM/<iter> .....	32
Table 42	Cryptographic table for FCS_COP.1/CS/MLKEM/<iter> .....	33
Table 43	FCS_CKM.1/CS/MLKEM/<iter> .....	33
Table 44	Cryptographic table for FCS_CKM.1/CS/MLKEM/<iter> .....	33
Table 45	FCS_RNG.1/CS/PTG2 .....	34
Table 46	FCS_RNG.1/CS/PTG3 .....	34



**List of tables**

Table 47	FCS_RNG.1/CS/DRG3.....	35
Table 48	FCS_RNG.1/CS/DRG4.....	36
Table 49	SAR list and refinements .....	37
Table 50	SFRs excluded from SPM.....	38
Table 51	Rationale for SFRs related to O.Firewall.....	39
Table 52	Rationale for additional SFRs related to O.Ctrl_Auth_Loader .....	40
Table 53	Dependencies of SFRs .....	40
Table 54	TOE Security Features .....	43
Table 55	SHA256 hash values .....	47
Table 56	Cryptographic table .....	48

## 1 Introduction (ASE\_INT)

### 1.1 ST reference

The ST has the title TEGRION™ SLC21 Post-Quantum Edition Security Target Lite, Rev. 1.0.2 and is dated 2024-10-14.

### 1.2 TOE reference

The full TOE name is:

IFX\_CCI\_000068h, IFX\_CCI\_000080h design step G12 with firmware version 80.505.04.1,  
optional PQ Crypto Suite version 5.00.012,  
optional HSL version 04.05.0040,  
optional UMSLC version 02.01.0040,  
and user guidance documents

The TOE is identified by the components as described in the physical scope, chapter 1.4.2.

A customer shall identify the TOE.

- The Hardware version, design step and Firmware version can be read out from the chip by the Generic Chip Identification Mode (GCIM). The procedure how to read that data is described in the Programmers Reference Manual.
- The correct library versions can be verified by the corresponding hash values as defined in chapter 8.
- The correct user guidance documents can be verified by chapter 1.4.2.3.

### 1.3 TOE overview

#### 1.3.1 TOE definition and usage

The TOE consists of a smart card IC (Security Controller), firmware and user guidance meeting high requirements in terms of performance and security. The TOE is designed by Infineon Technologies AG and is intended to be used in smart cards for security-relevant applications and as developing platform for smart card operating systems according to the life cycle model from [PP0084]. The TOE is the platform for the Embedded Software but the Embedded Software itself is not part of the TOE. The TOE does not require any non-TOE hardware/software/firmware.

#### 1.3.2 TOE major security features

- Dual CPU in lockstep mode to detect integrity errors during processing
- Memory integrity protection
- Memory encryption
- Bus masking for security peripherals
- Hardware True RNG
- Symmetric coprocessor for AES encryption and decryption
- Coprocessor to accelerate long integer and modular arithmetic operations for asymmetric cryptography. It relies on the embedded software to implement securely cryptographic algorithms.
- Global alarm system with security life control
- Tearing safe NVM write
- Armv8-M compliant MPU and SAU
- Robust set of sensors and detectors

## Introduction (ASE\_INT)

- Redundant alarm propagation and system deactivation principle
- Peripheral access control
- Leakage control of data dependent code execution
- Device phase management
- The optional PQ Crypto Suite provides hardened cryptographic functions for MLKEM operations.
- The MISE provides masked and side-channel hardened arithmetical and logical CPU instructions.
- Instruction Stream Signature (ISS) coprocessor. The ISS can optionally be used to protect the CPU instruction flow. The hardware-based integrity protection concept of the TOE already provides a very effective program flow protection, such that the ISS is not needed. The ISS can nevertheless be used for compatibility reasons or as a very conservative additional countermeasure.

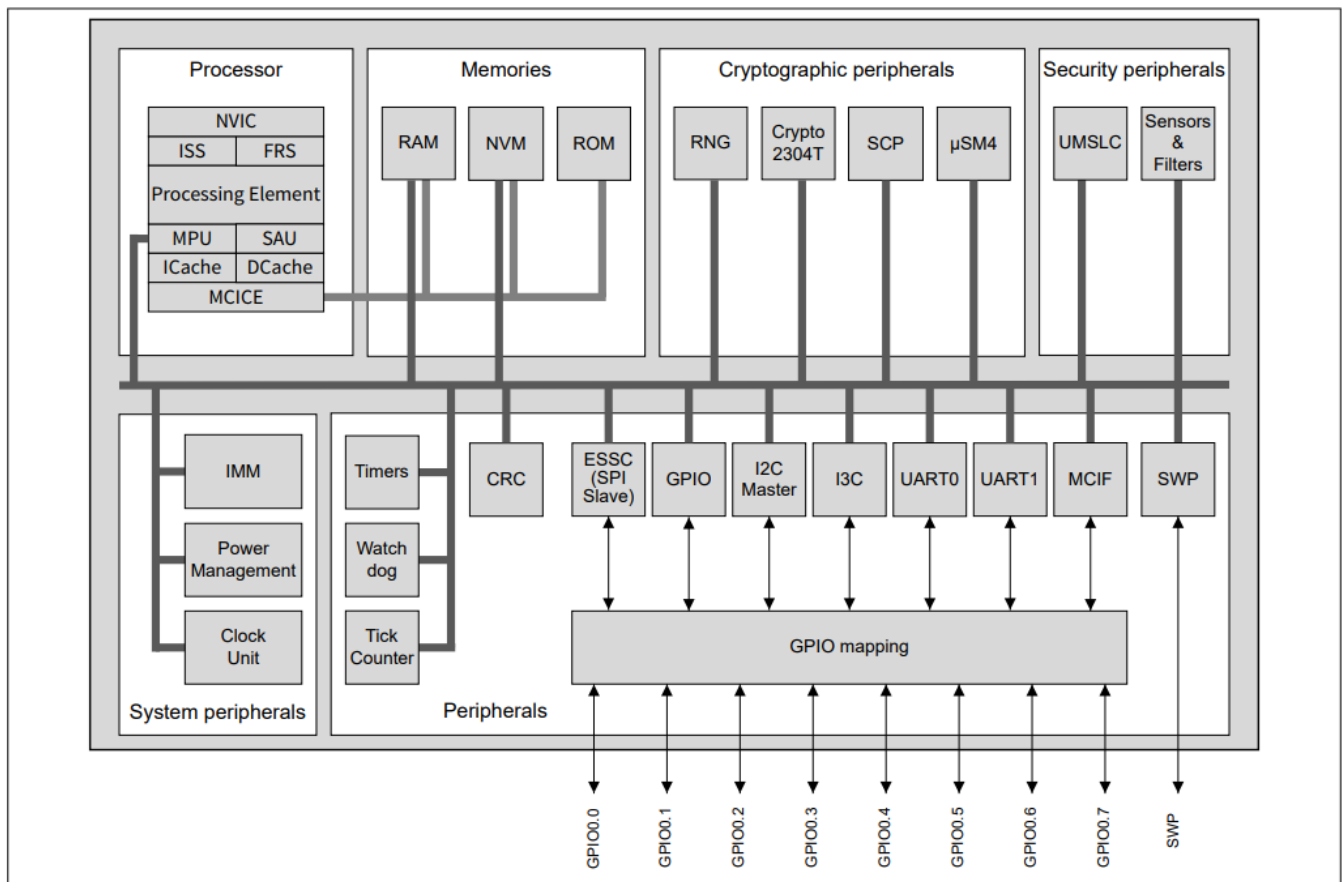
## 1.4 TOE description

### 1.4.1 TOE components

#### 1.4.1.1 TOE hardware

Figure 1 shows schematically the TOE hardware.

**Figure 1 TOE hardware**



The TOE hardware consists of the following blocks:

- Processor
  - CPU according to Armv8-M mainline architecture.

**Introduction (ASE\_INT)**

- Armv8-M compatible NVIC controller
- Armv8-M compatible Memory Protection Unit (MPU) with 8 regions
- Armv8-M compatible Security Attribution Unit (SAU) with 8 regions
- Instruction Stream Signature (ISS) coprocessor
- Fast Random Source (FRS) nonce generator coprocessor
- EDC protected caches for memory access and instruction fetch
- MCICE provides encryption and EDC protection for RAM, ROM and NVM
- Memories
  - encrypted and EDC protected ROM
  - encrypted and EDC protected RAM
  - encrypted and EDC protected NVM
- Peripherals
  - Timers
  - Watchdog
  - Tick counter
  - CRC accelerator
- System peripherals
  - Clock unit
  - Interface Management Module (IMM)
  - Power Management
  - System Peripheral Access Unit (SPAU) to manage access to peripherals.
- Cryptographic peripherals
  - RNG according to class PTG.2 of [AIS 31]
  - Crypto2304T coprocessor for long modular integer arithmetic
  - SCP for secure AES computation
  - $\mu$ SM4 accelerator for SM4 cryptographic algorithm.
- Security peripherals
  - UMSLC
  - Sensors
- I/O Interfaces
  - UART for ISO 7816-3
  - I3C slave which can also be used as I2C slave
  - I2C master
  - SPI slave
  - SWP slave
  - Miller interface
  - GPIO ports

The TOE has a global alarm system that puts the TOE into a secure state after tamper detection.



### 1.4.1.2 Firmware

The TOE Firmware consists of the Boot software, which provides secure start-up and contains the Flash Loader code.

### 1.4.1.3 Libraries

The TOE can be ordered with the following certified libraries to support the security coding of embedded software:

- UMSLC library to test the chips sensors
- HSL library to provide tearing safe write for the NVM
- PQ Crypto Suite library to provide certified MLKEM Post Quantum algorithms.

## 1.4.2 Physical scope

### 1.4.2.1 Hardware/Firmware

**Table 1 Hardware/Firmware components**

Component	Version
Hardware	IFX_CCI_000068h IFX_CCI_000080h
Design step	G12
Firmware	80.505.04.1
Flash Loader	10.01.0001

### 1.4.2.2 Libraries

The following libraries can be optionally ordered.

**Table 2 Libraries**

Component	Version
HSL	04.05.0040
UMSLC	02.01.0040
PQ Crypto Suite	5.00.012

*Note: The user guidance for the UMSLC and HSL libraries is located in the Programmers Reference Manual. The PQ Crypto Suite library comes with dedicated user guidance.*

### 1.4.2.3 User guidance documents

**Table 3 User guidance**

Component	Version	Date
TEGRION™ SLC21 (32-bit Security Controller – V24) Hardware Reference Manual	6.0	2024-06-26
TEGRION™ SLx2 security controller family Programmer's Reference Manual SLx2_DFP	1.4.0	2024-05-22

## Introduction (ASE\_INT)

Component	Version	Date
SLC21 32-bit Security Controller - V24 Security Guidelines	1.00-3001	2023-07-26
SLC21 (32-bit Security Controller – V24) Production and personalization manual Flash Loader V10	10.01	2023-06-28
Crypto2304T V4, User Manual	3.0	2024-06-21
TEGRION™ SLC21 (32-bit Security Controller – V24) Errata sheet	3.0	2024-01-09
CS-SLC21V24 CryptoSuite 32-bit Security Controller User interface manual	5.00.012	2024-08-27

### 1.4.3 Logical scope

The logical scope of the TOE consists of the logical security features provided by the TOE. These features are listed in chapter 1.3.2. This chapter explains the features in more detail.

