IFX_CCI_00007Ah/8Fh A11/R11/M11 with optional Crypto Suite

Security Target Lite

Release

About this document

Scope and purpose

This document is the Security Target for the Infineon IFX_CCI_00007Ah/8Fh A11/R11/M11 security controllers.

Intended audience

Composite product developers, Common Criteria Evaluators and Certifiers.

Table of contents

Table of contents

List of figures

List of tables

List of tables

1 Introduction (ASE_INT)

1.1 ST reference

The ST has the title IFX_CCI_00007Ah/8Fh A11/R11/M11 with optional Crypto Suite Security Target Lite, Rev 1.0 and is dated 2025-10-31.

1.2 TOE reference

The full TOE name is:

IFX_CCI_00007Ah/8Fh A11/R11/M11 with firmware version 80.510.04.03, optional Crypto Suite 5.02.003 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 of 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 Table 3 in 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
- Redundant alarm propagation and system deactivation principle
- Peripheral access control
- Leakage control of data dependent code execution
- Device phase management
- The optional Crypto Suite provides hardened cryptographic functions for AES, TDES, RSA, ECC, finite field DH, Hash functions, SHAKE XOF, MLKEM, MLDSA and deterministic random number generators.
- 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.

1.3.3 Self Secured Digital Protection (SDP)

Self-secured digital protection (SDP) covers protection of a TOE subsystem against modification attacks. This protection is directly provided by hardware. No support from the embedded software is necessary.

This TOE provides an SDP subsystem consisting of the green marked components of Figure 1.

Digital error detection is key in providing robustness to protect against modification attacks, as physical countermeasures may degrade over time due to improved fault attack equipment in the future. For instance, a light sensor grid blocking a state-of-the-art laser may be bypassed by a future laser with smaller spot size.

Verification of the functionality of the SDP subsystem is difficult by using the existing assurance classes in [PP0084] only. This is especially related to the AVA_VAN class of physical attacks. In most (physical) attack scenarios the error detection mechanism will not be triggered because the sensors or filters have already detected and blocked the attack. Therefore, new test methods are defined in this ST which will exhaustively cover the digital error detection functionality.

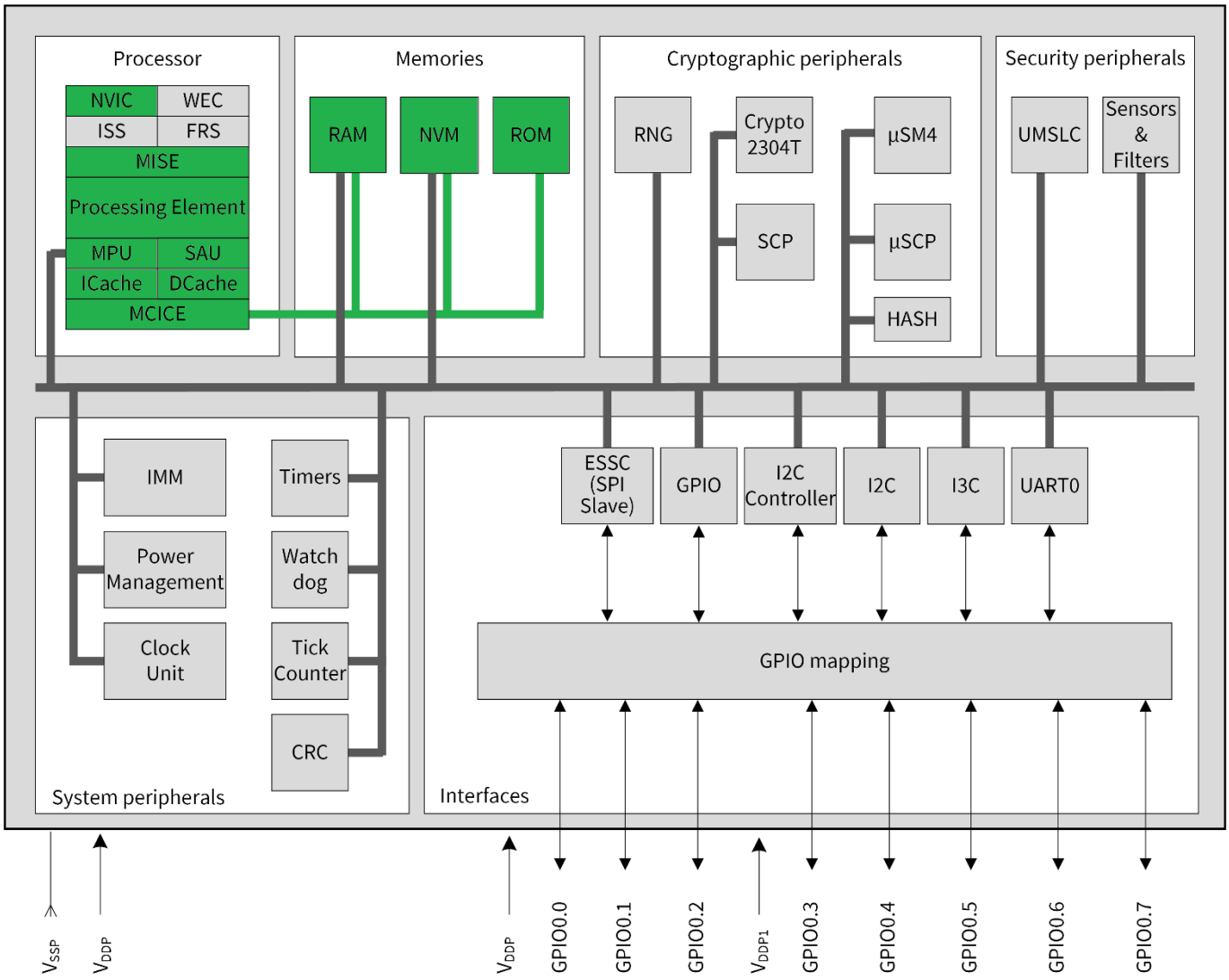
1.4 TOE description

1.4.1 TOE components

1.4.1.1 TOE hardware

Figure 1 shows schematically the TOE hardware. The green-coloured blocks define the SDP subsystem.

Figure 1 TOE hardware



The TOE hardware consists of the following blocks:

- Processor
  - CPU according to Armv8-M mainline architecture.
  - 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
  - Wakeup Event Controller (WEC)
- 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)
  - Memory Peripheral Access Unit (MPAU) with system MPU
- Cryptographic peripherals
  - RNG according to class PTG.2 of [AIS 31]
  - Crypto2304T coprocessor for long modular integer arithmetic
  - SCP for secure AES computation in hardware
  - µSCP for secure AES and TDES support in hardware
  - Hash module for fast MD5, SHA-1 and SHA-2 hash computation in hardware
- Security peripherals
  - UMSLC
  - Sensors and filters
- I/O Interfaces
  - UART for ISO 7816-3
  - SPI-compatible interface
  - I2C device
  - I3C device
  - 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. The Flash Loader allows downloading of User Software into the NVM during the manufacturing process and in field during phase 7 of the TOE life cycle.

1.4.1.3 Libraries

The TOE can be ordered with the following 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
- Crypto Suite library to provide certified cryptographic functions.

1.4.2 Physical scope

1.4.2.1 Hardware/Firmware

Table 1 Hardware/Firmware components

Component	Version
Hardware	IFX_CCI_00007Ah A11 IFX_CCI_00007Ah M11 IFX_CCI_00008Fh M11 IFX_CCI_00007Ah R11 IFX_CCI_00008Fh R11
Firmware	80.510.04.03
Flash Loader	10.09.0000

1.4.2.2 Libraries

The following libraries are part of the TOE.

Table 2 Libraries

Component	Version
HSL	04.07.0000
UMSLC	02.01.0040
Crypto Suite	5.02.003

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

1.4.2.3 User guidance documents

Table 3 User guidance

Component	Version	Date
TEGRION™ SLC22 (32-bit Security Controller – V31) Hardware Reference Manual	3.1	2025-04-22
TEGRION™ SLx2 security controller family Programmer's Reference Manual SLx2_DFP	1.9.0	2025-10-16
SLC22 32-bit Security Controller – V31 Security Guidelines	1.00-3076	2025-04-30
TEGRION SLC22 (32-bit Security Controller – V31) Preliminary Production and personalization manual	10.09	2025-05-12
Crypto2304T V4, User Manual	3.0	2024-06-21
CS-SLC22V31 CryptoSuite	5.02.003	2025-10-02

Component	Version	Date
32-bit Security Controller User interface manual		
TEGRION™ SLx22 (32-bit Security Controller – V31) Errata sheet	Revision 2.0	2025-06-27

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.

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 encryption.
- The data buses connecting the CPU and the cryptographic peripherals are confidentiality protected using a dynamic bus mask which is 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.
- 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 Crypto Suite supporting
  - RSA key generation, encryption and signature up to 4224 bits
  - ECDSA and ECDH up to 521 bits
  - PACE Integrated mapping

  - Brainpool curves, NIST curves, W-25519, Koblitz secp256k1, BN P256, ANSII FRP256V1
  - X25519, X448, EdDSA
  - Finite Field DH up to 4096 bits
  - AES 128, 192,256 in different chaining. MAC and authenticated encryption modes
  - TDES 112, 168 in different chaining and MAC modes
  - SHA-1, SHA-2, SHA-3 Hash and SHAKE XOF
  - HMAC with SHA-1 and SHA-2
  - Module lattice based Post Quantum cryptography
  - RNG functions supporting PTG.2, PTG.3, DRG.3, DRG.4 according to [AIS 20] and [AIS 31]
- 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. Appropriate countermeasures are provided by the optional Crypto Suite library.

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 on wafer and sawn	3	Customer chooses delivery method. This ST describes under which circumstances transport protection is provided by the TOE.
	Modules or IC case	4	Customer chooses delivery method. This ST describes under which circumstances transport protection is provided by the TOE.
Firmware	binary image	3 or 4[1]	In ROM/NVM of hardware
Libraries	object files	n/A	secure download via iShare
Documents	personalized PDF	n/A	secure download via iShare

[1] 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	1800, 1192, 1512 kB	IFX mailbox
RAM	48, 64, 96 kB	IFX mailbox
SPI TPM Extension	Available / not available	Hardware register

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] extended

[CCErrata] and [CCTrans] are taken into consideration.

2.1.1 PP claims

This ST is strictly conformant to [PP0084]. The assurance level is EAL6 with the augmentation ALC_FLR.1 and ATE_SDP.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 only available, if the Flash Loader is available and mutual authentication of the Flash Loader with the roles Administrator User and Download Operator User is not deactivated.
- Package Loader, Package 1: Loader dedicated for usage in secured environment only, section 7.3.1, conformant. This package is only applicable if a Flash Loader is available.
- Package Loader, Package 2: Loader dedicated for usage by authorized users only, section 7.3.2, augmented. This package only available, if the Flash Loader is available and mutual authentication of the Flash Loader with the roles Administrator User and Download Operator User is not deactivated.
- Package TDES; section 7.4.1, augmented. This package is only available if the optional Crypto Suite is available.
- Package AES; section 7.4.2, augmented.

The assurance level for the TOE is EAL6 augmented with the component ALC_FLR.1 and ATE_SDP.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].

2.2.1 CC:2022 conformance

With CC:2022 several SFR changes are introduced. Due to this ST claiming conformance to CC:2022 and [PP0084], rationales are provided that these changes do not affect the conformance claim to [PP0084]:

- FCS_COP.1: for this SFR dependencies are changed in CC:2022. FCS_CKM.4 is removed and instead FCS_CKM.6 added. FCS_CKM.5 is formally added but is optional and not used in this ST.
- FCS_CKM.1: for this SFR dependencies are changed in CC:2022. Additionally, to FCS_CKM.2 and FCS_COP.1, one further SFR is introduced as alternative: FCS_CKM.5. This SFR targets key derivation after FCS_CKM.1. In CC:2022 key derivation would have been part of FCS_CKM.1 and thus conformance to [PP0084] can still

be claimed. All other dependencies are in addition to the already existing ones, i.e. add stricter requirements.

- FCS_CKM.6 replaces FCS_CKM.4 and adds further requirements on the timing of key destruction.
- FCS_RNG.1: this SFR is taken from [CC2] rather than [PP0084]. The SFR is identical in [CC2]
- FMT_LIM.1 and FMT_LIM.2 are taken from [CC2] rather than [PP0084]. There is a slightly different phrasing (i.e. removing redundancy from FMT_LIM.1) and availability and capability policy mentioned in both SFR's. The meaning though is the same and therefore conformance can still be claimed.
- FIA_API.1: this SFR is taken from [CC2] rather than [PP0084]. The SFR requires additional information.
- FDP_SDC.1: this SFR is taken from [CC2] rather than [PP0084]; An assignment from [PP0084] is changed to a selection [CC2], which means the requirement is more stringent and thus can be considered a subset of the requirement from [PP0084].

Further with CC:2022 some SAR changes were introduced. Rationales are provided that these changes do not affect the conformance claim to [PP0084]:

- ASE_CCL.1: for CC:2022 several extensions were introduced (e.g. exact conformance to PP), which add to the already existing assurance requirements. No relaxation was introduced.
- ASE_INT.1: introduction of multi-assurance in combination with PP-configuration: not relevant for [PP0084]
- ASE_REQ.2: extended for multi assurance: not relevant for [PP0084]
- AVA_VAN.5: extension about third party components introduced. No relaxation was introduced.
- ALC_TAT.1: extension with guidance on the minimum content for an implementation standards description and rules with ADV_COMP.1. No relaxation was introduced.

3 Security Problem Definition (ASE_SPD)

3.1 Threats

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

Table 8 Threats defined in this ST

Threats	Description
T.Open_Samples_Diffusion	**Diffusion of Open Samples** An attacker may get access to open samples of the TOE and use them to gain information about the TSF (loader, memory management unit, ROM code …). He may also use the open samples to characterize the behavior of the IC and its security functionalities (for example: characterization of side channel profiles, perturbation cartography …). The execution of a dedicated security features (for example: execution of a AES computation without countermeasures or by de-activating countermeasures) through the loading of an adequate code would allow this kind of characterization and the execution of enhanced attacks on the IC.
T.Load_Non_Authentic	**Loading of an unauthentic image** An attacker may try to load a manipulated image or reload an outdated and less secure image onto the TOE during phase 7.
T.Corrupt_Image_Load	**Corrupting an image by non atomic update** An attacker may try to partially load an image e.g. by a tearing attack in order to provoke establishing and executing a non authentic image during phase 7.

Rationale for adding threats in this ST

The availability of the flash loader means that an attacker may be able to execute dedicated security features without countermeasures through the loading of adequate code. This will allow him to characterize and execute enhanced attacks. Due to this, additional objectives for secure loading and activation of the additional code as well as protection of confidentiality of TSF are required and the threats T.Open_Sample_Diffusion, T.Load_Non_Authentic and T.Corrupt_Image_Load are introduced.

3.2 Organizational security policies

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

Table 9 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

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

Table 10 Organizational 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

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

Table 11 Assumptions from [PP0084]

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

There are no additional assumptions defined in this ST.

4 Security Objectives (ASE_OBJ)

4.1 Security objectives for the TOE

Table 12 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 (refined, see below)
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
O.TDES	Cryptographic service Triple-DES

Note: The objectives O.TDES is only applicable if the optional Crypto Suite is ordered

Table 13 Security objectives for the TOE from [PDCL]

Objective	Description
O.Secure_UC_Load	**Secure loading of the Update Code** The Loader of the Initial TOE shall check an evidence of authenticity and integrity of the loaded Update Code. The Loader enforces that only the allowed version of the Update Code can be loaded on the Initial TOE. The Loader shall forbid the loading of an Update Code not intended to be assembled with the Initial TOE. During the Load Phase of an Update Code, the TOE shall remain secure.
O.Secure_UC_Activation	**Secure activation of the Update Code** Activation of the Update Code and update of the Identification Data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the Updated TOE shall be completed before activation. If the Atomic Activation is successful, then the resulting product is the Updated TOE, otherwise (in case of interruption or incident which prevents the forming of the Updated TOE such as tearing, integrity violation, error case…), the Initial TOE shall remain in its initial state or fail secure.

The security objectives O.Secure_UC_Load, O.Secure_UC_Activation are only applicable if the Flash Loader is active.

The security objectives O.Authentication and O.Ctrl_Auth_Loader are only applicable if the Flash Loader is active and mutual authentication is not disabled.

Table 14 Security Objectives for the TOE defined in this ST

Objective	Definition
O.Firewall	**Firewall based Access Control** 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.
O.FFC	**Finite Field cryptographic services** The TOE shall provide the following specific security functionality to the Smartcard Embedded Software: Finite Field cryptographic services
O.RSA	**RSA cryptographic services** The TOE shall provide the following specific security functionality to the Smartcard Embedded Software: Rivest-Shamir-Adleman Cryptography (RSA)
O.ECC	**Elliptic Curve cryptographic services** The TOE shall provide the following specific security functionality to the Smartcard Embedded Software: Elliptic Curve Cryptography (ECC)
O.ML	**Module Lattice cryptographic services** The TOE shall provide the following specific security functionality to the Smartcard Embedded Software: Module Lattice based Post Quantum algorithms
O.HMAC	**HMAC Cryptographic services** The TOE provides secure cryptographic services for HMAC calculation.
O.Hash	**Cryptographic service hash** The TOE provides secure cryptographic services for Hash generation and Extended Output Functions.
O.Prot_TSF_Confidentiality	**Protection of confidentiality of TSF** The TOE must provide protection against disclosure of confidential operations of the security IC (loader, memory management unit, …) through the use of a dedicated code loaded on open samples.

Note: The objectives O.FFC, O.RSA,O.ECC, O.ML, O.HMAC and O.Hash are only applicable if the optional Crypto Suite is ordered

4.2 Security objectives for the operational environment (OE)

Table 15 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

Objective	Description
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.

Table 16 Security objectives for the operational environment defined in this ST

Objective	Description
OE.Secure_UC_Load	User software is required to support secure image loading in phase 7 by providing the relevant key material for the image. This OE is only applicable if the Flash Loader is active.
OE.Secure_Delivery	When the TOE is ordered with a disabled Flash Loader or with a enabled Flash Loader but Mutual Authentication disabled, it does not provide transport protection. Therefore, technical and / or organizational 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 objective O.Firewall added in this ST covers 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.FFC, O.RSA, O.ECC, O.HMAC, O.Hash, O.ML cover the policy P.Crypto-Service. This policy intends to allow adding various cryptographic services to the TSF.

The objective O.Prot_TSF_Confidentiality counters the threat T.Open_Samples_Diffusion. In addition, T.Open_Samples_Diffusion is countered by O.Leak-Inherent and O.Leak-Forced. The objectives O.Secure_UC_Load and OE.Secure_UC_Load cover the threat T.Load_Non_Authentic, because O.Secure_UC_Load requires only authentic images to be loaded and OE.Secure_UC_Load requires the relevant key material to be provided by the user OS. The objective O.Secure_UC_Activation requests atomic image loading with fail secure in case final correct image cannot be activated. This counters T.Corrupt_Image_Load, which attempts to distort atomic image loading and activation.

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.Open_Samples_Diffusion and T.Masquerade_TOE.

5 Extended Components Definition (ASE_ECD)

5.1 FAU_SAS.1 - Audit storage

This extended component is defined in [PP0084].

5.2 FPT_SDP - Self-secured digital protection

5.2.1 Family behaviour

This security functional family extending class FPT provides requirements in case TOE hardware components contain self-protection features in terms of error detection logic, capable of preserving a secure state under certain fault induced errors. This family provides an additional layer of self-protection next to physical countermeasures like sensors, filters and other physical means as required by FPT_PHP and FPT_FLS.1. It also complements FDP_SDI by explicitly requiring strong error detection capabilities in protected memory.

It is important that the SDP subsystem detects bit errors by *digital error detection* logic. This augments analogue fault detection methods by sensors and filters. It defines a second layer of defence which comes into play in case the fault attack has managed to bypass the sensors or filters. Having such a strong second layer of defense in hardware makes additional countermeasures by the embedded software unnecessary.

It is not necessary that the complete TOE is SDP protected. Therefore, the TOE can be divided into the SDP protected subsystem and the non-SDP protected subsystem (see Figure 2). The SDP subsystem must contain at least the CPU and the protected memories of the TOE and may optionally contain protected peripherals. Protected peripherals can be CPU instruction set extensions or private peripherals directly attached to the CPU or devices attached to the system bus.

The error detection logic in the CPU and the protected peripherals must enforce detection of single bit errors in sequential elements (flip-flops and latches), where single bit error means that one bit error can occur on one arbitrary sequential cell at one arbitrary clock cycle. Detection of single bit errors is not considered as sufficient for the highly regular structure of the logic cell arrays in memories or caches. Error detection logic for the cell arrays of protected memories and caches must therefore enforce detection of multiple bit errors. Ideally, any bit errors can be detected. However, the detection logic must at least detect a sufficient large region of adjacent memory cells. This reflects an attack by a localised laser spot.

Detection may be probabilistic; however, the probability of a missed error must be smaller than 1:1.000.000.

Detection of bit errors does not mean that the bit errors must be detected immediately after the error occurred. It is sufficient to detect the error will before it can propagate to a failure in the system. For instance, flipping bits in memory cells must be detected before the affected memory cells are read out by the CPU.