#### 1.4.3.1 TSF

The following features of the TOE are part of the TSF:

- The Processor has a duplicated CPU running in lockstep mode to detect integrity errors. The CPU registers and the cache RAM are protected by 32-bit ECC codes used as an EDC.
- ROM, RAM and NVM content is cryptographically encrypted according to [AIS 46]
- ROM, RAM and NVM content is integrity protected by an EDC with at least 28 bits.
- A hardware true RNG according to class PTG.2 of [AIS 31].
- A symmetric coprocessor for performing masked AES ECB encryption.
- The data buses connecting the CPU and the cryptographic peripherals are encrypted using a bus encryption with dynamic keys which are changed in each transfer.
- Peripheral access control can be used to provide individual access control of all peripherals for the different security states of the processor (i.e. secure/non-secure, privilege/non-privilege).
- The chip has the following sensors
  - voltage low and high
  - temperature low and high
  - low frequency
  - light fault attack detectors
 If the values are out of range a security alarm is issued.
- Security Life control is used to check proper working of sensors and alarm system by runtime triggered tests.
- In case the core or a peripheral detects a security violation it performs three countermeasures
  - goes into local alarm state
  - propagates the alarm to the other peripherals and core which then go also into alarm state
  - triggers a security reset.
- The HSL detects if NVM has not been correctly written due to a tearing event. The next time an HSL function is called, the embedded software is informed by the HSL that a tearing event has occurred. The HSL provides functions to correct the corrupted data by either roll-back or roll-forward.

## Introduction (ASE\_INT)

- The Armv8-M Memory Protection Unit (MPU) and Security Attribution Unit (SAU) with 8 regions each are provided which can be used as a logical firewall for the embedded software.
- If the chip is switched to User mode it cannot be switched back to Test mode. If the Flash Loader is permanently disabled, it cannot be reactivated again.
- The Masked Instruction Set Extension (MISE) coprocessor provides an Armv8-M Custom Data path Extension (CDE) with side-channel improved (masked) variants of common 32-bit instructions. The most important instructions are MAND, MBIC, MEOR, MORN, MORR, MADD and MSUB (with and without carries).
- The optional PQ Crypto Suite supporting MLKEM key generation, encapsulation and decapsulation as well as SHA-3 Hash, XOF and RNG functions.
- The processor system comes with several supporting features to assist side-channel protected software implementations. One common software countermeasure is masking. However, the user needs to take special care when processing masked data together with its masks to avoid that they (or parts of them) are unintentionally combined in hardware resulting in a degradation of the desired side-channel security level. Therefore, the processor system has several measures in place to support the software in keeping the masked data and masks separated.
- Fast Random Source (FRS) nonce generator coprocessor. This RNG was not evaluated according to [AIS 31] and its output shall therefore not be used for applications requiring a certified RNG. It is used internally to support security features of the security architecture.
- Program flow integrity protection: The Instruction Stream Signature (ISS) coprocessor can optionally be used by the IC embedded software to detect illegal program flows and trigger an alarm.
- A coprocessor for accelerating long arithmetic operations to support RSA and ECC cryptography. This coprocessor has no dedicated security countermeasures. The embedded software must implement security countermeasures.

The TOE has memory-mapped registers as interfaces to the peripherals.

The interfaces to the libraries are C language APIs.

### 1.4.4 TOE delivery

The TOE delivery formats and delivery lifecycle according to [PP0084] application note 1 are shown in the following table.

**Table 4** Forms of delivery

Component	Format	Life cycle	Delivery method
Hardware	bare die (sawn wafer)	3	Customer chooses delivery method. This ST describes under which circumstances transport protection is provided by the TOE.
	PG-X2QFN-20	4	
	CSP	4	
Firmware	binary image	3 or 4 <sup>1</sup>	In ROM/NVM of hardware
Libraries	object files	n/A	secure download via iShare
Documents	personalized PDF	n/A	secure download via iShare

<sup>1</sup> depends on hardware delivery format

### 1.4.5 Production sites

The TOE may be handled at different production sites but the silicon is produced at TSMC fab 15 in Taiwan only. The production site can be determined by reading out the GCIM.

### 1.4.6 Configurations

This TOE is represented by various configurations called products. The module design, layout and footprint, of all products are identical. The degree of freedom for configuring the TOE is predefined by Infineon Technologies AG. Table 5 shows TOE hardware/firmware configurations. The chip must be ordered with the desired NVM value. The value cannot be changed afterwards. Bill per Use is not supported.

**Table 5 TOE configuration options**

Component	Values	Identification
NVM	800, 1024, 1800 kb	IFX mailbox

### 1.4.7 Initialisation with embedded software

This TOE is equipped with Flash Loader software (FL) to download user software, i.e. an operating system and applications. Various options can be chosen by the user to store software onto the NVM.

**Table 6 Order Options to initialize the TOE with customer software**

Option	TOE status
The user or/and a subcontractor downloads the software into the NVM. Infineon Technologies does not receive any user software.	The Flash Loader can be activated or reactivated by the user or subcontractor to download software into NVM. In case the Flash Loader is active, it may be either in life cycle stage “Pinletter” or “Activated”. When “Activated” a mutual authentication needs to be performed. In “Pinletter” a valid Pinletter provided by Infineon Technologies AG needs to be presented to enter “Activated” stage.
The user provides software to download into NVM to Infineon Technologies AG. The software is loaded into NVM during chip production.	There is no Flash Loader present.
The user provides software to download into NVM to Infineon Technologies AG. The software is loaded into NVM during chip production.	The Flash Loader is blocked by Infineon but can be activated or reactivated by the user or subcontractor to download software into NVM. The user is required to provide a reactivation procedure as part of the software to Infineon Technologies AG.
The user provides software to download into NVM to Infineon Technologies AG. The software is loaded into NVM during chip production.	The Flash Loader is active. The user can either download software or activate the software already present in NVM.

## 2 Conformance (ASE\_CCL)

### 2.1 Conformance claims

This ST and TOE claim conformance to

- [CC2] extended
- [CC3] conformant

#### 2.1.1 PP claims

This ST is strictly conformant to [PP0084]. The assurance level is EAL6 with the augmentation ALC\_FLR.1.

The Security IC Platform Protection Profile with Augmentation Packages is registered and certified by the Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference [PP0084].

#### 2.1.2 Package claims

This ST claims conformance to the following additional packages taken from [PP0084]:

- Package Authentication of the Security IC, section 7.2, conformant. This package is only claimed in case the Flash Loader is not permanently deactivated.
- Package Loader, Package 1: Loader dedicated for usage in secured environment only, section 7.3.1, conformant.
- Package Loader, Package 2: Loader dedicated for usage by authorized users only, section 7.3.2, augmented.
- Package AES; section 7.4.2, conformant.

The assurance level for the TOE is EAL6 augmented with the component ALC\_FLR.1. Therefore, this ST is package-augmented to the packages in [PP0084].

### 2.2 Conformance rationale

The TOE is a typical security IC as defined in [PP0084].

The security problem definition of [PP0084] is enhanced by adding the Organisational Security Policy P.Firewall due to addition of the Armv8-M Memory Protection Unit and Security Extension. The security target remains conformant to [CC1] due to claim 289 as the possibility to introduce additional restrictions is given. The security target fulfils the strict conformance claim of [PP0084] due to application note 5.

### 3 Security Problem Definition (ASE\_SPD)

#### 3.1 Threats

##### 3.1.1 Threats from PP0084

The following threats are defined and described in [PP0084] sections 3.2 and 7.2.1.

**Table 7 Threats from [PP0084]**

Threat	Description
T.Phys-Manipulation	Physical Manipulation
T.Phys-Probing	Physical Probing
T.Malfunction	Malfunction due to Environmental Stress
T.Leak-Inherent	Inherent Information Leakage
T.Leak-Forced	Forced Information Leakage
T.Abuse-Func	Abuse of Functionality
T.RND	Deficiency of Random Numbers
T.Masquerade_TOE	Masquerade the TOE

#### 3.2 Organizational security policies

##### 3.2.1 Organizational security policies from PP0084

The organizational policies from [PP0084] sections 3.3, 7.3.1, 7.3.2 and 7.4 are applicable.

**Table 8 Organisational Security Policies from [PP0084]**

OSP	Description
P.Process-TOE	Protection during TOE Development and Production
P.Crypto-Service	Cryptographic services of the TOE
P.Lim_Block_Loader	Limiting and Blocking the Loader Functionality
P.Ctrl_Loader	Controlled usage to Loader Functionality

##### 3.2.2 Organizational security policies defined in this ST

This ST defines an additional organisational security policy specific to the MPU Extension and Security Extension.

**Table 9 Organisational Security Policies defined in this ST**

OSP	Definition
P.Firewall	The TOE must enable the IC dedicated software and the end-user embedded software to manage and control access to regions in memory.

### 3.3 Assumptions

#### 3.3.1 Assumptions defined in [PP0084]

The TOE assumptions about the operational environment are defined and described in [PP0084] section 3.4.

**Table 10 Assumptions from [PP0084]**

Assumption	Description
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalization
A.Resp-Appl	Treatment of User Data

#### 3.3.2 Assumptions defined in this ST

There are no additional assumptions defined in this ST.

## 4 Security Objectives (ASE\_OBJ)

### 4.1 Security objectives for the TOE

#### 4.1.1 Security objectives for the TOE defined in PP0084

**Table 11 Security objectives for the TOE from [PP0084]**

Objective	Description
O.Phys-Manipulation	Protection against Physical Manipulation
O.Phys-Probing	Protection against Physical Probing
O.Malfunction	Protection against Malfunctions
O.Leak-Inherent	Protection against Inherent Information Leakage
O.Leak-Forced	Protection against Forced Information Leakage
O.Abuse-Func	Protection against Abuse of Functionality
O.Identification	TOE Identification
O.RND	Random Numbers
O.Cap_Avail Loader	Capability and availability of the Loader
O.Ctrl_Auth Loader	Access control and authenticity for the Loader
O.Authentication	Authentication to external entities
O.AES	Cryptographic service AES

#### 4.1.2 Security objectives for the TOE defined in this ST

**Table 12 Security Objectives for the TOE**

Objective	Definition
O.Firewall	<p><b>Firewall based Access Control</b></p> <p>The TOE must provide the IC dedicated software and the end-user embedded software with the capability to define restricted memory access and code execution to memory addresses. The TOE must enforce the access of software to these memory regions depending on access attributes.</p>
O.MLKEM	<p><b>MLKEM cryptographic services</b></p> <p>The TOE shall provide the following specific security functionality to the Smartcard Embedded Software: Module Lattice key Encapsulation Mechanism.</p>
O.Hash	<p><b>Cryptographic service hash</b></p> <p>The TOE provides secure cryptographic services for Hash generation and Extended Output Functions.</p>



## 4.2 Security objectives for the operational environment (OE)

### 4.2.1 OEs defined in [PP0084]

**Table 13 Security objectives for the operational environment from [PP0084]**

Objective	Description
OE.Resp-Appl	Treatment of user data of the Composite TOE
OE.Process-Sec-IC	Protection during composite product manufacturing
OE.Lim_Block_Loader	Limitation of capability and blocking the Loader
OE.Loader_Usage	Secure communication and usage of the Loader
OE.TOE_Auth	External entities authenticating of the TOE

Note: OE.TOE\_Auth is available if the Flash Loader is available.