Figure 2 Overview of SDP related components and interfaces

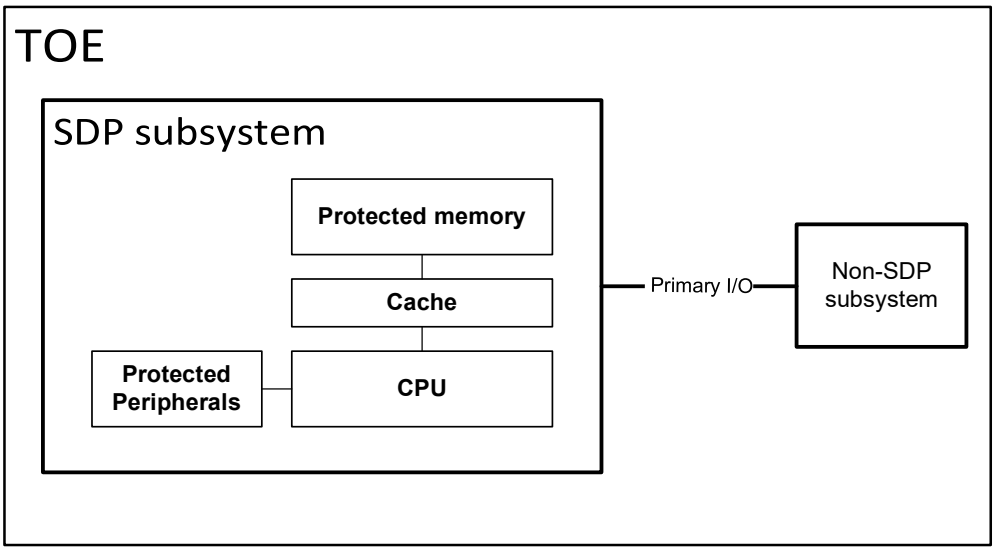


Figure 3 FPT_SDP: Component levelling

Management: there is no management functionality foreseen in context of FPT_SDP.1, as the self-secured digital protection shall be always active.

Audit: there are no auditable events foreseen in the context of FPT_SDP.1.

5.2.2 FPT_SDP.1 - Self-secured digital protection

Table 17 FPT_SDP.1

FPT_SDP.1	Self-secured digital protection
Hierarchical to	No other components.
Dependencies:	ATE_SDP.1 FDP_SDI.2
FPT_SDP.1.1	The TSF shall enforce detection of modified user and TSF data resulting from single bit errors in the sequential logic cells of the CPU and [assignment: protected peripherals].
FPT_SDP.1.2	The TSF shall enforce detection of modified user and TSF data resulting from [selection: any, adjacent [assignment: number of bits]] bit errors in the memory cells of the caches.
FPT_SDP.1.3	The TSF shall enforce detection of modified user and TSF data resulting from [selection: any, adjacent [assignment: number of bits]] bit errors in the memory cells of the protected memories.

Application Notes:

- *Single bit error* means that one bit flip can occur at one specific arbitrary sequential cell at an arbitrary clock cycle.
- *Any bit errors* mean that any number of bit flips may occur on the sequential cells of the memory at an arbitrary clock cycle.
- *Adjacent <number of bits> bit errors* means that in any set of < *number of bits* > adjacent memory cells each bit may be bit flipped.
- Protected memories are the memories defined in FDP_SDI.2
- Detection may be probabilistic. However, the probability of a missed detection shall be smaller than 1:1.000.000.
- Detection means that the TOE shall prevent exploitation of the modified User or TSF data.

5.3 ATE_SDP – SDP Testing

5.3.1 Objectives

The objective of this test family is to provide additional assurance concerning the correctness and effectiveness of implemented SDP functionality, extending beyond what is achievable by functional testing according to ATE_FUN and penetration testing according to AVA_VAN. While ATE_FUN evaluation activities focus on the functional testing of the TOE, ATE_SDP evaluation activities involve explicit fault testing of the digital logic of the SDP subsystem. SDP testing allows the injection of specific one-bit faults into the SDP subsystem of the TOE, providing a level of control that is impossible for physical fault injection testing according to the AVA_VAN family. Verification methodologies, e.g., simulation or emulation shall be used for fault injection to verify that single bit errors in logical cells belonging to CPU and protected peripherals are properly detected.
Arbitrary multi-bit flips cannot be usefully tested by fault simulation or emulation due to computational complexity reasons. Therefore, the logic cells and memory fields of protected memories and caches are not in scope of fault simulation or emulation testing. Instead, this functionality shall be verified as part of the analysis performed in ADV_TDS.4 and ADV_ARC.1.
The goal of SDP testing is to achieve applicability, comparability, and completeness within a defined verification scope by means of focusing verification on the essential logic first and avoiding technical barriers for verification of logic inferred by electronic design automation (EDA), like clock trees and clock gates, reset trees, design for testability (DfT) logic, and power control logic.

5.3.2 Component levelling

This family contains one component

5.3.3 Application notes

Similar as in functional testing, SDP testing uses a test script for test stimulus or a test program to stimulate a specific functional behaviour of the SDP subsystem.

Each test stimulus shall iterate over injecting one bitflip fault in any sequential cell of the sequential logic of the CPU and protected peripherals. Logic cells which define the data of protected memory and caches or whose outputs are exclusively connected to unprotected hardware components or EDA-inferred logic may be disregarded concerning fault injection.

The fault emulation test shall be considered as pass if the fault injection is detected or the CPU enters a secure state before the injected error causes exploitable failures.

5.3.4 ATE_SDP.1 SDP Testing

Dependencies:

- ADV_TDS.4
- ADV_IMP.1
- ADV_ARC.1

Developer action elements:

ATE_SDP.1.1D: The developer shall document the *fault injection testing* of the SDP subsystem.

Content and presentation elements:

ATE_SDP.1.1C: The test documentation shall contain one or more SDP-testing netlists which cover the SDP subsystem.

ATE_SDP.1.2C: The test documentation shall contain a rationale demonstrating that the SDP-testing netlists are adequate design representations of the TOE's product netlist. Adequate means that

- The provided SDP-testing netlists may be partly or fully independent of any product cell libraries.
- logic inferred by electronic design automation (EDA), like clock trees and clock gates, reset trees, design for testability (DfT) logic, and power control logic may be omitted.
- logic of memory cells may be replaced by functional equivalent mock-ups.

ATE_SDP.1.3C: The test documentation shall contain the test stimulus specifications and expected results.

ATE_SDP.1.4C: The SDP testing shall be considered as pass if the fault injection is detected before it causes exploitable failures.

ATE_SDP.1.5C: The test documentation shall contain a rationale demonstrating that all sequential cells contained in CPU and protected peripherals are covered by each test stimulus. Logic whose outputs are exclusively connected to unprotected hardware components may be omitted

ATE_SDP.1.6C: The test documentation shall contain a mapping demonstrating that all security relevant functions of the SDP subsystem are covered by fault injection tests.

Evaluator action elements:

ATE_SDP.1.1E: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

The detailed CEM part of ATE_SDP.1 is defined in chapter 10.

5.3.5 Assurance rationale

The extended assurance component ATE_SPD.1 only adds how to functionally test for correct and effective implementation of the self-secured digital protection and therefore neither contradict the inherited assurance components from the base PP [PP0084], nor any other assurance components of [CC3].

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 18 FCS_RNG.1/TRNG

FCS_RNG.1/TRNG	Random Number Generation
Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1/TRNG	The TSF shall provide a physical random number generator that implements: (PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output. (PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source[1]. (PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected. (PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon. (PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered continuously[2]. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.
FCS_RNG.1.2/TRNG	The TSF shall provide 32-bit numbers[3] that meet

[1] [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]

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

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

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

6.1.2 Cryptographic services implemented in hardware

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

Table 19 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 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/AES	The TSF shall perform decryption and encryption in accordance with a specified cryptographic algorithm AES in ECB mode[2] and cryptographic key sizes 128 bit, 192 bit, 256 bit[3] 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 20 FCS_CKM.6/AES

FCS_CKM.6/AES	Timing and event of cryptographic key destruction
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]
FCS_CKM.6.1/AES	The TSF shall destroy a cryptographic AES key[4] when a user instructs the symmetric coprocessor to clear the key[5].
FCS_CKM.6.2/AES	The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method overwriting or zeroing[6] that meets the following: None[7]

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.

[1] [assignment: additional standard test suites]
[2] [selection: ECB mode, CBC mode]
[3] [selection: 128 bit, 192 bit, 256 bit]
[4] [assignment: list of cryptographic keys (including keying material)]
[5] [selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]
[6] [assignment: cryptographic key destruction method]
[7] [assignment: list of standards]

Table 21 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 the following self-tests at the request of the authorized user[1] to demonstrate the correct operation of parts of TSF[2] : tests to verify correct behaviour of alarm propagation and security optimized wiring[3].
FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of the boot code[4].
FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of alarm behavior and security optimized wiring[5].

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
- CPU stall

According to [PP0084] Application Note 15

No Audit data are collected, because the effect entering the secure state is immediately visible.

[7] [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]]

[2] [selection: [assignment: parts of TSF], the TSF]

[2] [assignment: list of self-tests run by the TSF].

[3] [selection: [assignment: parts of TSF data], TSF data]

[5] [selection: [assignment: parts of TSF], TSF]

6.1.4.1 Self-Secured Digital Protection (SDP)

The following SFR enforces digital error detection by the SDP subsystem

Table 22 FPT_SDP.1

FPT_SDP.1	Self-secured digital protection
Hierarchical to	No other components.
Dependencies:	ATE_SDP.1 FDP_SDI.2
FPT_SDP.1.1	The TSF shall enforce detection of modified user and TSF data resulting from single bit errors in the sequential logic cells of the CPU and cache controller,NVIC, MISE, MPU, SAU, MCICE[1].
FPT_SDP.1.2	The TSF shall enforce detection of modified user and TSF data resulting from adjacent 28[2] bit errors in the memory cells of the caches.
FPT_SDP.1.3	The TSF shall enforce detection of modified user and TSF data resulting from any bit errors in the memory cells of the protected memories.

The following security policy enforces that the TOE hardware (i.e. SDP error detection together with physical countermeasures) provides sufficient protection against any kinds of modification attacks targeting the SDP subsystem. It is important to emphasize that no support from the embedded software is needed.

Table 23 FDP_ITT.1/SDP

FDP_ITT.1/SDP	Basic internal transfer protection
Hierarchical to	No other components.
Dependencies:	FDP_ACC.1 or FDP_IFC.1
FDP_ITT.1.1/SDP	The TSF shall enforce the SDP Processing Policy[3] to prevent the modification[4] of user data when it is transmitted between physically-separated parts of the TOE

Refinement: The components of the SDP subsystem are seen as physically-separated parts of the TOE.

Table 24 FPT_ITT.1/SDP

FPT_ITT.1/SDP	Basic internal TSF data transfer protection
Hierarchical to	No other components.
Dependencies:	No dependencies
FPT_ITT.1.1/SDP	The TSF shall protect TSF data from modification[5] when it is transmitted between separate parts of the TOE.

Refinement: The components of the SDP subsystem are seen as physically-separated parts of the TOE.

[1] [assignment: protected peripherals]
[2] [selection: any, adjacent [assignment: number of bits]]
[3] [assignment: access control SFP(s) and/or information flow control SFP(s)]
[4] [selection: disclosure, modification, loss of use]
[5] [selection: disclosure, modification]

Table 25 FDP_IFC.1/SDP

FDP_IFC.1/SDP	Subset information flow control
Hierarchical to	No other components.
Dependencies:	FDP_IFF.1
FDP_IFC.1.1/SDP	The TSF shall enforce the SDP Processing Policy[1] on all data when they are processed or transferred by the SDP subsystem or by the Security IC Embedded Software[2].