### 4.2.2 OEs defined in this ST

**Table 14 Additional Security objectives for the environment**

Environmental objective	Description
OE.Secure_Delivery (only applicable if Flash Loader deactivated (i.e. cases 2 and 3 from Table 1Table 6))	When the TOE is ordered with a disabled Flash Loader, it does not provide transport protection. Therefore, technical and / or organisational security procedures (e.g. a custom mutual authentication mechanism or a security transport) should be put in place by the customer to secure the personalized TOE during delivery as required by the security needs of the loaded IC Embedded Software.

## 4.3 Security objectives rationale

The security objectives rationale of the TOE is defined and described in [PP0084] section 4.4, 7.3.1, 7.3.2 and section 7.4.2.

The objectives O.Firewall added in this ST cover the organisational security policy P.Firewall that states that IC dedicated software and end-user embedded software must be able to manage and control access to regions in memory.

The objectives O.Hash and O.MLKEM cover the policy P.Crypto-Service. This policy intends to allow adding various cryptographic services to the TSF.

The objective OE.Secure\_Delivery requires the customers to provide transport protection in case the TOE is delivered with flash loader deactivated (i.e. cases 2, 3 from Table 6). In that case O.Authentication is not available and needs to be compensated by customer measures. This environmental objective counters T.Masquerade\_TOE.

## 5 Extended Components Definition (ASE\_ECD)

### 5.1 Extended components defined in [PP0084]

The [PP0084] defines the following extended components used in this ST:

- FMT\_LIM.1
- FMT\_LIM.2
- FAU\_SAS.1
- FDP\_SDC.1
- FCS\_RNG.1
- FIA\_API.1

### 5.2 Extended components defined in this ST

There are no extended components defined in this ST.

## 6 Security Requirements (ASE\_REQ)

### 6.1 Security functional requirements

For the CC operations the following convention is used:

- CC operations which have been already completed in [PP0084] or [AIS 31] are typeset without underline.
- CC (nested) iteration operations are started by a slash “/” symbol, followed by an iteration identifier text. Iterations may be recursively nested.
- CC operations which are completed in this ST are underlined and the assigned footnote shows the original template text. Iteration operations are typed in normal font (i.e. without underline).

#### 6.1.1 Hardware random number generators

Random numbers generation according to **Class PTG.2** of [AIS 31].

**Table 15 FCS\_RNG.1/TRNG**

FCS_RNG.1/TRNG	Random Number Generation
Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1/TRNG	<p>The TSF shall provide a physical random number generator that implements:</p> <p>(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.</p> <p>(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG <u>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</u><sup>1</sup>.</p> <p>(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.</p> <p>(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.</p> <p>(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered <u>continuously</u><sup>2</sup>. 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.</p>
FCS_RNG.1.2/TRNG	The TSF shall provide <u>32-bit numbers</u> <sup>3</sup> that meet

<sup>1</sup> [selection: prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy]

<sup>2</sup> [selection: externally, at regular intervals, continuously, applied upon specified internal events].

<sup>3</sup> [selection: bits, octets of bits, numbers [assignment: format of the numbers]]

FCS_RNG.1/TRNG	Random Number Generation
	(PTG.2.6) Test procedure A ( <u>None</u> ) <sup>1</sup> does not distinguish the internal random numbers from output sequences of an ideal RNG. (PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.

### 6.1.2 Cryptographic services implemented in hardware

This chapter defines cryptographic algorithms which are directly supported by the hardware.

**Table 16 FCS\_COP.1/AES**

FCS_COP.1/AES	Cryptographic operation
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4
FCS_COP.1.1/AES	The TSF shall perform decryption and encryption in accordance with a specified cryptographic algorithm AES in <u>ECB mode</u> <sup>2</sup> and cryptographic key sizes <u>128 bit, 192 bit, 256 bit</u> <sup>3</sup> that meet the following: [FIPS 197], [SP 800-38A].

*Note: The input to the AES algorithm must be provided in two XOR shares. By fixing one share to zero a standard ECB mode results.*

**Table 17 FCS\_CKM.4**

FCS_CKM.4	Cryptographic key destruction
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method <u>overwriting or zeroing</u> <sup>4</sup> that meets the following: <u>None</u> <sup>5</sup> .

### 6.1.3 TSF testing

An attacker may try to circumvent the alarm system and secure wiring by physical manipulation (e.g. by cutting alarm lines). To counter those threats, the chip provides the User Mode Life Cycle (UMSLC) tests to check the integrity of those security features. Those test functions are provided as a software library and can be triggered on demand of the Embedded Software of the Composite TOE.

<sup>1</sup> [assignment: additional standard test suites]

<sup>2</sup> [selection: 128 bit, 192 bit, 256 bit]

<sup>3</sup> [selection: 128 bit, 192 bit, 256 bit]

<sup>4</sup> [assignment: cryptographic key destruction method]

<sup>5</sup> [assignment: list of standards]

Table 18 TSF testing

FPT_TST.1	TSF testing
Hierarchical to	No other components.
Dependencies	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of self-tests <u>at the request of the authorized user<sup>1</sup></u> to demonstrate the correct operation of <u>alarm test and security optimized wiring tests<sup>2</sup></u> .
FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of <u>the boot code<sup>3</sup></u> .
FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of <u>alarm behavior and security optimized wiring<sup>4</sup></u> .

*Note: If the integrity of the boot code is violated, a security reset is triggered. The authorized user (i.e. the embedded software) can check if a security reset has been performed by reading the reset status register.*

### 6.1.4 Malfunctions

This chapter relates to the section “Malfunctions” in [PP0084] ch. 6.1.

The SFRs FRU\_FLT.2 and FPT\_FLS.1 are specified in [PP0084].

#### Secure state of the TOE

Application note 14 of FPT\_FLS.1 requires to define the secure state of the TOE.

Definition: A **secure state** of the TOE is either a correct operation or one of the following exceptional states

- security reset
- global deactivation of the TOE (a.k.a. alarm state)
- fault handler

### 6.1.5 Abuse of Functionality

This chapter relates to the section “Abuse of Functionality” in [PP0084] ch. 6.1.

The SFRs FMT\_LIM.1 and FMT\_LIM.2 are specified in [PP0084].

Table 19 FAU\_SAS.1

FAU_SAS.1	Audit Storage
Hierarchical to	No other components.
Dependencies	No dependencies.

<sup>1</sup> [selection: during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self-test should occur]]

<sup>2</sup> [selection: [assignment: parts of TSF], the TSF]

<sup>3</sup> [selection: [assignment: parts of TSF data], TSF data]

<sup>4</sup> [selection: [assignment: parts of TSF], TSF]

## Security Requirements (ASE\_REQ)

FAU_SAS.1	Audit Storage
FAU_SAS.1.1	The TSF shall provide <u>the test process before TOE delivery</u> <sup>1</sup> with the capability to store <u>the initialization data and/or pre-personalization data and/or supplements of the security IC embedded software</u> <sup>2</sup> in the <u>access protected and not changeable areas of the non-volatile memory</u> <sup>3</sup> .

### 6.1.6 Physical Manipulation and Probing

This chapter relates to the section “Physical Manipulation and Probing” in [PP0084] ch. 6.1.

The SFR FPT\_PHP.3 is specified in [PP0084].

#### Automatic response of the TOE

Application note 19 of FPT\_PHP.3 requires to define the automatic response of the TOE.

Definition: An **automatic response of the TOE** means entering a secure state of the TOE.

**Table 20 FDP\_SDC.1**

FDP_SDC.1	Stored data confidentiality
Hierarchical to	No other components.
Dependencies	No dependencies
FDP_SDC.1.1	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the <u>RAM, ROM, and NVM</u> <sup>4</sup> .

**Table 21 FDP\_SDI.2**

FDP_SDI.2	Stored data integrity monitoring and action
Hierarchical to	FDP_SDI.1
Dependencies	No dependencies.
FDP_SDI.2.1	The TSF shall monitor user data stored in containers controlled by the TSF for <u>EDC integrity errors</u> <sup>5</sup> on all objects, based on the following attributes: <u>the corresponding EDC value with a length of at least 28 bits in the RAM, ROM, and NVM</u> <sup>6</sup> .
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall <u>enter a secure state</u> <sup>7</sup> .

<sup>1</sup> [assignment: list of subjects]

<sup>2</sup> [assignment: list of audit information]

<sup>3</sup> [assignment: type of persistent memory]

<sup>4</sup> [assignment: memory area]

<sup>5</sup> [assignment: integrity errors]

<sup>6</sup> [assignment: user data attributes]

<sup>7</sup> [assignment: action to be taken]

### 6.1.7 Leakage

This chapter relates to the section “Leakage” in [PP0084] ch. 6.1.

The SFRs FDP\_ITT.1, FPT\_ITT.1 and FDP\_IFC.1 are specified in [PP0084].

### 6.1.8 Application Firewall

The Application Firewall allows the embedded software to execute in four security levels and to assign access conditions to the address space related to the security levels. The security levels are:

- secure privilege
- secure non-privilege
- non-secure privilege
- non-secure non-privilege

The policy allows the embedded software to enforce the following trust relationship

- secure doesn't trust non-secure independent of the privilege level
- secure privilege doesn't trust secure non-privilege
- non-secure privilege doesn't trust non-secure non-privilege

#### 6.1.8.1 Policy definition

##### Subjects:

- Processor

##### Objects:

- Memory addresses

##### Operations:

- FETCH(x): any instruction fetch from address x
- READ(x): any read access from address x
- WRITE(x): any write access to address x
- SG: secure gateway instruction
- BNS: any of the branch to non-secure code instructions
- FNC\_RETURN: return from secure mode to non-secure mode
- HANDLER\_S: call of any secure handler code. Of specific importance for this policy are the following handlers:
  - MEMFAULT\_S: memory fault handler in secure mode
  - SECFAULT: security fault handler
- HANDLER\_NS: call of any non-secure handler code. Of specific importance for this policy is the following handler:
  - MEMFAULT\_NS: memory fault handler in non-secure mode
- EXC\_RETURN: return from handler code

**Security attributes for processor:**

- sec: Boolean attribute designating secure/non-secure with values
  - true: processor is in secure mode.
  - false: processor is non-secure mode.
  
- handlermode: Boolean attribute designating handler / thread mode:
  - true: processor is in handler mode
  - false: processor is in thread mode
  
- nPriv\_S: Boolean attribute designating privilege mode when sec = true with values.
  - true: processor runs in non-privilege mode.
  - false: processor runs in privilege mode.
  
- nPriv\_NS: Boolean attribute designating privilege mode when sec = false with values.
  - true: processor runs in non-privilege mode.
  - false: processor runs in privilege mode.

**Security attributes for addresses:**

- PO(x): Boolean attribute assigned to address x.
  - true: Only privilege mode has access.
  - false: Privilege and non-privilege mode have access.
  
- acc(x): {N, R, RW} attribute assigned to address x.
  - N: no access allowed.
  - R: write access is declined
  - RW: read and write access is not declined.
  
- sec(x): {S, NS, NSC} attribute assigned to address x.
  - S: secure address.
  - NS: non-secure address.
  - NSC: secure address which is callable from non-secure address.
  
- XN(x): Boolean variable assigned to address x.
  - true: instruction fetch is declined.
  - false: instruction fetch is not declined.

**Definitions:**

- privileged := handlermode or (not nPriv\_S and sec) or (not nPriv\_NS and not sec)

**Rules:**



### Security Requirements (ASE\_REQ)

1. FETCH(x) is declined if  
(sec = false and sec(x) = S )  
or (sec = false and sec(x) = NSC and FETCH(x) ≠ SG)
2. READ(x) is declined if  
sec = false and sec(x) ≠ NS
3. WRITE(x) is declined if  
sec = false and sec(x) ≠ NS
4. FETCH(x) is declined if  
(privileged = false and PO(x) = true)  
or (XN(x) = true)  
or (acc(x) = N)
5. READ(x) is declined if  
(privileged = false and PO(x) = true)  
or (acc(x) = N)
6. WRITE(x) is declined if  
(privileged = false and PO(x) = true)  
or (acc(x) ≠ RW)
7. If one of rules 1, 2, 3 apply then the SECFault handler will be called.
8. If one of rules 4, 5 or 6 apply but none of rules 1, 2, 3 and sec=true then MEMFAULT\_S handler will be called.
9. If one of rules 4, 5 or 6 apply but none of rules 1, 2, 3 and sec=false then MEMFAULT\_NS handler will be called.
10. Modification of sec to value true is only allowed for SG, FNC\_RETURN, EXC\_RETURN or HANDLER\_S.
11. Modification of sec to value false is only allowed for BNS and EXC\_RETURN.
12. Modification of nPriv\_S to value false is only allowed when handlermode = true and sec = true.
13. Modification of nPriv\_S to value true is only allowed when sec = true.
14. Modification of nPriv\_NS to value false is only allowed when handlermode = true  
or (sec = true and nPriv\_S = false).
15. Modification of handlermode to value true is only allowed for HANDLER\_S or HANDLER\_NS.
16. Modification of handlermode to value false is only allowed for EXC\_RETURN.
17. Modification of nPriv\_NS to value true is only allowed when privileged = true

#### Roles for management:

The parameter x designates any address.

- secure AF management: privileged = true and sec = true
- non-secure AF management: privileged = true

### 6.1.8.2 SFRs

**Table 22 FDP\_ACC.2/AF**

FDP_ACC.2/AF	Complete access control
Hierarchical to	FDP_ACC.1

## Security Requirements (ASE\_REQ)

<b>FDP_ACC.2/AF</b>	<b>Complete access control</b>
Dependencies	FDP_ACF.1
FDP_ACC.2.1/AF	The TSF shall enforce the <u>Application Firewall access control policy</u> <sup>1</sup> on <u>subjects, objects and operations defined in 6.1.8.1</u> <sup>2</sup> and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/AF	The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

Table 23 FDP\_ACF.1/AF

<b>FDP_ACF.1/AF</b>	<b>Security attribute based access control</b>
Hierarchical to	No other components.
Dependencies	FDP_ACC.1 FMT_MSA.3
FDP_ACF.1.1/AF	The TSF shall enforce the <u>Application Firewall access control policy</u> <sup>3</sup> to objects based on the following: <u>The subjects, objects, operations and associated security attributes defined in 6.1.8.1</u> <sup>4</sup> .
FDP_ACF.1.2/AF	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: <u>Rules defined in 6.1.8.1</u> <sup>5</sup> .
FDP_ACF.1.3/AF	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: <u>None</u> <sup>6</sup> .
FDP_ACF.1.4/AF	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: <u>None</u> <sup>7</sup> .

Table 24 FMT\_MSA.3/AF

<b>FMT_MSA.3/AF</b>	<b>Static attribute initialisation</b>
Hierarchical to	No other components.
Dependencies	FMT_MSA.1 FMT_SMR.1
FMT_MSA.3.1/AF	The TSF shall enforce the <u>Application Firewall access control policy</u> <sup>8</sup> to provide <u>restrictive</u> <sup>9</sup> default values for security attributes that are used to enforce the SFP.

<sup>1</sup> [assignment: access control SFP]<sup>2</sup> [assignment: list of subjects and objects, and operations among subjects and objects covered by the SFP]<sup>3</sup> [assignment: access control SFP]<sup>4</sup> [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]<sup>5</sup> [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]<sup>6</sup> [assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects].<sup>7</sup> [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].<sup>8</sup> [assignment: access control SFP, information flow control SFP]<sup>9</sup> [selection, choose one of: restrictive, permissive, [assignment: other property]]

## Security Requirements (ASE\_REQ)

<b>FMT_MSA.3/AF</b>	<b>Static attribute initialisation</b>
FMT_MSA.3.2/AF	The TSF shall allow the <u>none</u> <sup>1</sup> to specify alternative initial values to override the default values when an object or information is created.

Note: Restrictive means that the security attributes for all addresses are  $sec(x) = S$

Table 25 FMT\_MSA.1/AF/S

<b>FMT_MSA.1/AF/S</b>	<b>Management of security attributes</b>
Hierarchical to	No other components.
Dependencies	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1
FMT_MSA.1.1/AF/S	The TSF shall enforce the <u>Application Firewall access control policy</u> <sup>2</sup> to restrict the ability to <u>modify</u> <sup>3</sup> the security attributes <u>PO(x)</u> , <u>acc(x)</u> , <u>sec(x)</u> , <u>XN(x)</u> <sup>4</sup> to <u>secure AF management in case <math>sec(x) = S</math></u> <sup>5</sup> .

Table 26 FMT\_MSA.1/AF/NS

<b>FMT_MSA.1/AF/NS</b>	<b>Management of security attributes</b>
Hierarchical to	No other components.
Dependencies	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1
FMT_MSA.1.1/AF/NS	The TSF shall enforce the <u>Application Firewall access control policy</u> <sup>6</sup> to restrict the ability to <u>modify</u> <sup>7</sup> the security attributes <u>PO(x)</u> , <u>acc(x)</u> , <u>XN(x)</u> <sup>8</sup> to <u>secure or non-secure AF management in case <math>sec(x) = NS</math></u> <sup>9</sup> .

Table 27 FMT\_SMF.1/AF

<b>FMT_SMF.1/AF</b>	<b>Specification of management functions</b>
Hierarchical to	No other components.
Dependencies	No dependencies.
FMT_SMF.1.1/AF	The TSF shall be capable of performing the following management functions: <u>Modification of the security attributes PO(x)</u> , <u>acc(x)</u> , <u>sec(x)</u> , <u>XN(x)</u> <sup>10</sup> .

<sup>1</sup> [assignment: the authorized identified roles]

<sup>2</sup> [assignment: access control SFP(s), information flow control SFP(s)]

<sup>3</sup> [selection: change\_default, query, modify, delete, [assignment: other operations]]

<sup>4</sup> [assignment: list of security attributes]

<sup>5</sup> [assignment: the authorized identified roles]

<sup>6</sup> [assignment: access control SFP(s), information flow control SFP(s)]

<sup>7</sup> [selection: change\_default, query, modify, delete, [assignment: other operations]]

<sup>8</sup> [assignment: list of security attributes]

<sup>9</sup> [assignment: the authorized identified roles]

<sup>10</sup> [assignment: list of management functions to be provided by the TSF]

Table 28 FMT\_SMR.1/AF

FMT_SMR.1/AF	Security Roles
Hierarchical to	No other components.
Dependencies	FIA_UID.1
FMT_SMR.1.1/AF	The TSF shall maintain the roles <u>secure AF management</u> and <u>non-secure AF management</u> <sup>1</sup> .
FMT_SMR.1.2/AF	The TSF shall be able to associate users with roles.

### 6.1.9 Authentication of the Security IC

The TOE shall implement the Package “Authentication of the Security IC” from [PP0084], ch. 7.2.

Table 29 FIA\_API.1

FIA_API.1	Authentication Proof of Identity
Hierarchical to	No other components.
Dependencies	No dependencies.
FIA_API.1.1	The TSF shall provide a <u>authentication mechanism according to [ISO9798 2] section 7.3.3, Mechanism MUT.CR-Three-pass authentication</u> <sup>2</sup> to prove the identity of the TOE to an external entity.

Note: FIA\_API is only available, if the Flash Loader is active.

### 6.1.10 Flash loader

The TOE provides a Flash Loader to download user data into the NVM, either during production of the TOE or at customer site. This TOE shall support both Loader packages from [PP0084] section 7.3.

- Package 1: Loader dedicated for usage in secured environment only
- Package 2: Loader dedicated for usage by authorized users only

The SFRs FDP\_UCT.1 and FDP\_UIT.1 are specified in [PP0084].

Table 30 FMT\_LIM.1/Loader

FMT_LIM.1/Loader	Limited Capabilities - Loader
Hierarchical to	No other components.
Dependencies	FMT_LIM.2
FMT_LIM.1.1/Loader	The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced: Deploying Loader functionality after <u>permanent deactivation</u> <sup>3</sup> does not allow stored user data to be disclosed or manipulated by unauthorized user.

<sup>1</sup> [assignment: the authorised identified roles]

<sup>2</sup> [assignment: authentication mechanism]

<sup>3</sup> [assignment: action]

Table 31 FMT\_LIM.2/Loader

FMT_LIM.2/Loader	Limited availability - Loader
Hierarchical to	No other components.
Dependencies	FMT_LIM.1
FMT_LIM.2.1/Loader	The TSF shall be designed in a manner that limits its availability so that in conjunction with “Limited capabilities (FMT_LIM.1)” the following policy is enforced: The TSF prevents deploying the Loader functionality after <u>permanent deactivation</u> <sup>1</sup> .