SDP Processing Policy: The following Security Function Policy (SFP) SDP Processing Policy is defined for the requirement FDP_IFC.1/SDP:
Modification of User data of the Composite TOE and TSF data in the SDP subsystem shall be detected before the modified data is used by the embedded software. The detection shall be done solely by hardware without the support from the Security IC Embedded Software. Please note that write suppression of NVM (e.g. by tearing due to power loss) cannot be prevented and therefore is not considered as a modification of user or TSF data.

6.1.5 Abuse of Functionality

This chapter relates to the section “Abuse of Functionality” in [PP0084] ch. 6.1. SFR’s FMT_LIM.1 and FMT_LIM.2 from [PP0084] are adapted due to CC:2022.

Table 26 FMT_LIM.1

FMT_LIM.1	Limited Capabilities
Hierarchical to:	No other components.
Dependencies:	FMT_LIM.2: Limited availability
FMT_LIM.1.1	The TSF shall limit its capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced: Deploying Test Features after TOE Delivery does not allow user data of the Composite TOE to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks[3].

Table 27 FMT_LIM.2

FMT_LIM.2	Limited availability
Hierarchical to:	No other components.
Dependencies:	FMT_LIM.1 Limited capabilities.
FMT_LIM.2.1	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: Deploying Test Features after TOE Delivery does not allow user data of the Composite TOE to be disclosed or manipulated, TSF data to be disclosed or

[1] [assignment: information flow control SFP]
[2] [assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]
[3] [assignment: Limited capability and availability policy]

	manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks[1].

Table 28 FAU_SAS.1

FAU_SAS.1	Audit Storage
Hierarchical to	No other components.
Dependencies	No dependencies.
FAU_SAS.1.1	The TSF shall provide the test process before TOE delivery[2] with the capability to store the initialization data and/or pre-personalization data and/or supplements of the security IC embedded software[3] in the access protected and not changeable areas of the non-volatile memory[4].

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 29 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 all user data[5] while it is stored in the any memory[6].

Table 30 FDP_SDI.2

FDP_SDI.2	Stored data integrity monitoring and action
Hierarchical to	FDP_SDI.1
Dependencies	No dependencies.

[1] [assignment: Limited capability and availability policy]
[2] [assignment: list of subjects]
[3] [assignment: list of audit information]
[4] [assignment: type of persistent memory]
[5] [selection: all user data, the following user data [assignment: list of user data]]
[6] [selection: temporary memory, persistent memory, any memory]

FDP_SDI.2	Stored data integrity monitoring and action
FDP_SDI.2.1	The TSF shall monitor user data stored in containers controlled by the TSF for EDC integrity errors[1] on all objects, based on the following attributes: the corresponding EDC value with a length of at least 28 bits in the RAM, ROM, and NVM[2].
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall enter a secure state[3].

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

[1] [assignment: integrity errors]
[2] [assignment: user data attributes]
[3] [assignment: action to be taken]

- 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:

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 31 FDP_ACC.2/AF

FDP_ACC.2/AF	Complete access control
Hierarchical to	FDP_ACC.1
Dependencies	FDP_ACF.1
FDP_ACC.2.1/AF	The TSF shall enforce the Application Firewall access control policy[1] on subjects, objects and operations defined in 6.1.8.1[2] 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 32 FDP_ACF.1/AF

FDP_ACF.1/AF	Security attribute based access control
Hierarchical to	No other components.
Dependencies	FDP_ACC.1 FMT_MSA.3
FDP_ACF.1.1/AF	The TSF shall enforce the Application Firewall access control policy[3] to objects based on the following: The subjects, objects, operations and associated security attributes defined in 6.1.8.1[4].
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: Rules defined in 6.1.8.1[5].
FDP_ACF.1.3/AF	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: None[6].
FDP_ACF.1.4/AF	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: None[7].

[1] [assignment: access control SFP]

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

[3] [assignment: access control SFP]

[4] [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]

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

[6] [assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects].

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

Table 33 FMT_MSA.3/AF

FMT_MSA.3/AF	Static attribute initialisation
Hierarchical to	No other components.
Dependencies	FMT_MSA.1 FMT_SMR.1
FMT_MSA.3.1/AF	The TSF shall enforce the Application Firewall access control policy[1] to provide restrictive[2] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/AF	The TSF shall allow the none[3] 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 34 FMT_MSA.1/AF/S

FMT_MSA.1/AF/S	Management of security attributes
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 Application Firewall access control policy[4] to restrict the ability to modify[5] the security attributes PO(x), acc(x), sec(x), XN(x)[6] to secure AF management in case sec(x) = S[7].

Table 35 FMT_MSA.1/AF/NS

FMT_MSA.1/AF/NS	Management of security attributes
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 Application Firewall access control policy[8] to restrict the ability to modify[9] the security attributes PO(x), acc(x), XN(x)[10] to secure or non-secure AF management in case sec(x) = NS[11].

[1] [assignment: access control SFP, information flow control SFP]
[2] [selection, choose one of: restrictive, permissive, [assignment: other property]]
[3] [assignment: the authorized identified roles]
[4] [assignment: access control SFP(s), information flow control SFP(s)]
[5] [selection: change_default, query, modify, delete, [assignment: other operations]]
[6] [assignment: list of security attributes]
[7] [assignment: the authorized identified roles]
[8] [assignment: access control SFP(s), information flow control SFP(s)]
[9] [selection: change_default, query, modify, delete, [assignment: other operations]]
[10] [assignment: list of security attributes]
[11] [assignment: the authorized identified roles]

Table 36 FMT_SMF.1/AF

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

Table 37 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 secure AF management and non-secure AF management[2].
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 38 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 an authentication mechanism according to [ISO 9798-2] section 7.3.3 MUT.CR-Three-pass authentication[3] to prove the identity of the TOE[4] by including the following properties proof of knowledge of Flash Loader administrator or download operator credentials[5] to an external entity.

Note: FIA_API is only available, if the Flash Loader is active and mutual authentication of the Flash Loader with the roles Administrator User and Download Operator User is not deactivated.

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 or during operation in the field (life cycle phase 7). This TOE shall support both Loader packages from [PP0084] section 7.3.

[1] [assignment: list of management functions to be provided by the TSF]
[2] [assignment: the authorised identified roles]
[3] [assignment: authentication mechanism]
[4] [selection: TOE, [assignment: object, authorized user or role]]
[5] [assignment: list of properties]

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

Note: The TOE can optionally deactivate mutual authentication of the Flash Loader with the roles Administrator User, Download Operator User. In that case, package 2 is no longer applicable.However, a subset of SFRs is still applicable in that case. SFRs which are not applicable in that case, are explicitly stated in the notes following the SFR.

The SFRs FDP_UCT.1 and FDP_UIT.1 are specified in [PP0084].

Table 39 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 limit its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Deploying Loader functionality after permanent deactivation does not allow stored user data to be disclosed or manipulated by unauthorized user. The TSF prevents deploying the Loader functionality after permanent deactivation[1].

Table 40 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: Deploying Loader functionality after permanent deactivation does not allow stored user data to be disclosed or manipulated by unauthorized user. The TSF prevents deploying the Loader functionality after permanent deactivation[2].

Table 41 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 Administrator User or Download Operator User and Image Provider[3] 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.

[1] [assignment: Limited capability and availability policy]
[2] [assignment: Limited capability and availability policy]
[3] [assignment: users authorized for using the Loader]

FTP_ITC.1	Inter-TSF trusted channel
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for deploying Loader for downloading User Data and modification of authentication keys[1].

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.

Note: This SFR is not applicable if mutual authentication is disabled in the Flash Loader.

Table 42 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 Administrator User, Download Operator User and Image Provider[2], (2) the objects user data in NVM[3], (3) the operation deployment of Loader.

Table 43 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 Administrator User, Download Operator User and Image Provider[4] with security attributes None[5]. (2) the objects user data in NVM[6] with security attributes None[7].
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: Not in field use life cycle: The authenticated Administrator User or authenticated Download Operator User can modify the user data by new user data when the new user data is authorized by the Image Provider In field us life cycle:

[1] [assignment: rules]
[2] [assignment: authorized roles for using Loader]
[3] [assignment: memory areas]
[4] [assignment: authorized roles for using Loader]
[5] [assignment: SFP relevant security attributes, or named groups of SFP relevant security attributes]
[6] [assignment: memory areas]
[7] [assignment: SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

FDP_ACF.1/Loader	Security attribute based access control - Loader
	The authenticated Download Operator User can modify the user data by new user data when the new user data is authorized by the Image Provider[1].
FDP_ACF.1.3/Loader	The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: None[2].
FDP_ACF.1.4/Loader	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: None[3].

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.

Note: During In field use life cycle state the administrator role does not exist.

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 44 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 modify, delete[4] the Authentication keys for Administrator User, Download Operator User and Image Provider[5] to Administrator User, Download Operator User, User OS[6].

Note: During not in field life cycle state: 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, otherwise manage the keys for the Download Operator User only. The image provider cannot modify any keys or perform authentication with the Flash Loader. It can simply build authentic loadable images.

During in field life cycle state: The User OS can manage (modify) the Image Provider key.

Table 45 FMT_SMR.1/Loader

FMT_SMR.1/Loader	Security roles
Hierarchical to	No other components.
Dependencies	FIA_UID.1

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

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

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

[4] [selection: change_default, query,modify, delete, clear, [assignment: other operations]]

[5] [assignment: list of TSF data]

[6] [assignment: the authorised identified roles]

FMT_SMR.1/Loader	Security roles
FMT_SMR.1.1/Loader	The TSF shall maintain the roles Administrator User, Download Operator User, Image Provider, User OS[1].
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 46 FMT_SMF.1/Loader

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

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 47 FIA_UID.2/Loader

FIA_UID.2/Loader	User identification before any action
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.

Table 48 FPT_FLS.1/Loader

FPT_FLS.1/Loader	Failure with preservation of secure state
Hierarchical to	No other components.
Dependencies	No dependencies.
FPT_FLS.1.1/Loader	The TSF shall preserve a secure state when the following types of failures occur: flash loader active and image not (yet) successfully loaded[3].

The security functional requirements FIA_API.1, 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 und FPT_FLS.1/Loader apply only to TOE products with activated Flash Loader. In other cases, the Flash Loader is not available anymore and the user data download is completed.

Application Note: Secure State refers to a state, whereby the additional code is not successfully loaded (yet) and flash loader is active. This includes the process of downloading a user image. No user image can be executed in the secure state.

[1] [assignment: the authorised identified roles]

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

[3] [assignment: list of types of failures in the TSF]

6.1.11 Crypto Suite

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

6.1.11.1 AES

This section describes AES ciphers provided by the Crypto Suite.