Note: The User Guidance for this TOE requires the Flash Loader to be permanently deactivated prior delivery to the end user (Phase 7).

Table 32 FTP\_ITC.1

FTP_ITC.1	Inter-TSF trusted channel
Hierarchical to	No other components.
Dependencies	No dependencies.
FTP_ITC.1.1	The TSF shall provide a communication channel between itself and <u>Administrator User or Download Operator User and Image Provider</u> <sup>2</sup> that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit another trusted IT product to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for <u>deploying Loader for downloading User Data and modification of authentication keys</u> <sup>3</sup> .

Note: The download operation is authenticated by the Administrator User or the Download Operator User but the download image may be encrypted and authenticated by a different role. This role is called the “Image Provider”. Thus, the download operation provides in effect a trusted channel between the Image Provider and the Flash Loader.

Table 33 FDP\_ACC.1/Loader

FDP_ACC.1/Loader	Subset access control - Loader
Hierarchical to	No other components.
Dependencies	FDP_ACF.1
FDP_ACC.1.1/Loader	The TSF shall enforce the Loader SFP on (1) the subjects <u>Administrator User, Download Operator User and Image Provider</u> <sup>4</sup> , (2) the objects user data in <u>NVM</u> <sup>5</sup> , (3) the operation deployment of Loader.

<sup>1</sup> [assignment: action]

<sup>2</sup> [assignment: users authorized for using the Loader]

<sup>3</sup> [assignment: rules]

<sup>4</sup> [assignment: authorized roles for using Loader]

<sup>5</sup> [assignment: memory areas]

Table 34 FDP\_ACF.1/Loader

FDP_ACF.1/Loader	Security attribute based access control - Loader
Hierarchical to	No other components.
Dependencies	FMT_MSA.3
FDP_ACF.1.1/Loader	The TSF shall enforce the Loader SFP to objects based on the following: (1) the subjects <u>Administrator User, Download Operator User and Image Provider</u> <sup>1</sup> with security attributes <u>None</u> <sup>2</sup> . (2) the objects user data in <u>NVM</u> <sup>3</sup> with security attributes <u>None</u> <sup>4</sup> .
FDP_ACF.1.2/Loader	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: <u>The authenticated Administrator User or authenticated Download Operator User can replace the user data by new user data when the new user data is authorized by the Image Provider</u> <sup>5</sup> .
FDP_ACF.1.3/Loader	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: <u>None</u> <sup>6</sup> .
FDP_ACF.1.4/Loader	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: <u>None</u> <sup>7</sup> .

Note: The Image provider authenticates with the flash loader implicitly by providing a correctly signed and encrypted download image. An Image provider authentication must always be preceded by an Administrator User or Download Operator User authentication.

### 6.1.10.1 SFRs added in this ST

The following SFRs have been added to the SFRs from Flash Loader package 2 of [PP0084] in order to describe the management of the various Flash Loader authentication keys.

Table 35 FMT\_MTD.1/Loader

FMT_MTD.1/Loader	Management of TSF data
Hierarchical to	No other components.
Dependencies	FMT_SMR.1 FMT_SMF.1
FMT_MTD.1.1/Loader	The TSF shall restrict the ability to <u>modify, delete</u> <sup>8</sup> the <u>Authentication keys for Administrator User, Download Operator User and Image Provider</u> <sup>9</sup> to <u>Administrator User, Download Operator User</u> <sup>10</sup> .

<sup>1</sup> [assignment: authorized roles for using Loader]

<sup>2</sup> [assignment: SFP relevant security attributes, or named groups of SFP relevant security attributes]

<sup>3</sup> [assignment: memory areas]

<sup>4</sup> [assignment: SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

<sup>5</sup> [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

<sup>6</sup> [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects].

<sup>7</sup> [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

<sup>8</sup> [selection: change\_default, query, modify, delete, clear, [assignment: other operations]]

<sup>9</sup> [assignment: list of TSF data]

<sup>10</sup> [assignment: the authorised identified roles]

## Security Requirements (ASE\_REQ)

*Note: The Administrator User can manage the keys for Administration User, Download Operator User and Image Provider.*

*The Download Operator User can delete the key for Image Provider and Download Operator and modify keys for the Download Operator User only.*

*The image provider cannot modify any keys or perform authentication with the Flash Loader. It can only build encrypted and authenticated loadable images.*

**Table 36 FMT\_SMR.1/Loader**

<b>FMT_SMR.1/Loader</b>	<b>Security roles</b>
Hierarchical to	No other components.
Dependencies	FIA_UID.1
FMT_SMR.1.1/Loader	The TSF shall maintain the roles <u>Administrator User, Download Operator User, Image Provider</u> <sup>1</sup> .
FMT_SMR.1.2/Loader	The TSF shall be able to associate users with roles.

*Note: Image provider is the role who maintains the key which is used to encrypt and integrity protect the download image.*

**Table 37 FMT\_SMF.1/Loader**

<b>FMT_SMF.1/Loader</b>	<b>Specification of Management Functions</b>
Hierarchical to	No other components.
Dependencies	No dependencies.
FMT_SMF.1.1/Loader	The TSF shall be capable of performing the following management functions: <u>Change Key, Invalidate Key</u> <sup>2</sup> .

*Note: “Change Key” of this SFR means the “modify” operations from SFR FMT\_MTD.1/Loader, “Invalidate Key” of this SFR means the “delete” operation from SFR FMT\_MTD.1/Loader.*

**Table 38 FIA\_UID.2/Loader**

<b>FIA_UID.2/Loader</b>	<b>User identification before any action</b>
Hierarchical to	FIA_UID.1
Dependencies	No dependencies.
FIA_UID.2.1/Loader	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

## 6.1.11 PQ Crypto Suite

This chapter defines the SFRs related to the optional PQ Crypto Suite. Please note that the SFRs from this chapter are only applicable if the optional PQ Crypto Suite is delivered.

<sup>1</sup> [assignment: the authorised identified roles]

<sup>2</sup> [assignment: list of management functions to be provided by the TSF]

### 6.1.11.1 Hash

This section describes hash related algorithms provided by the PQ Crypto Suite.

**Table 39** FCS\_COP.1/CS/Hash/<iter>

FCS_COP.1/CS/Hash/<iter>	Cryptographic operation
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4
FCS_COP.1.1/CS/Hash/<iter>	The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithms] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

The operations performed in this SFR are defined in the following table. Please note that <iter> is a placeholder for different SFR iterations defined in the first column.

**Table 40** Cryptographic table for FCS\_COP.1/CS/Hash/<iter>

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
SHA3	Hash digest	SHA3-224, SHA3-256, SHA3-384, SHA3-512	none	[FIPS 202]
SHAKE	XOF	SHAKE-128, SHAKE-256	none	[FIPS 202]

### 6.1.11.2 MLKEM

This section describes MLKEM ciphers provided by the PQ Crypto Suite.

**Table 41** FCS\_COP.1/CS/MLKEM/<iter>

FCS_COP.1/CS/MLKEM/<iter>	Cryptographic operation
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4
FCS_COP.1.1/CS/MLKEM/<iter>	The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithms] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

The operations performed in this SFR are defined in the following table. Please note that <iter> is a placeholder for different SFR iterations defined in the first column.



Table 42 Cryptographic table for FCS\_COP.1/CS/MLKEM/&lt;iter&gt;

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
ENC	encapsulation	ML-KEM-512 ML-KEM-768 ML-KEM-1024	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.2
DEC	decapsulation	ML-KEM-512 ML-KEM-768 ML-KEM-1024	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.3

Table 43 FCS\_CKM.1/CS/MLKEM/&lt;iter&gt;

FCS_CKM.1/CS/ MLKEM /<iter>	Cryptographic key generation
Hierarchical to	No other components.
Dependencies	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4
FCS_CKM.1.1/CS/ MLKEM /<iter>	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].

The operations performed in this SFR are defined in the following table. Please note that <iter> is a placeholder for different SFR iterations defined in the first column.

Table 44 Cryptographic table for FCS\_CKM.1/CS/MLKEM/&lt;iter&gt;

<iter>	[assignment: cryptographic key generation algorithm]	[assignment: cryptographic key sizes] <sup>1</sup>	[assignment: list of standards]
KeyGen	MLKEM key generation	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.1
Recalc	MLKEM key generation with seed	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.1

*Note: Recalc generates a key by using a seed parameter instead of using a random number generator.*

### 6.1.11.3 Random

This section describes random number generation provided by the PQ Crypto Suite.

<sup>1</sup> Public key size

Table 45 FCS\_RNG.1/CS/PTG2

FCS_RNG.1/CS/PTG2	Random Number Generation
Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1/CS/PTG2	<p>The TSF shall provide a physical random number generator that implements:</p> <p>(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.</p> <p>(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG <u>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</u><sup>1</sup>.</p> <p>(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.</p> <p>(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.</p> <p>(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered <u>continuously</u><sup>2</sup>. 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.</p>
FCS_RNG.1.2/CS/PTG2	<p>The TSF shall provide <u>32-bit numbers</u><sup>3</sup> that meet</p> <p>(PTG.2.6) Test procedure A (<u>None</u>)<sup>4</sup> does not distinguish the internal random numbers from output sequences of an ideal RNG.</p> <p>(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.</p>

Note: The PTG.2 API provided by the PQ Crypto Suite is only a wrapper to the hardware-provided PTG.2 defined in section 6.1.1.

Table 46 FCS\_RNG.1/CS/PTG3

FCS_RNG.1/CS/PTG3	Random number generation
Hierarchical to	No other components.
Dependencies	No dependencies.

<sup>1</sup> [selection: prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy]

<sup>2</sup> [selection: externally, at regular intervals, continuously, applied upon specified internal events]

<sup>3</sup> [selection: bits, octets of bits, numbers [assignment: format of the numbers]]

<sup>4</sup> [assignment: additional standard test suites]

FCS_RNG.1/CS/PTG3	Random number generation
FCS_RNG.1.1/CS/PTG3	<p>The TSF shall provide a hybrid physical random number generator that implements:</p> <p>(PTG.3.1) 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.</p> <p>(PTG.3.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG <u>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</u><sup>1</sup>.</p> <p>(PTG.3.3) 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 and the seeding of the DRG.3 post-processing algorithm have been finished successfully or when a defect has been detected.</p> <p>(PTG.3.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.</p> <p>(PTG.3.5) The online test procedure checks the raw random number sequence. It is triggered <u>continuously</u><sup>2</sup>. 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.</p> <p>(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.</p>
FCS_RNG.1.2/CS/PTG3	<p>The TSF shall provide <u>numbers in 32-bit packets</u><sup>3</sup> that meet:</p> <p>(PTG.3.7) Statistical test suites cannot practically distinguish the internal random numbers from output sequences of an ideal RNG. The internal random numbers must pass test procedure A (<u>none</u>)<sup>4</sup>.</p> <p>(PTG.3.8) The internal random numbers shall <u>use PTRNG of class PTG.2 as random source for the post processing</u><sup>5</sup>.</p>

Table 47 FCS\_RNG.1/CS/DRG3

FCS_RNG.1/CS/DRG3	Random number generation
Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1/CS/DRG3	The TSF shall provide a deterministic random number generator that implements:

<sup>1</sup> [selection: prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy]

<sup>2</sup> [selection: externally, at regular intervals, continuously, upon specified internal events]

<sup>3</sup> [selection: bits, octets of bits, numbers [assignment: format of the numbers]]

<sup>4</sup> [assignment: additional test suites]

<sup>5</sup> [selection: use PTRNG of class PTG.2 as random source for the post-processing, have [assignment: work factor], require [assignment: guess work]]

FCS_RNG.1/CS/DRG3	Random number generation
	<p>(DRG.3.1) If initialized with a random seed <u>using a PTRNG of class PTG.2 as random source</u><sup>1</sup>, the internal state of the RNG shall <u>have at least 100 bit of entropy</u><sup>2</sup>.</p> <p>(DRG.3.2) The RNG provides forward secrecy.</p> <p>(DRG.3.3) The RNG provides backward secrecy even if the current internal state is known.</p>
FCS_RNG.1.2/CS/DRG3	<p>The TSF shall provide numbers that meet:</p> <p>(DRG.3.4) The RNG, initialized with a random seed, <u>where the seed has at least 100 bit of entropy and is derived by a PTG.2 certified PTRNG</u><sup>3</sup> generates output for which <u><math>2^{48}</math> strings of bit length 128 are mutually different with probability that is greater than <math>1 - 2^{(-24)}</math></u><sup>5</sup></p> <p>(DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from the output sequences of an ideal RNG. The random numbers must pass test procedure A <u>and the U.S. National Institute of Standards and Technology (NIST) test suite for RNGs used for cryptographic purposes [SP 800-22]</u><sup>6</sup>.</p>

Table 48 FCS\_RNG.1/CS/DRG4

FCS_RNG.1/CS/DRG4	Random number generation
Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1/CS/DRG4	<p>The TSF shall provide a hybrid deterministic random number generator that implements:</p> <p>(DRG.4.1) The internal state of the RNG shall <u>use PTRNG of class PTG.2 as random source</u><sup>7</sup>.</p> <p>(DRG.4.2) The RNG provides forward secrecy.</p> <p>(DRG.4.3) The RNG provides backward secrecy even if the current internal state is known.</p> <p>(DRG.4.4) The RNG provides enhanced forward secrecy <u>on demand</u><sup>8</sup>.</p> <p>(DRG.4.5) The internal state of the RNG is seeded by an <u>PTRNG of class PTG.2</u><sup>9</sup>.</p>
FCS_RNG.1.2/CS/DRG4	The TSF shall provide numbers that meet:

<sup>1</sup> [selection: using a PTRNG of class PTG.2 as random source, using a PTRNG of class PTG.3 as random source, using an NPTNG of class NTG.1 [assignment: other requirements for seeding]]

<sup>2</sup> [selection: have [assignment: amount of entropy], have [assignment: work factor], require [assignment: guess work]]

<sup>3</sup> [assignment: requirements for seeding]

<sup>4</sup> [assignment: number of strings]

<sup>5</sup> [assignment: probability]

<sup>6</sup> [assignment: probability]

<sup>7</sup> [selection: use PTRNG of class PTG.2 as random source, have [assignment: work factor], require [assignment: guess work]]

<sup>8</sup> [selection: on demand, on condition [assignment: condition], after [assignment: time]]

<sup>9</sup> [selection: internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]]

FCS_RNG.1/CS/DRG4	Random number generation
	<p>(DRG.4.6) The RNG generates output for which <math>2^{48}</math> strings of bit length 128 are mutually different with probability <u>that is greater than <math>1 - 2^{(-24)}</math></u><sup>2</sup>.</p> <p>(DRG.4.7) Statistical test suites cannot practically distinguish the random numbers from the output sequences of an ideal RNG. The random numbers must pass test procedure A <u>and the U.S. National Institute of Standards and Technology (NIST) test suite for RNGs used for cryptographic purposes [SP 800-22]</u><sup>3</sup>.</p>

## 6.2 Security assurance requirements

In the following Table 49, the security assurance requirements and compliance rationale for augmented refinements are given.

**Table 49 SAR list and refinements**

SAR	Refinement
ADV_ARC.1	Refined in [PP0084]
ADV_FSP.5	The refinement of ADV_FSP.4 from [PP0084] can also be applied to the assurance level EAL 6 comprising ADV_FSP.5. The assurance component ADV_FSP.4 is extended to ADV_FSP.5 with aspects regarding the level of description. ADV_FSP.5 requires a semi-formal description in addition. The refinement is still valid.
ADV_IMP.2	The refinement of ADV_IMP.1 in [PP0084] requires the evaluator to check for completeness. In case of ADV_IMP.2 the entire implementation representation has to be provided anyhow. A check for completeness is also applicable in case the entire implementation representation is provided.
ADV_INT.3	No refinement
ADV_TDS.5	No refinement
ADV_SPM.1	No refinement
AGD_OPE.1	Refined in [PP0084]
AGD_PRE.1	Refined in [PP0084]
ALC_CMC.5	The refinement of ALC_CMC.4 from [PP0084] details how configuration management has to be also applied to production. This is also applicable for ALC_CMC.5. ALC_CMC.5 is not specifically focused on production.
ALC_CMS.5	The refinement of ALC_CMS.4 from [PP0084] can also be applied to the assurance level EAL 6 comprising ALC_CMS.5. The assurance package ALC_CMS.4 is extended to ALC_CMS.5 with aspects regarding the configuration control system for the TOE. The refinement is still valid.
ALC_DEL.1	Refined in [PP0084]
ALC_DVS.2	Refined in [PP0084]
ALC_FLR.1	No refinement
ALC_LCD.1	No refinement

<sup>1</sup> [selection: internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]]

<sup>2</sup> [assignment: probability]

<sup>3</sup> [assignment: additional test suites]

## Security Requirements (ASE\_REQ)

SAR	Refinement
ALC_TAT.3	No refinement
ASE_CCL.1	No refinement
ASE_ECD.1	No refinement
ASE_INT.1	No refinement
ASE_OBJ.2	No refinement
ASE_REQ.2	No refinement
ASE_SPD.1	No refinement
ASE_TSS.1	No refinement
ATE_COV.3	The refinement of ATE_COV.2 in [PP0084] clarifies how to deal with testing of security mechanisms for physical protection. It further requests the TOE to be tested under different operating conditions. These refinements are also applicable for ATE_COV.3, which requires complete TSFI coverage.
ATE_DPT.3	No refinement
ATE_FUN.2	No refinement
ATE_IND.2	No refinement
AVA_VAN.5	Refined in [PP0084]

### 6.2.1 Security Policy Model (SPM) details

[CC3] requires in ADV\_SPM.1.1D to define the not modelled SFRs.

The rational for the excluded SFRs are as follows:

- SFRs for cryptographic services are not modelled by convention
- SFRs for physical functions cannot be logically modelled
- SFRs for internal functions have no visible logical interface

The developer shall provide a formal security policy model for the SFRs of this ST with the exception of the SFRs from the following table<sup>1</sup>.

**Table 50 SFRs excluded from SPM**

SFR	Reason for exclusion
FCS_RNG.1/*	cryptographic services
FCS_COP.1/*	cryptographic services
FCS_CKM.1/*	cryptographic services
FCS_CKM.4/*	cryptographic services
FPT_TST.1	physical function
FRU_FLT.2	physical function
FPT_FLS.1	physical function
FPT_PHP.3	physical function
FDP_SDC.1	physical function
FDP_ITT.1	internal function
FPT_ITT.1	Internal function

<sup>1</sup> [assignment: list of policies that are formally modelled]

SFR	Reason for exclusion
FDP_IFC.1	Internal function

Note: A star "\*" means all iterations of that SFR

## 6.3 Security requirements rationale

### 6.3.1 Rationale for the Security Functional Requirements

The security requirements rationale identifies the modifications and additions made to the rationale presented in [PP0084].

#### 6.3.1.1 Additional SFRs related to O.Firewall

**Table 51 Rationale for SFRs related to O.Firewall**

SFR	Rationale
FDP_ACC.2/AF	The SFR with the respective SFP require the implementation of an area-based memory access control.
FDP_ACF.1/AF	The SFR allows the TSF to enforce access to objects within the respective SFP based on security attributes and defines these attributes and defines the rules based on these attributes that enable explicit decisions.
FMT_MSA.3/AF	The SFR requires that the TOE provides default values for the security attributes used in the SFP. Because the TOE is a hardware platform, these default values are generated by the reset procedure.
FMT_MSA.1/AF/S	The SFR requires that authorized users can manage TSF attributes. It ensures that the access control attributes associated to secure addresses can be managed only by code running in secure and privilege mode.
FMT_MSA.1/AF/NS	The SFR requires that authorized users can manage TSF attributes. It ensures that the access control attributes associated to non-secure addresses can be managed by code running in secure or non-secure privilege mode.
FMT_SMF.1/AF	The SFR is used for the specification of the management functions to be provided by the TOE. Being a hardware platform, the TOE allows the management of the security attributes by making the hardware registers accessible to software to enable modification.
FMT_SMR.1/AF	This SFR defines the roles used for management of the security attributes. The roles are defined by the security attribute of the fetch address of the CPU instruction.

### 6.3.1.2 Additional SFRs related to O.Ctrl\_Auth\_Loader

**Table 52 Rationale for additional SFRs related to O.Ctrl\_Auth\_Loader**

SFR	Rationale
FMT_MTD.1/Loader	This SFR requires that the TOE provides management functions for modification and deletion of authentication keys.
FMT_SMR.1/Loader	This SFR requires that the roles to management keys are defined
FMT_SMF.1/Loader	This SFR requires that the key management functions are defined
FIA_UID.2/Loader	This SFR requires that management functions can only be executed by authorized roles.

### 6.3.1.3 Additional SFRs related to O.Phys-Manipulation

The FPT\_TST.1 component allows that particular parts of the security mechanisms and functions provided by the TOE can be tested after TOE delivery. This feature is important to detect direct physical manipulations by a FIB device in order to disable the alarm system of the chip.

### 6.3.1.4 Additional SFRs related to O.Hash

The optional PQ Crypto Suite provides Hash functions. The associated SFRs are defined in chapter 6.1.11.1.

### 6.3.1.5 Additional SFRs related to O.MLKEM

The optional PQ Crypto Suite provides MLKEM cryptographic services. The associated SFRs are defined in chapter 6.1.11.2.

### 6.3.1.6 Additional SFRs related to O.RND

The optional PQ Crypto Suite provides random functions. The associated SFRs are defined in chapter 6.1.11.3.