Table 49 FCS_COP.1/CS/AES/<iter>

FCS_COP.1/CS/AES/<iter>	Cryptographic operation
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/CS/AES/<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 50 Cryptographic table for FCS_COP.1/CS/AES/<iter>

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
ENC	encryption and decryption	AES in ECB, CBC, CTR mode	128,192,256 bits	[FIPS 197] [SP 800-38A]
CMAC	MAC generation	AES CMAC mode	128,192,256 bits	[FIPS 197] [SP 800-38B]
CBC-MAC	MAC generation	AES CBC MAC mode	128,192,256 bits	[FIPS 197] [ISO 9797-1] padding method 2
CCM	Authenticated encryption	AES CCM Mode	128,192,256 bits	[FIPS 197] [SP 800-38C]
GCM	Authenticated encryption	AES GCM Mode	128,192,256 bits	[FIPS 197] [SP 800-38D]

Table 51 FCS_CKM.6/CS/AES

FCS_CKM.6/CS/AES	Timing and event of cryptographic key destruction
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]

FCS_CKM.6.1/CS/AES	The TSF shall destroy cryptographic AES key[1] when a user sets new keys[2].
FCS_CKM.6.2/CS/AES	The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method overwriting or zeroing[3] that meets the following: None[4]

6.1.11.2 TDES

This section describes DES ciphers provided by the Crypto Suite.

Table 52 FCS_COP.1/CS/TDES/<iter>

FCS_COP.1/CS/TDES/<iter>	Cryptographic operation
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/CS/TDES/<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 53 Cryptographic table for FCS_COP.1/CS/TDES/<iter>

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
ENC	encryption and decryption	TDES ECB, CBC, CTR mode	112, 168 bits	[SP 800-67] [SP 800-38A]
RMAC	MAC calculation	TDES Retail MAC	112 bits	[SP 800-67] [ISO 9797-1]
CBC-MAC	MAC calculation	TDES CBC MAC	112, 168 bits	[SP 800-67] [ISO 9797-1] padding method 2

[1] [assignment: list of cryptographic keys (including keying material)]
[2] [selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]
[3] [assignment: cryptographic key destruction method]
[4] [assignment: list of standards]

Table 54 FCS_CKM.6/CS/TDES

FCS_CKM.6/CS/TDES	Timing and event of cryptographic key destruction
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]
FCS_CKM.6.1/CS/TDES	The TSF shall destroy cryptographic TDES key[1] when a user sets new keys[2].
FCS_CKM.6.2/CS/TDES	The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method overwriting or zeroing[3] that meets the following: None[4]

6.1.11.3 HMAC

This section describes HMAC algorithms provided by the Crypto Suite.

Table 55 FCS_COP.1/CS/HMAC/<iter>

FCS_COP.1/CS/HMAC/<iter>	Cryptographic operation
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/CS/HMAC/<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 56 Cryptographic table for FCS_COP.1/CS/HMAC/<iter>

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
SHA	HMAC calculation	HMAC with SHA-1, SHA256, SHA384, SHA512	160, 256, 384, 512 bits	[FIPS 180-4] [FIPS 198-1]

[1] [assignment: list of cryptographic keys (including keying material)]

[2] [selection: no longer needed, [assignment: other circumstances for key or keying material destruction]]

[3] [assignment: cryptographic key destruction method]

[4] [assignment: list of standards]

6.1.11.4 FFC

This section describes Finite Field algorithms provided by the Crypto Suite.

Table 57 FCS_COP.1/CS/FFC/<iter>

FCS_COP.1/CS/FFC/<iter>	Cryptographic operation
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/CS/FFC/<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 58 Cryptographic table for FCS_COP.1/CS/FFC/<iter>

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
DH	key agreement	Finite field Diffie-Hellman	1024-2048 bits	[SP 800-56A], ch. 5.7.1.1 w/o step 2

Table 59 FCS_CKM.1/CS/FFC

FCS_CKM.1/CS/FFC	Cryptographic key generation
Hierarchical to	No other components.
Dependencies	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1] [FCS_RBG.1 or FCS_RNG.1] FCS_CKM.6
FCS_CKM.1.1/CS/FFC	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm FFC key generation[1] and specified cryptographic key sizes of 1024-2048 bits[2] that meet the following: [SP 800-56A], ch. 5.6.1.1[3].

The following table lists the certified FFC domain parameters. Please note that the Crypto Suite supports any parameters which define cyclic groups of order up to 256 bits.

Table 60 Certified FFC domain parameter

Domain parameters	Reference
1024-bit MODP Group with 160-bit Prime Order Subgroup	[RFC 5114]

[1] [assignment: cryptographic key generation algorithm]

[2] [assignment: cryptographic key sizes]

[3] [assignment: list of standards]

Domain parameters	Reference
2048-bit MODP Group with 224-bit Prime Order Subgroup	[RFC 5114]
2048-bit MODP Group with 256-bit Prime Order Subgroup	[RFC 5114]

6.1.11.5 RSA

This section describes RSA related algorithms provided by the Crypto Suite.

Table 61 FCS_COP.1/CS/RSA/<iter>

FCS_COP.1/CS/RSA/<iter>	Cryptographic operation
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/CS/RSA/<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 62 Cryptographic table for FCS_COP.1/CS/RSA/<iter>

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
ENC	encryption	RSAEP	1024-4224 bits	[PKCS#1], ch. 5.1.1 [SP 800-56B], ch. 7.1.1
DEC	decryption	RSADP	1024-2112 bits	[PKCS#1], ch. 5.1.2, 2a [SP 800-56B], ch. 7.1.2.1
DEC_CRT	decryption	RSADP(CRT)	1024-4224 bits	[PKCS#1], ch. 5.1.2, 2b [SP 800-56B], ch. 7.1.2.3
SIG	signature generation	RSASP1	1024-2112 bits	[PKCS#1], ch. 5.2.1, 2a
SIG_CRT	signature generation	RSASP1(CRT)	1024-4224 bits	[PKCS#1], ch. 5.2.1, 2b
VER	signature verification	RSAVP1	1024-4224 bits	[PKCS#1], ch. 5.2.2

Note: For SIG and SIG_CRT, the additional constrains defined in [FIPS 186-5] ch. 5.4 can be fulfilled by the application software using the Crypto Suite.

Note: SIG, SIG_CRT and VER do not include padding of the input message.

Table 63 FCS_CKM.1/CS/RSA/<iter>

FCS_CKM.1/CS/RSA/<iter>	Cryptographic key generation
Hierarchical to	No other components.
Dependencies	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1] [FCS_RBG.1 or FCS_RNG.1] FCS_CKM.6
FCS_CKM.1.1/CS/RSA/<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 64 Cryptographic table for FCS_CKM.1/CS/RSA/<iter>

<iter>	[assignment: cryptographic key generation algorithm]	[assignment: cryptographic key sizes]	[assignment: list of standards]
PRIMES	generate probably random primes p and q	512-2064 bits	[FIPS 186-5], A.1.3 w/o Step 1
EXP	generate RSA (N, d) parameters from p, q	1024-2112 bits	[FIPS 186-5], A.1.1 [PKCS#1], 3.1, 3.2(1)
CRT	generate RSA CRT parameters from p, q	1024-4224 bits	[FIPS 186-5], A.1.1 [PKCS#1], 3.1, 3.2(2)
MR	Miller-Rabin primality test	512-2064 bits	[FIPS 186-5], ch. B.3.1
EMR	Enhanced Miller–Rabin primality test	512-2064 bits	[FIPS 186-5], ch. B.3.2

Note: The key size in PRIMES, MR and EMR are related to each individual prime of the resulting RSA key. This means, the resulting RSA key n = pq can have key size between 1024 and 4128 bits.

Note: PRIMES is only conformant to [FIPS 186-5], A.1.3 for prime Bitlength >= 2048; in case prime Bitlength < 2048 identical algorithm is used but considered proprietary.

Note: EXP generates N and d by two different API calls. The primes p, q must be calculated with the PRIMES algorithm.

Note: CRT does not explicitly generate d as described in [FIPS 186-5], A.1.1, but instead generates the CRT representation of d as described in [PKCS#1], 3.2(2). The primes p, q must be calculated with the PRIMES algorithm.

Note: The listed algorithms are cryptographic primitives which can be used by the composite TOE to generate RSA keys. The embedded application must implement the complete RSA key generation.

6.1.11.6 ECC

This section describes ECC related algorithms provided by the Crypto Suite.

Table 65 FCS_COP.1/CS/ECC/<iter>

FCS_COP.1/CS/ECC/<iter>	Cryptographic operation
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/CS/ECC/<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 66 Cryptographic table for FCS_COP.1/CS/ECC/<iter>

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
ECDSA_SIG	EC signature generation	ECDSA	160-521 bits	[FIPS 186-5], ch. 6.4.1
ECDSA_VER	EC signature verification	ECDSA	160-521 bits	[FIPS 186-5], ch. 6.4.2
ECDH	key agreement	ECDH	160-521 bits	[SP 800-56A], ch. 5.7.1.2 without cofactor multiplication
PACE_IM	PACE integrated mapping	PACE integrated mapping	160-521 bits	[ICAO], Appendix B.2
X25519	key agreement	X25519	256 bits	[RFC 7748]
X448	key agreement	X448	448 bits	[RFC 7748]
ECSDSA_SIG	EC signature generation	ECSDSA	160-521 bits	[ISO 14888-3], ch. 6.10.4 [TR-03111], ch. 4.2.3
ECSDSA_VER	EC signature generation	ECSDSA	160-521 bits	[ISO 14888-3], ch. 6.10.5 [TR-03111], ch. 4.2.3
EdDSA_SIG	EC signature generation	EdDSA	256 bits 456 bits	[FIPS 186-5], ch. 7.6
EdDSA_VER	EC signature verification	EdDSA	256 bits 456 bits	[FIPS 186-5], ch. 7.7

<iter>	[assignment: list of cryptographic operations]	[assignment: cryptographic algorithms]	[assignment: cryptographic key sizes]	[assignment: list of standards]
HashEdDSA_SIG	EC signature generation	EdDSA	256 bits 456 bits	[FIPS 186-5], ch. 7.8.1
HashEdDSA_VER	EC signature verification	EdDSA	256 bits 456 bits	[FIPS 186-5], ch. 7.8.2

Note: The ECDSA_SIG and ECDSA_VER operations will not perform a hash calculation of the input message.

Note: The ECDH operation returns the y-coordinate in addition to the x-coordinate.

Note: The ECDH operation does not perform cofactor multiplication. In case of curves with cofactor > 1, the embedded software must provide appropriate means to prevent the small subgroup attacks.

The following table lists the certified elliptic curves. Please note that the Crypto Suite supports any curve in short Weierstrass form up to 521 bits.

Table 67 Certified elliptic curves

Curve	Representation	Reference
NIST curves over prime fields	Weierstrass	[SP 800-186]
Brainpool curves	Weierstrass	[RFC 5639]
secp160k1, secp160r1, secp160r2, secp256k1	Weierstrass	[SEC2]
ANSII FRP256V1	Weierstrass	[ANSSI]
BN P256	Weierstrass	[ISO 15946]
W-25519, W-448	Weierstrass	[SP 800-186]
Curve25519, Curve448	Montgomery	[RFC 7748]
Ed25519, Ed448	Twisted Edwards	[FIPS 186-5]

Table 68 FCS_CKM.1/CS/ECC/<iter>

FCS_CKM.1/CS/ECC/<iter>	Cryptographic key generation
Hierarchical to	No other components.
Dependencies	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1] [FCS_RBG.1 or FCS_RNG.1] FCS_CKM.6
FCS_CKM.1.1/CS/ECC/<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 69 Cryptographic table for FCS_CKM.1/CS/ECC/<iter>

<iter>	[assignment: cryptographic key generation algorithm]	[assignment: cryptographic key sizes]	[assignment: list of standards]
ECDSA	ECDSA key generation	160-521 bits	[FIPS 186-5], A.2.1
EDDSA	EDDSA key generation	256, 456 bits	[FIPS 186-5], A.2.3
ECSDSA	ECSDSA key generation	160-521 bits	[ISO 14888-3], ch. 6.10.3.