## 6.3.2 Dependencies of Security Functional Requirements

The dependencies of the SFRs which are defined in [PP0084] are resolved in [PP0084], ch. 6.3.2. The following table lists the dependencies of the additional SFRs which are defined in this ST.

**Table 53 Dependencies of SFRs**

SFR	Dependencies	Rationale
FDP_ACC.2/AF	FDP_ACF.1	Fulfilled by FDP_ACF.1/AF
FDP_ACF.1/AF	FDP_ACC.1	Fulfilled by FDP_ACC.2/AF, which is hierarchically higher
FMT_MSA.3/AF	FMT_MSA.1	Fulfilled by FMT_MSA.1/AF/S and FMT_MSA.1/AF/NS
	FMT_SMR.1	Fulfilled by FMT_SMF.1/AF
FMT_MSA.1/AF/S FMT_MSA.1/AF/NS	FDP_ACC.1 or FDP_IFC.1	Fulfilled by FDP_ACC.2/AF, which is hierarchically higher
	FMT_SMR.1	Fulfilled by FMT_SMF.1/AF
	FMT_SMF.1	Fulfilled by FMT_SMF.1/AF
	FMT_SMR.1	Fulfilled by FMT_SMF.1/AF



## Security Requirements (ASE\_REQ)

SFR	Dependencies	Rationale
	FMT_SMF.1	Fulfilled by FMT_SMF.1/AF
FMT_SMR.1/AF	FIA_UID.1	The dependency is satisfied, because the role is identified by the execution context of the processor.
FMT_SMF.1/AF	None	No dependency
FDP_ACC.1/Loader	FDP_ACF.1	Fulfilled by FDP_ACF.1/Loader
FDP_ACF.1/Loader	FMT_MSA.3	Not applicable, because there are no security attributes defined
FMT_MTD.1/Loader	FMT_SMR.1	Fulfilled by FMT_SMR.1/Loader
	FMT_SMF.1	Fulfilled by FMT_SMF.1/Loader
FMT_SMR.1/Loader	FIA_UID.1	Fulfilled by FIA_UID.2/Loader
FMT_SMF.1/Loader	None	No dependency
FIA_UID.2/Loader	None	No dependency
FPT_TST.1	None	No dependency
FCS_COP.1/AES	FCS_CKM.4	Fulfilled by FCS_CKM.4
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	The TOE does not provide services to generate symmetric keys. This will be done by the embedded software for the composite TOE.
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	The TOE does not provide services to generate or import symmetric keys. This will be done by the embedded software for the composite TOE.
FCS_COP.1/CS/MLKEM/<iter>	FCS_CKM.4	The TOE does not provide services to destroy MLKEM keys. This will be done by the embedded software for the composite TOE.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	Fulfilled by FCS_CKM.1/CS/MLKEM/<iter>
FCS_CKM.1/CS/MLKEM/<iter>	FCS_CKM.2 or FCS_COP.1]	Fulfilled by FCS_COP.1/CS/MLKEM/<iter>
	FCS_CKM.4	The TOE does not provide services to destroy MLKEM keys. This will be done by the embedded software for the composite TOE.
FCS_COP.1/CS/Hash/<iter>	FCS_CKM.4	n/A, as a hash function does not use keys.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	n/A, as a hash function does not use keys.
FCS_RNG.1/CS/PTG2	None	No dependency
FCS_RNG.1/CS/PTG3	None	No dependency
FCS_RNG.1/CS/DRG3	None	No dependency
FCS_RNG.1/CS/DRG4	None	No dependency

### 6.3.3 Rationale of the Assurance Requirements

The TOE is a typical security IC as defined in [PP0084]. The rationale for EAL level and augmentation is as follows.

---

**Security Requirements (ASE\_REQ)**

An assurance level EAL6 with the augmentations ALC\_FLR.1 is required for this type of TOE since it is intended to defend against highly sophisticated attacks without a protective environment. This evaluation assurance package was selected to permit a developer to gain maximum assurance from positive security engineering based on good commercial practices. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defence against such attacks, the evaluators should have access to all information regarding the TOE including the TSF internals, the low level design and source code including the testing of the modular design. Additionally the mandatory technical document [JIL] shall be taken as a basis for the vulnerability analysis of the TOE.

## 7 TOE Summary Specification (ASE\_TSS)

The product overview is given in section 1.3.1. The Security Features are described below and the relation to the security functional requirements is shown. The TOE is equipped with the following security features to meet the security functional requirements:

**Table 54 TOE Security Features**

Security Feature	Description
SF_DPM	Device Phase Management
SF_PS	Protection against Snooping
SF_PMA	Protection against Modification Attacks
SF_PLA	Protection against Logical Attacks
SF_HC	Hardware provided Cryptography
SF_CS	PQ Crypto Suite Services

### 7.1 SF\_DPM: Device Phase Management

The life cycle of the TOE is split up into several phases (see [PP0084], ch. 1.2.3). Chip development and production (phase 2, 3, 4) and final use (phases 4-7) is a rough split-up from the TOE point of view. These phases are implemented in the TOE as test mode (phase 3) and user mode (phases 4-7). In addition a chip identification mode exists which is active in all phases. The chip identification data (O.Identification) is stored in a non-modifiable configuration page area of the non-volatile memory. Further TOE configuration data is stored in the same area. In addition, user initialization data can be stored in the NVM during the production phase as well. During this first data programming, the TOE is still in the secured environment and in test mode.

The covered security functional requirement is FAU\_SAS.1 “Audit storage”.

During start-up of the TOE the decision for one of the various operation modes is taken dependent on phase identifiers. The decision of accessing a certain mode is defined as phase entry protection. The phases follow also a defined and protected sequence. The sequence of the phases is protected by means of authentication.

The covered security functional requirements are FMT\_LIM.1 and FMT\_LIM.2.

During the production phase (phase 3 and 4) or after the delivery to the customer (phase 5 or phase 6), the TOE provides the possibility to download a user specific encryption key and user code and data into the empty (erased) NVM area as specified by the associated control information of the Flash Loader software. Alternatively in case the user has ordered TOE derivatives without Flash Loader, software download by the user (phase 5 or phase 6) is disabled and all user data of the embedded software is stored on the TOE at Infineon premises. In case the user has ordered the TOE derivatives with Flash Loader enabled, the Flash Loader may either be received in a way, which requires an authentic Pinletter and authentication afterwards, or it may be received in a state, which immediately requires successful mutual authentication. The Pinletter process can exchange the default authentication key. Successful authentication is required before being able to use the download functionality of the Flash Loader. Once authenticated, the functionality to exchange the Flash Loader keys depending on the user’s identity is enabled. One of the keys, which can be exchanged is the Image Provider key. This key is used to decrypt and verify the integrity protected and encrypted download image. The authenticated user may also invalidate authentication keys depending on the user’s identity. After finishing the download operation, the Flash Loader has to be permanently deactivated prior delivery to the end user, so that no further load operation with the Flash Loader is possible. The Flash Loader uses AES CCM mode [SP 800-38C] for encryption and integrity protection of payload and for authentication. For key usage diversification, the Flash Loader uses key derivation according to [SP 800-108].

**TOE Summary Specification (ASE\_TSS)**

The covered security functional requirements are FMT\_LIM.1/Loader, FMT\_LIM.2/Loader, FTP\_ITC.1, FDP\_UCT.1, FDP\_UIT.1, FDP\_ACC.1/Loader, FDP\_ACF.1/Loader, FMT\_MTD.1/Loader, FMT\_SMR.1/Loader, FMT\_SMF.1/Loader, FIA\_UID.2/Loader and FIA\_API.1.

Note that the SFRs FTP\_ITC.1, FDP\_UCT.1, FDP\_UIT.1, FDP\_ACC.1/Loader, FDP\_ACF.1/Loader, FMT\_MTD.1/Loader, FMT\_SMR.1/Loader, FMT\_SMF.1/Loader, FIA\_UID.2/Loader and FIA\_API.1 are only part of the TOE if the flash loader is active.

Each operation phase is protected by means of authentication and encryption.

The covered security functional requirements are FDP\_ITT.1 and FPT\_ITT.1.

## 7.2 SF\_PS: Protection against Snooping

All contents of the memories RAM, ROM and NVM of the TOE are encrypted on chip to protect them against data analysis. The encryption of the memory content is done by the MCICE using a proprietary cryptographic algorithm. A complex key management and address scrambling provides protection against cryptographic analysis attacks. All security relevant transfers via the peripheral bus are dynamically masked and thus protected against readout and analysis. Leakage of data dependent code execution can be reduced by employing specific hardware features.

In addition, the Masked Instruction Set Extension (MISE) coprocessor provides an Armv8-M Custom Data path Extension (CDE) with side-channel improved (masked) variants of common 32-bit instructions. This will provide additional means for the embedded software to minimize side channel leakage.

The covered security functional requirements are FDP\_SDC.1, FDP\_IFC.1, FPT\_PHP.3, FPT\_ITT.1, FPT\_FLS.1 and FDP\_ITT.1.

Most components of the design are synthesized to disguise allocation of elements to certain modules of the IC. Physical regularity of the logic functions is thereby removed. The covered security functional requirement is FPT\_PHP.3.

A further protective design method used is security optimized wiring. Certain security-critical wires have been identified and protected by special routing measures against probing. Additionally specific signal lines, required to operate the device, are embedded into shield lines of the chip to prevent successful probing. The covered security functional requirements are FPT\_PHP.3, FPT\_ITT.1, FDP\_ITT.1 and FDP\_IFC.1.

A low system frequency sensor FSE is implemented to prevent the TOE from single stepping. The sensor is tested by the User Mode Security Life Control UMSLC. The UMSLC library provides some wrapper functionality around the UMSLC hardware part containing measures against fault attacks. The covered security functional requirements are FPT\_PHP.3 and FPT\_FLS.1.

## 7.3 SF\_PMA: Protection against Modifying Attacks

The TOE has implemented a dual CPU running in lockstep mode and registers protected with 32 bit EDC. This mechanism reliably detects attacks on the code flow and data processed by the CPU. In the case of a detected attack, the TOE enters the secure state.

The TOE is equipped with a 28 bit EDC in RAM, a 28 bit EDC in NVM and a 32 bit EDC in ROM, which is realized in the MCICE peripheral. The EDC detects detect single- and multi-bit errors. In the case of an EDC error, the TOE enters the secure state.

The covered security functional requirements are FRU\_FLT.2, FPT\_PHP.3 and FDP\_SDI.2.

A life test on internal security features is provided – it is called User Mode Security Life Control (UMSLC), which checks alarm lines for correct operation. This test can be triggered by user software during normal operation or via the UMSLC lib. If physical manipulation or a physical probing attack is detected, the TOE enters the secure

state (as defined in chapter 6.1.4). To further decrease the risk of manipulation and tampering of the detection system a redundant alarm propagation and system deactivation is provided.

The covered security functional requirements are FPT\_FLS.1, FPT\_PHP.3 and FPT\_TST.1

The Instruction Stream Signature Checking (ISS) calculates a hash over all executed instructions and automatically checks the correctness of this hash value. If the code execution follows an illegal path an alarm is triggered. This feature can optionally be used for program flow integrity protection but it is not needed as the dual CPU and memory EDC mechanisms are far better suited to detect such attacks.

The Online Configuration Check (OCC) function controls the modification of relevant system settings. It is also useful as a measure against fault attacks and accidental changes. The content of the protected registers is permanently hashed and checked against a reference value. A violation generates an alarm event and leads to the secure state.

The TOE supports dynamical locking of dedicated peripherals. This way data flow between CPU and peripherals can be controlled. Manipulations utilizing access to specific peripherals can be prevented with this locking mechanism.

As physical effects or manipulative attacks may also target the program flow of the user software, a watchdog timer and a check point register are implemented. These features allow the user to check the correct processing time and the integrity of the program flow of the user software.

The covered security functional requirements are FPT\_FLS.1, FDP\_IFC.1, FPT\_ITT.1, FDP\_ITT.1 and FPT\_PHP.3.

The HSL provides tearing safe write operations which can be utilized by the embedded software.

The covered security functional requirement is FPT\_PHP.3.

The correct function of the TOE is only given in the specified range of environmental operating parameters. To prevent an attack exploiting that circumstance the TOE is equipped with a temperature sensor, glitch sensor and voltage sensor as well as backside light detection. The TOE falls into the defined secure state in case of a specified range violation. The defined secure state causes the chip internal reset process.

The covered security functional requirements are FRU\_FLT.2 “Limited fault tolerance” and FPT\_FLS.1 “Failure with preservation of secure state”.

## 7.4 SF\_PLA: Protection against Logical Attacks

The TOE implements the Armv8-M Memory Protection Unit (MPU) with 8 regions and the Security Attribution Unit (SAU) with 8 regions according to [Armv8-M], ch. B10.

The SAU contains an Implementation Defined Attribution Unit (IDAU). The IDAU exempts the address ranges 4000 0000H - 5FFF FFFFH and A000 0000H - FFFF FFFFH from Security attribution.

During each start-up of the TOE the address ranges and MPU access rights are initialized by the Boot Software (BOS) with predefined values. The BOS maps a small region containing the start-up code for access of privilege software.

The SAU is disabled and all addresses are marked secure and non-secure not callable.

The covered security functional requirements are FDP\_ACC.2/AF, FDP\_ACF.1/AF, FMT\_MSA.1/AF/S, FMT\_MSA.1/AF/NS, FMT\_MSA.3/AF, FMT\_SMF.1/AF and FMT\_SMR.1/AF.

## 7.5 SF\_HC: Hardware provided cryptography

The TOE is equipped with a random number generator as defined in the SFRs FCS\_RNG.1/TRNG in chapter 6.1.1.

The covered security functional requirement is FCS\_RNG.1/TRNG.

The TOE supports the encryption and decryption in accordance with the Advanced Encryption Standard (AES) and cryptographic key sizes of 128 bits or 192 bits or 256 bits that meet the standards as defined in chapter 6.1.2.

The covered security functional requirement are FCS\_COP.1/AES and FCS\_CKM.4.

## 7.6 SF\_CS: PQ Crypto Suite Services

The optional PQ Crypto Suite utilizes MISE operations to implement MLKEM cryptographic algorithms. The associated SFRs are listed in chapter 6.1.11.

## 8 Hash values of libraries

This chapter list the SHA256 hashes of the libraries from section 1.4.2.2.

**Table 55** SHA256 hash values

Lib	SHA256
UMSLC	74091c50254bc348a48a2e261a125735dca41f2419cd9541c3e6fbfdff63b529
HSL	5357585cff662d4dd45766bd682303ee31f66ebc41998a489b0124cef9b87e55
PQ Crypto Suite	CS-SLC21V24-core.lib: SHA256=2e651bd6001ca4032134489b2182fa11e3970b97c6188a56a1ba139e118b92fe CS-SLC21V24-rng-drbg.lib: SHA256=9bb6c0f1b12c94038739e1d73b312fe3b5a040e65be00a9a0069fac8939fc269 CS-SLC21V24-sym-hash.lib: SHA256=7aa50c893c2a514628b56639b65e3a560f48fdc34b7befd54756a2a40024a6f8 CS-SLC21V24-pqc-lattice.lib: SHA256=3e28d11f110843f806f209c64343273609d2b298a0867d19bba8c1c65066ae4c CS-SLC21V24-pqc-lattice-mlkem.lib: SHA256=900a9bb506e63b9f5734b3863f1ebed358c36fcd6809413e6702bf85c7686967 CS-SLC21V24-pqc.lib: SHA256=4692ab58466d1b055b9005ba3b3a2ed9a51867b936509effcaff6eb4f3b78a77 CS-SLC21V24-rng.lib: SHA256=608823c53fb3f962932a669c45781e32be4a8805866a5ceb0fbd081766713ec1 CS-SLC21V24-sym-hash-sha3.lib: SHA256=bbdd5eb4f9bf982b09d5c2cbfaff898a4a96b1688417097a624b34710221352c CS-SLC21V24-sym-hash-sha3-shake.lib: SHA256=52c23469f316f71fc945a55da375fb549da6c0c19365f10b1693ac584c73d4e0 CS-SLC21V24-sym.lib: SHA256=e767222b3bd3232d34e6ce76b7b3364402b170fdfa58ed628ed1687a0ccee56

## 9 Cryptographic Table

**Table 56 Cryptographic table**

Purpose	Cryptographic operation	Key size in bits	Standards
Confidentiality	AES in ECB mode provided by hardware	128, 192, 256	[FIPS 197] [SP 800-38A]
Authenticated encryption	AES in CCM mode used by the Flash Loader	128	[FIPS 197] [SP 800-38C]
Integrity	AES CMAC mode used by the Flash Loader	128	[FIPS 197] [SP 800-38B]
Hash	SHA-3 Hash digest 224, 256, 384, 512 bits	N/A	[FIPS 202]
Hash	SHAKE-128, SHAKE-256	N/A	[FIPS 202]
Random	Physical RNG PTG.2	N/A	[AIS 31]
Random	Hybrid Physical RNG PTG.3	AES128 AES256	[AIS 31] [SP 800-90A], ch. 10.2
Random	Deterministic RNG DRG.3	AES128 AES256	[AIS 20], [SP 800-90A], ch. 10.2
Random	Hybrid Deterministic RNG DRG.4	AES128 AES256	[AIS 31] [SP 800-90A], ch. 10.2
Key generation	MLKEM	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203]
Key encapsulation	MLKEM	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203]
Key decapsulation	MLKEM	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203]
XOF	SHAKE-128, SHAKE-256	N/A	[FIPS 202]
Key derivation	KDF in counter mode with AES CMAC as PRF used by Flash Loader	128	[SP 800-108], ch. 4.1 [SP 800-38B], ch. 6.2



## Acronyms

## Acronyms

Acronym	Description
AES	Advanced Encryption Standard
CSP	Chip Scale Package
ECC	Elliptic Curve Cryptography or Error Correction Code
EDC	Error Detection Code
FFC	Finite Field Cryptography
ISS	Instruction Stream Signature
MISE	Masked Instruction Set Extension
MCICE	Memory Cache Confidentiality Integrity Engine
MLKEM	Module Lattice Key Encapsulation Mechanism
MPU	Memory Protection Unit
NVIC	Nested Vectored Interrupt Controller
NVM	Non-Volatile Memory
OCC	Online Configuration Check
RAM	Random Access Memory
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest Shamir Adleman
SAU	Security Attribution Unit
SE	Security Extension
SPI	Serial Peripheral Interface
SPM	Security Policy Model
SWP	Single Wire Protocol
TOE	Target Of Evaluation
UMSLC	User Mode Security Life Control

## References

[AIS 20]	Anwendungshinweise und Interpretationen zum Schema AIS 20, Version 3, 15.05.2013 Bundesamt für Sicherheit in der Informationstechnik
[AIS 31]	Anwendungshinweise und Interpretationen zum Schema AIS 31, Version 3, 15.05.2013 Bundesamt für Sicherheit in der Informationstechnik
[AIS 46]	Anwendungshinweise und Interpretationen zum Schema AIS 46, Version 3, 2013-12-04 Bundesamt für Sicherheit in der Informationstechnik
[Armv8-M]	Arm® v8-M Architecture Reference Manual DDI0553A.m

## Acronyms

[CC1]	Common Criteria for Information Technology Security Evaluation Part12: Security Functional Requirements; Version 3.1 Revision 5 April 2017, CCMB-2017-04-001
[CC2]	Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements; Version 3.1 Revision 5 April 2017, CCMB-2017-04-002
[CC3]	Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements; Version 3.1 Revision 5 April 2017, CCMB-2017-04-003
[FIPS 197]	Federal Information Processing Standards Publication, U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL), Advanced Encryption Standard (AES), FIPS PUB 197, as of 05/09/23
[FIPS 202]	Federal Information Processing Standards Publication, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions, August 2015
[FIPS 203]	Federal Information Processing Standards Publication, FIPS 203 August 13, 2024
[ISO9798_2]	ISO/IEC 9798-2: 2008 - Information Technology - Security techniques - Entity authentication - Part 2: Mechanisms using symmetric encipherment algorithms
[JIL]	Joint Interpretation Library, Application of Attack Potential to Smartcards, Version 3.2.1 February 2024
[PP0084]	Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, 13.01.2014, BSI-CC-PP-0084-2014
[SP 800-108]	National Institute of Standards and Technology (NIST), Recommendation for Key Derivation Using Pseudorandom Functions, NIST SP 800-108r1, August 2022
[SP 800-22]	National Institute of Standards and Technology(NIST), Technology Administration, US Department of Commerce, Special Publication 800-22, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, in the revision 1a as of April 2010
[SP 800-38A]	National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard, NIST Special Publication 800-38A, Edition 2001
[SP 800-38B]	National Institute of Standards and Technology (NIST), Special Publication 800-38B, May 2005
[SP 800-38C]	NIST SP 800-38C: Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, 2004-05 (up-dated: 2007-07-20)
[SP 800-90A]	NIST Special Publication 800-90A Revision 1 Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015



**Acronyms**

**Revision history**

<b>Version</b>	<b>Date</b>	<b>Description</b>
Rev. 1.2	2024-10-14	Released Version

#### **Trademarks**

All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2024-10-14**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2024 Infineon Technologies AG.**

**All Rights Reserved.**

**Do you have a question about this document?**

**Email:**

[dsscusterservice@infineon.com](mailto:dsscusterservice@infineon.com)

#### **IMPORTANT NOTICE**

The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on the product, technology delivery terms and conditions and prices please contact your nearest Infineon Technologies office ([www.infineon.com](http://www.infineon.com)).

#### **WARNINGS**

Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.