6.1.11.7 ML

This section describes Module Lattice based key encapsulation and signature provided by the Crypto Suite.

Table 70 FCS_COP.1/CS/ML/<iter>

FCS_COP.1/CS/ML/<iter>	Cryptographic operation
Hierarchical to	No other components.
Dependencies	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5] FCS_CKM.6
FCS_COP.1.1/CS/ML/<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 71 Cryptographic table for FCS_COP.1/CS/ML/<iter>

<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
SIG	Signature generation	ML-DSA-44 ML-DSA-65 ML-DSA-87	ML-DSA-44 ML-DSA-65 ML-DSA-87	[FIPS 204] ch. 5.2, ch.5.4
VER	Signature verification	ML-DSA-44 ML-DSA-65 ML-DSA-87	ML-DSA-44 ML-DSA-65 ML-DSA-87	[FIPS 204] ch. 5.3, ch.5.4

Note: In SIG only the hedged variant is certified

Table 72 FCS_CKM.1/CS/ML/<iter>

FCS_CKM.1/CS/ ML/<iter>	Cryptographic key generation
Hierarchical to	No other components.
Dependencies	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1] [FCS_RBG.1 or FCS_RNG.1] FCS_CKM.6
FCS_CKM.1.1/CS/ML/<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 73 Cryptographic table for FCS_CKM.1/CS/ML/<iter>

<iter>	[assignment: cryptographic key generation algorithm]	[assignment: cryptographic key sizes][1]	[assignment: list of standards]
KEM_GEN	MLKEM key generation	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.1
DSA_GEN	MLDSA key generation	ML-DSA-44 ML-DSA-65 ML-DSA-87	[FIPS 204] ch. 5.1

Note: KEM_GEN and DSA_GEN can optionally generate a key by using a seed parameter instead of using a random number generator.

Note: KEM_GEN supports an optional compact key decapsulation format which stores private keys in non-NTT presentation and performs NTT on-the-fly during decapsulation.

Note: DSA_GEN supports a key format where the public matrix A is stored expanded.

6.1.11.8 Hash

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

[1] Public key size

Table 74 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 or FCS_CKM.5] FCS_CKM.6
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 75 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]
SHA1	Hash digest generation	SHA-1	None	[FIPS 180-4]
SHA2	Hash digest generation	SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256	None	[FIPS 180-4]
SHA3	Hash digest generation	SHA3-224, SHA3-256, SHA3-384, SHA3-512	None	[FIPS 202]
XOF	Extendable Output Function	SHAKE128, SHAKE256	None	[FIPS 202]

6.1.11.9 Random

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

Table 76 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	The TSF shall provide a physical random number generator that implements: (PTG.2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.

FCS_RNG.1/CS/PTG2	Random Number Generation
	(PTG.2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source[1]. (PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected. (PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon. (PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered continuously[2]. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.
FCS_RNG.1.2/CS/PTG2	The TSF shall provide 32-bit numbers[3] that meet (PTG.2.6) Test procedure A (None)[4] does not distinguish the internal random numbers from output sequences of an ideal RNG. (PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.

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

Table 77 FCS_RNG.1/CS/PTG3

FCS_RNG.1/CS/PTG3	Random number generation
Hierarchical to	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1/CS/PTG3	The TSF shall provide a hybrid physical random number generator that implements: (PTG.3.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output. (PTG.3.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random

[1] [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]

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

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

[4] [assignment: additional standard test suites]

FCS_RNG.1/CS/PTG3	Random number generation
	number that depends on some raw random numbers that have been generated after the total failure of the entropy source[1]. (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. (PTG.3.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon. (PTG.3.5) The online test procedure checks the raw random number sequence. It is triggered continuously[2]. 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. (PTG.3.6) The algorithmic post-processing algorithm belongs to Class DRG.3 with cryptographic state transition function and cryptographic output function, and the output data rate of the post-processing algorithm shall not exceed its input data rate.
FCS_RNG.1.2/CS/PTG3	The TSF shall provide numbers in 32-bit packets[3] that meet: (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 (none)[4]. (PTG.3.8) The internal random numbers shall use PTRNG of class PTG.2 as random source for the post processing[5].

Table 78 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: (DRG.3.1) If initialized with a random seed using a PTRNG of class PTG.2 as random source[6], the internal state of the RNG shall have at least 100 bit of entropy[7]. (DRG.3.2) The RNG provides forward secrecy.

[1] [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]

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

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

[4] [assignment: additional test suites]

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

[6] [selection: using a PTRNG of class PTG.2 as random source, using a PTRNG of class PTG.3 as random source, using an NPTRNG of class NTG.1 [assignment: other requirements for seeding]]

[7] [selection: have [assignment: amount of entropy], have [assignment: work factor], require [assignment: guess work]]

FCS_RNG.1/CS/DRG3	Random number generation
	(DRG.3.3) The RNG provides backward secrecy even if the current internal state is known.
FCS_RNG.1.2/CS/DRG3	The TSF shall provide numbers that meet: (DRG.3.4) The RNG, initialized with a random seed, where the seed has at least 100 bit of entropy and is derived by a PTG.2 certified PTRNG[1] generates output for which 2^48[2] strings of bit length 128 are mutually different with probability that is greater than 1 – 2^(-24)[3] (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 and the U.S. National Institute of Standards and Technology (NIST) test suite for RNGs used for cryptographic purposes [SP 800-22][4].

Table 79 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	The TSF shall provide a hybrid deterministic random number generator that implements: (DRG.4.1) The internal state of the RNG shall use PTRNG of class PTG.2 as random source[5]. (DRG.4.2) The RNG provides forward secrecy. (DRG.4.3) The RNG provides backward secrecy even if the current internal state is known. (DRG.4.4) The RNG provides enhanced forward secrecy on demand[6]. (DRG.4.5) The internal state of the RNG is seeded by an PTRNG of class PTG.2[7].
FCS_RNG.1.2/CS/DRG4	The TSF shall provide numbers that meet: (DRG.4.6) The RNG generates output for which 2^48[8] strings of bit length 128 are mutually different with probability that is greater than 1 – 2^(-24)[9]. (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 and the U.S. National Institute of

[1] [assignment: requirements for seeding]
[2] [assignment: number of strings]
[3] [assignment: probability]
[4] [assignment: probability]
[5] [selection: use PTRNG of class PTG.2 as random source, have [assignment: work factor], require [assignment: guess work]]
[6] [selection: on demand, on condition [assignment: condition], after [assignment: time]]
[7] [selection: internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]]
[8] [selection: internal entropy source, PTRNG of class PTG.2, PTRNG of class PTG.3, [other selection]]
[9] [assignment: probability]

FCS_RNG.1/CS/DRG4	Random number generation
	Standards and Technology (NIST) test suite for RNGs used for cryptographic purposes [SP 800-22][1].

6.2 Security assurance requirements

In the following Table 80, the security assurance requirements and compliance rationale for augmented refinements are given.

Table 80 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
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

[1] [assignment: additional test suites]

SAR	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	ATE_SDP.1 shall be added to the dependency list in order to reflect reporting of ATE_SDP.1 testing.
ATE_IND.2	No refinement
AVA_VAN.5	Refined in [PP0084]
ATE_SDP.1	Defined in this ST

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

Table 81 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.

Table 82 Rationale for 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.

Table 83 Rationale for SFRs related to O.Secure_UC_Load

SFR	Rationale
FDP_UCT.1	This SFR requires integrity protection of the download image
FDP_UIT.1	This SFR requires confidentiality protection of the download image
FDP_ACC.1/Loader FDP_ACF.1/Loader	Both SFRs defines the roles of the loader policy. The User OS can change the image provider key and thus exclude images, which were not generated using this specific key. The image provider role is responsible for generating authentic and integrity protected images-
FMT_MTD.1/Loader	This SFR requires that the TOE provides management functions for modification and deletion of authentication keys.
FPT_FLS.1/Loader	This SFR requires atomic secure updates or secure fail safe. When loader is active in the field, it can only exit this state, if a successful download is performed.

Note: FDP_UCT.1 and FDP_UIT.1 do not require the FTP_ITC.1 dependency, because O.Secure_UC_Load does not require a mutual authentication.

Table 84 Rationale for SFRs related to O.Secure_UC_Activation

SFR	Rationale
FPT_FLS.1/Loader	This SFR requires atomic secure updates or secure fail safe. When loader is active in the field, it can only exit this state, if a successful download is performed.

Table 85 Rationale for SFRs related to O.Prot_TSF_Confidentiality

SFR	Rationale
FTP_ITC.1	This SFR requires a trusted channel
FDP_ACC.1/Loader FDP_ACF.1/Loader	Both SFRs defines the roles of the loader policy. The User OS can change the image provider key and thus exclude images, which were not generated using this specific key. The image provider role is responsible for generating authentic and integrity protected images.

6.3.1.2 Additional SFRs related to O.Malfunction

The FPT_SDP.1 component defines digital error detection capability for the SDP subsystem. This SFR states that the TOE does not only self-protect by using physical countermeasures like sensors and filters but also uses strong forms of digital error detection by redundancy and strong EDCs.

FDP_ITT.1/SDP , FPT_ITT.1/SDP and FDP_IFC.1/SDP define that the TOE hardware of the SDP subsystem must protect against malfunction without support from the embedded software.

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.AES

The optional Crypto Suite provides the AES chaining modes ECB, CTR, CBC, the MAC modes CMAC and CBC-MAC and the authenticated encryption modes GCM and CCM. The associated SFRs are defined in chapter 6.1.11.1.

6.3.1.5 Additional SFRs related to O.TDES

The optional Crypto Suite provides the DES/TDES chaining modes ECB, CTR, CBC, CBC-MAC and Retail-MAC. The associated SFRs are defined in chapter 6.1.11.2. The SFR FCS_COP.1/TDES is remapped to FCS_COP.1/CS/TDES and FCS_CKM.4/TDES is remapped to FCS_CKM.6/CS/TDES.

6.3.1.6 Additional SFRs related to O.HMAC

The optional Crypto Suite provides the HMAC calculation. The associated SFR are defined in chapter 6.1.11.3

6.3.1.7 Additional SFRs related to O.FFC

The optional Crypto Suite provides Finite Field cryptographic services. The associated SFRs are defined in chapter 6.1.11.4

6.3.1.8 Additional SFRs related to O.RSA

The optional Crypto Suite provides RSA functions. The associated SFRs are defined in chapter 6.1.11.5.

6.3.1.9 Additional SFRs related to O.ECC

The optional Crypto Suite provides ECC functions. The associated SFRs are defined in chapter 6.1.11.6.

6.3.1.10 Additional SFRs related to O.Hash

The optional Crypto Suite provides Hash functions. The associated SFRs are defined in chapter 6.1.11.8.

6.3.1.11 Additional SFRs related to O.ML

The optional Crypto Suite provides Module lattice based Post Quantum cyrptography. The associated SFRs are defined in chapter .

6.3.1.12 Additional SFRs related to O.RND

The Hardware and the optional Crypto Suite provide random functions. The associated SFRs are defined in chapters 6.1.1 and 6.1.11.9.

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 86 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	Fulfilled by FMT_MSA.3/AF
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_SMR.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_SMR.1/AF
	FMT_SMF.1	Fulfilled by FMT_SMF.1/AF
FMT_SMR.1/AF	FIA_UID.1	Not applicable, because the role is implicitly 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
	FDP_ACC.1	Fulfilled by FDP_ACC.1/Loader
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_FLS.1/Loader	None	No dependency
FMT_LIM.1/Loader	FMT_LIM.2	Fulfilled by FMT_LIM.2/Loader
FMT_LIM.2/Loader	FMT_LIM.1	Fulfilled by FMT_LIM.1/Loader
FPT_TST.1	None	No dependency
FCS_COP.1/AES	FCS_CKM.6	Fulfilled by FCS_CKM.6/AES
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	The TOE does not provide services to generate symmetric keys. This will be done by the embedded software for the composite TOE.

SFR	Dependencies	Rationale
FCS_CKM.6/AES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	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/AES/<iter>	FCS_CKM.6	Fulfilled by FCS_CKM.6/CS/AES
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	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_CKM.6/CS/AES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	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/TDES/<iter>	FCS_CKM.6	Fulfilled by FCS_CKM.6/CS/TDES
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	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_CKM.6/CS/TDES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	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/HMAC/<iter>	FCS_CKM.6	The TOE does not provide services to destroy HMAC keys. This will be done by the embedded software for the composite TOE.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	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/FFC/<iter>	FCS_CKM.6	The TOE does not provide services to destroy FFC keys. This will be done by the embedded software for the composite TOE.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	Fulfilled by FCS_CKM.1/CS/FFC/<iter>
FCS_CKM.1/CS/FFC/<iter>	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1]	Fulfilled by FCS_COP.1/CS/FFC/<iter>
	[FCS_RBG.1 or FCS_RNG.1]	Fulfilled by FCS_RNG.1/TRNG or FCS_RNG.1/CS/*
	FCS_CKM.6	The TOE does not provide services to destroy FFC keys. This will be done by the embedded software for the composite TOE.

SFR	Dependencies	Rationale
FCS_COP.1/CS/RSA/<iter>	FCS_CKM.6	The TOE does not provide services to destroy RSA keys. This will be done by the embedded software for the composite TOE.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	Fulfilled by FCS_CKM.1/CS/RSA/<iter>
FCS_CKM.1/CS/RSA/<iter>	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1]	Fulfilled by FCS_COP.1/CS/RSA/<iter>
	[FCS_RBG.1 or FCS_RNG.1]	Fulfilled by FCS_RNG.1/TRNG or FCS_RNG.1/CS/*
	FCS_CKM.6	The TOE does not provide services to destroy RSA keys. This will be done by the embedded software for the composite TOE.
FCS_COP.1/CS/ECC/<iter>	FCS_CKM.6	The TOE does not provide services to destroy ECC keys. This will be done by the embedded software for the composite TOE.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	Fulfilled by FCS_CKM.1/CS/ECC/<iter>
FCS_CKM.1/CS/ECC/<iter>	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1]	Fulfilled by FCS_COP.1/CS/ECC/<iter>
	[FCS_RBG.1 or FCS_RNG.1]	Fulfilled by FCS_RNG.1/TRNG or FCS_RNG.1/CS/*
	FCS_CKM.6	The TOE does not provide services to destroy ECC keys. This will be done by the embedded software for the composite TOE
FCS_CKM.1/CS/ML/KEM_GEN	FCS_CKM.6	The TOE does not provide services to destroy MLKEM keys. This will be done by the embedded software for the composite TOE.
	[FCS_RBG.1 or FCS_RNG.1]	Fulfilled by FCS_RNG.1/TRNG or FCS_RNG.1/CS/*
	[FCS_CKM.2 or FCS_CKM.5 or FCS_COP.1]	Fulfilled by FCS_COP.1/CS/ML/ENC Fulfilled by FCS_COP.1/CS/ML/DEC
FCS_COP.1/CS/ML/ENC FCS_COP.1/CS/ML/DEC	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	Fulfilled by FCS_CKM.1/CS/ML/KEM_GEN
	FCS_CKM.6	The TOE does not provide services to destroy ML keys. This will be done by the embedded software for the composite TOE

SFR	Dependencies	Rationale
FCS_CKM.1/CS/ML/DSA_GEN	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.2 or FCS_CKM.5]	Fulfilled by FCS_COP.1/CS/ML/SIG or FCS_COP.1/CS/ML/VER
	[FCS_RBG.1 or FCS_RNG.1]	Fulfilled by FCS_RNG.1/TRNG or FCS_RNG.1/CS/*
	FCS_CKM.6	The TOE does not provide services to destroy ML keys. This will be done by the embedded software for the composite TOE
FCS_COP.1/CS/ML/SIG FCS_COP.1/CS/ML/VER	FCS_CKM.6	The TOE does not provide services to destroy MLDSA keys. This will be done by the embedded software for the composite TOE.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	Fulfilled by FCS_CKM.1/CS/ML/DSA_GEN
FCS_COP.1/CS/Hash/<iter>	FCS_CKM.6	n/A, as a hash function does not use keys.
	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 or FCS_CKM.5]	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
FPT_SDP.1	ATE_SDP.1	Fulfilled by ATE_SDP.1
	FDP_SDI.2	Fulfilled by FDP_SDI.2
FDP_ITT.1/SDP	[FDP_ACC.1 or FDP_IFC.1]	Fulfilled by FDP_IFC.1/SDP
FPT_ITT.1/SDP	None	No dependency
FDP_IFC.1/SDP	FDP_IFF.1	Part 2 of the Common Criteria defines the dependency of FDP_IFC.1 on FDP_IFF.1. The specification of FDP_IFF.1 would not capture the nature of the security functional requirement nor add any detail. As stated in the SDP Processing Policy referred to in FDP_IFC.1/SDP there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1/SDP and its SDP Processing Policy (FDP_IFC.1/SDP).

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.

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.

The assurance class ATE_SDP.1 is defined because this TOE claims error detection mechanisms by FPT_SDP. 1 which cannot be exhaustively tested by the existing ATE and AVA classes alone. ATE_SDP.1 has dependencies to the assurance components ADV_ARC.1, ADV_IMP.1 and AVA_TDS.4, which are always fulfilled because this ST claims EAL.6 assurance package.

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 87 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	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 to phase 7), 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. For phase 7 usage, the user can optionally deactivate the mutual authentication between Administrator/Download Operator User and Flash Loader. In that case, Loader Package 2 and Authentication of the Security IC is not applicable.

Alternatively in case the user has ordered TOE derivatives without Flash Loader, software download by the user (phase 5 to phase 7) 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. 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].

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.

The Flash Loader only allows switching to User OS when a valid (i.e. MAC verified) image is loaded. The covered security functional requirements is FPT_FLS.1/Loader.

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.

In addition to the MPU and SAU, the TOE supports dynamical locking of dedicated peripherals depending on the secure and privilege security attributes. This way data flow between CPU and peripherals can be controlled. Manipulations utilizing access to specific peripherals can be restricted to the security state of the CPU or completely prevented.

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 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 and FPT_FLS.1

The TOE contains a dual CPU in lockstep mode to detect bit flips in digital logic. It also has strong EDCs with 28 bits or more code size. The covered security functional requirement is FPT_SDP.1.

The TOE provides a part (i.e. the SDP subsystem) which completely protects itself against modification attacks. The covered security functional requirement are FDP_ITT.1/SDP, FPT_ITT.1/SDP and FDP_IFC.1/SDP.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.

When the BOS hands over control to the embedded software, the SAU is disabled, and thus 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.

7.4 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.6/AES.

7.5 SF_CS: Crypto Suite Services

The optional Crypto Suite utilizes the symmetric coprocessor and the asymmetric coprocessor of the hardware to implement standard cryptographic algorithms.

For details, see the confidential Security Target.

8 Hash values of libraries

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

Table 88 SHA256 hash values

Lib	SHA256
UMSLC 02.01.0040	74091c50254bc348a48a2e261a125735dca41f2419cd9541c3e6fbfdff63b529
HSL	bff6fd92f692abfc3e440364056e1e9f8ec002b98b76975b01aaf7978762a895
Crypto Suite 5.02.003	CS-SLC22V31-sym-ae.lib: SHA256=134122e30585375620d567ee8b00a8f15b35131f0dc90d146cc46f828dde3567 CS-SLC22V31-sym-cipher-aes.lib: SHA256=80bd60d5f6039bd8f9e83005481feb9716ba15eb3af9ecaf329e1ddabf9be0f7 CS-SLC22V31-asym.lib: SHA256=3ee35cdbad40470baa96fbf0200532a6b53e2f219cbb88a6d9dbdc95fa587925 CS-SLC22V31-asym-base.lib: SHA256=16957e6c216898fac2163411e9e60fff975677c1415aa614eed007e48c69ea5f CS-SLC22V31-sym-mac-ciphermac-cbcmac.lib: SHA256=02c9b1453698329ccd8a6c27b31b6cc55592eda971c60e3952a8230ebf424b84 CS-SLC22V31-sym-ae-ccm.lib: SHA256=89cf32abbf8411f0ed7ca906a52b5cb5fa474860ff5d2b343ebe43580935f65b CS-SLC22V31-sym-cipher.lib: SHA256=b7bd5a87cb9c64096a6ee53adaa9c302c3483a7eb649b3a9b4d3003ae7babf92 CS-SLC22V31-sym-mac-ciphermac.lib: SHA256=f9aa85c66c4d2ca34baf7a60f68f513437fb411d68e42a30ec277f6c9ac31b48 CS-SLC22V31-sym-mac-ciphermac-cmac.lib: SHA256=3b2b5739c0d2df27830bcb45b9687854f10a52bea0c145570beb07b1cbd5929a CS-SLC22V31-core.lib: SHA256=d76d7ed0e797bb428ad6be9a9b569130c6558f52812ce95a1f5cf2534db620b9 CS-SLC22V31-asym-ecc-curve.lib: SHA256=16122c616883bdc565f3899f42b37482ea089b0e357bb9432329c0357201092a CS-SLC22V31-sym-cipher-des.lib: SHA256=df798c145d669e2ac2c5df4f7ad4c3c6e47ee0b3a10e940fc633c09c02813e82 CS-SLC22V31-rng-drbg.lib: SHA256=7416a297b30434c4178552889433c40126ac1026b19b1536dafbcdd957228337 CS-SLC22V31-asym-ecc.lib: SHA256=383df7e80689c163ef606651463f168d2710835bb58abe22d31ab1ec2df1b265 CS-SLC22V31-asym-ecc-ecdsa.lib: SHA256=c93e9e5c75e9b25554ad972d3a9e003ba6442796b02de0ce9987a022a44f8cff CS-SLC22V31-asym-ecc-ecsdsa.lib: SHA256=ec701cf3109d71373f78a2b768b2551ef75f10394c3b3c3ba06bb10fb849e0ff CS-SLC22V31-asym-ecc-eddsa.lib: SHA256=7af7c6929ae80ecc883baf8f740d4e464abc162c3fff52b1c4a2d9f71dce0a28 CS-SLC22V31-asym-ffc.lib: SHA256=2adc6ce4d2199d4763be7e160477fd00a49731c47f5fb4130a7e0bbcacf445fe

Hash values of libraries

Lib	SHA256
	CS-SLC22V31-sym-ae-gcm.lib: SHA256=1005cda8c8926ec7bab7f1c00a728c24dc25ce26466bab4c732c4156539bc579 CS-SLC22V31-sym-hash.lib: SHA256=77caae6de2551e972082e4ae87a715f966371e45b7e4809828894197530d8669 CS-SLC22V31-sym-mac-hashmac.lib: SHA256=2c1e7ce13d4bf10e26474ade8d1e7c8758b690336aa8a2eb03b4a7f8fcd6b935 CS-SLC22V31-sym-mac-hashmac-hmac.lib: SHA256=02f458da1a02c7bc19d734e971a3de76d7f85a6c0e12e49feb2addff9d59ec0e CS-SLC22V31-rng-hwrng.lib: SHA256=e8253fd84ee4f53bce1fa3aed07ed85a037fa78a8422af8630925b171da8f523 CS-SLC22V31-pqc-lattice.lib: SHA256=4c8ac4bac77905e6886f59daf8ba7e8f1a1c9bd564e4cb53ee632627eca6fec1 CS-SLC22V31-sym-mac.lib: SHA256=c7f3a182b33a35c188a606483379f7036759f94e3657399accc153620af621dc CS-SLC22V31-pqc-lattice-mldsa.lib: SHA256=0f9055cacf56f808879c533380c0083b1e2cd890ef9288c1fe8c7c467efcea91 CS-SLC22V31-pqc-lattice-mlkem.lib: SHA256=af0134daa7f11222c69303223f8341cfecc6487167ba3d3c14e938c25d30ed0e CS-SLC22V31-asym-ecc-curve-montgomerygfp.lib: SHA256=a186ff8199d2f22076c5b30901b123782194a23f92e17f6fac43b8fa42316c1c CS-SLC22V31-asym-ecc-pace.lib: SHA256=ac23192dff2be6c2f50b3e0ca20e2cd8927d4c627dac3ffaca2a9cf480380b1b CS-SLC22V31-pqc.lib: SHA256=62c931dc0a2b02c8777101118124630726310 2fc14b59f85a1148c4d5698230c CS-SLC22V31-sym-mac-ciphermac-retailmac.lib: SHA256=903a1c31ed77cdf4ad2e7813f0fb5e39b280e0f8ca408c2249c2f776e2280cd7 CS-SLC22V31-rng.lib: SHA256=88dca8f0d670563ffe888613a984d68d9c5ef5e73138cc7576a03072f8924097 CS-SLC22V31-asym-rsa.lib: SHA256=cb23a7395ba3d7f33ae341acdb9a8985c3502a2b829966b998a26149bde17423 CS-SLC22V31-sym-hash-sha1.lib: SHA256=78cbc14e181309bbf553c8d6c0de48856a93dac9ea209c2ce353b120e2a00421 CS-SLC22V31-sym-hash-sha2.lib: SHA256=70fd0f9f077eb8af67a28b419cd1acaa4ad387ca2987a5216662f160770d01aa CS-SLC22V31-sym-hash-sha3.lib: SHA256=08d6892dcd83b6a67166e9fd19f7d03a331050e65bb595675150188949c2dfa1 CS-SLC22V31-sym-hash-sha3-shake.lib: SHA256=62fc4fa99254d6389b8095afc0cda1ba323f74a068ac0d44007d37cae32d83db CS-SLC22V31-sym.lib: SHA256=f347a7854d76679f8c19ba43b3797dbdd1386db1c260bdedd860a714346cb291 CS-SLC22V31-asym-tlbx.lib: SHA256=2ce7e42d815584f4c2deda6099c24678be811075f3328d2f2338e4b8410dd5d6 CS-SLC22V31-asym-ecc-curve-twistededwardsgfp.lib: SHA256=a7291b346c713fb97b2d2f8e602dfda45f794414efb83c1f44287a3dcfa9fed0

Hash values of libraries

Lib	SHA256
	CS-SLC22V31-asym-ecc-curve-weierstrassgf2n.lib: SHA256=eb5db42386acf7a33108cfb2b5040b5de85ad112276dec8449f0de9bde50a00b CS-SLC22V31-asym-ecc-curve-weierstrassgfp.lib: SHA256=5b84b43350214b83267229445b26e2a754ed414a9bee9b53ad1258844f6e10d2

9 Cryptographic Table

The cryptographic table contains all cryptographic algorithms which are either used by the dedicated software or are provided by the hardware or the Crypto Suite. Please note that the Crypto Suite provides more flexibility in combining cryptographic primitives than what is allowed in the referenced standard. To be compliant to the standard, the user of the Crypto Suite is responsible for using the correct combination of primitives.

Table 89 Cryptographic table

Purpose	Cryptographic operation	Implementation	Key size in bits	Standards
Confidentiality	AES encryption and decryption in ECB mode	Hardware	128, 192, 256	[FIPS 197] [SP 800-38A]
Confidentiality	AES encryption and decryption in ECB, CBC, CTR mode	Crypto Suite	128, 192, 256	[FIPS 197] [SP 800-38A]
Integrity	AES CMAC mode	Crypto Suite	128, 192, 256	[FIPS 197] [SP 800-38B]
Integrity	AES CBC-MAC mode	Crypto Suite	128, 192, 256	[FIPS 197] [ISO 9797-1] padding method 2
Authenticated encryption	AES CCM	Crypto Suite	128,129,256	[FIPS 197] [SP 800-38C]
Authenticated encryption	AES GCM	Crypto Suite	128,192,256	[FIPS 197] [SP 800-38D]
Confidentiality	DES encryption and decryption in ECB, CBC, CTR mode	Crypto Suite	112, 168	[SP 800-67] [SP 800-38A]
Integrity	TDES Retail MAC	Crypto Suite	112	[SP 800-67] [ISO 9797-1]
Integrity	TDES CBC-MAC	Crypto Suite	112, 168	[SP 800-67] [ISO 9797-1] padding method 2
Integrity	HMAC with SHA-1, SHA256, SHA384, SHA512	Crypto Suite	160, 256, 384, 512	[FIPS 180-4] [FIPS 198-1]
Confidentiality	RSAEP RSA encryption	Crypto Suite	1024-4224	[PKCS#1], ch. 5.1.1 [SP 800-56B], ch. 7.1.1
Confidentiality	RSADP RSA decryption with exponential representation	Crypto Suite	1024-2112	[PKCS#1], ch. 5.1.2, 2a [SP 800-56B], ch. 7.1.2.1
Confidentiality	RSADP RSA decryption with CRT representation	Crypto Suite	1024-4224	[PKCS#1], ch. 5.1.2, 2b [SP 800-56B], ch. 7.1.2.3
Integrity	RSA signature generation RSASP1	Crypto Suite	1024-2112	[PKCS#1], ch. 5.2.1, 2a

Cryptographic Table

Purpose	Cryptographic operation	Implementation	Key size in bits	Standards
Integrity	RSA signature generation RSASP1 with CRT	Crypto Suite	1024-4224	[PKCS#1], ch. 5.2.1, 2b
Integrity	RSA signature verification RSAVP1	Crypto Suite	1024-4224	[PKCS#1], ch. 5.2.2
Key generation	generate probably random primes p and q for RSA keys	Crypto Suite	512-2064[1]	[FIPS 186-5], ch. A.1.3 w/o Step 1[2]
Key generation	generate RSA (N, d) parameters from p, q	Crypto Suite	1024-2112	[FIPS 186-5], ch. A.1.1 [PKCS#1], ch. 3.1, 3.2(1)
Key generation	generate RSA CRT parameters from p, q	Crypto Suite	1024-4224	[FIPS 186-5], ch. A.1.1 [PKCS#1], ch. 3.1, 3.2(2)
Primality test	Miller-Rabin primality test	Crypto Suite	512-20641	[FIPS 186-5], ch. B.3.1
Primality test	Enhanced Miller-Rabin primality test	Crypto Suite	512-20641	[FIPS 186-5], ch. B.3.2
Key generation	ECC key generation for ECDSA	Crypto Suite	160-521	[FIPS 186-5], ch. A.2.1
Key generation	ECC key generation for ECSDSA	Crypto Suite	160-521	[ISO 14888-3], ch. 6.10.3.
Key generation	ECC key generation for EdDSA	Crypto Suite	256, 456	[FIPS 186-5], ch. A.2.3
Integrity	ECDSA signature generation	Crypto Suite	160-521	[FIPS 186-5], ch. 6.4.1
Integrity	ECDSA signature verifications	Crypto Suite	160-521	[FIPS 186-5], ch. 6.4.2
Key agreement	ECDH key agreement	Crypto Suite	160-521	[SP 800-56A], ch. 5.7.1.2 w/o cofactor
Key agreement	PACE integrated mapping	Crypto Suite	160-521	[ICAO], Appendix B.2
Key agreement	X25519	Crypto Suite	256	[RFC 7748]
Key agreement	X448	Crypto Suite	448	[RFC 7748]
Integrity	ECSDSA signature generation	Crypto Suite	160-521	[ISO 14888-3], ch. 6.10.4 [TR-03111], ch. 4.2.3
Integrity	ECSDSA signature verification	Crypto Suite	160-521	[ISO 14888-3], ch. 6.10.5

[1] Size for each prime used in the RSA key. This means, the resulting RSA key can have size 1024 - 4128 bits.

[2] Prime generation is only conformant to [FIPS 186-5], A.1.3 for prime Bitlength >= 2048; in case prime Bitlength < 2048 identical algorithm is used, but considered proprietary

Cryptographic Table

Purpose	Cryptographic operation	Implementation	Key size in bits	Standards
				[TR-03111], ch. 4.2.3
Integrity	EdDSA signature generation	Crypto Suite	256, 456	[FIPS 186-5], ch. 7.6
Integrity	EdDSA signature verification	Crypto Suite	256, 456	[FIPS 186-5], ch. 7.7
Integrity	EdDSA pre-hash signature generation	Crypto Suite	256, 456	[FIPS 186-5], ch. 7.8.1
Integrity	EdDSA pre-hash signature verification	Crypto Suite	256, 456	[FIPS 186-5], ch. 7.8.2
Key agreement	Finite Field Diffie-Hellman	Crypto Suite	1024-2048	[SP 800-56A], ch. 5.7.1.1 w/o step 2
Hash	SHA-1 Hash digest	Crypto Suite	N/A	[FIPS 180-4]
Hash	SHA-2 Hash digest 224, 256, 384, 512, 512/224, 512/256 bits	Crypto Suite	N/A	[FIPS 180-4]
Hash	SHA3-224, SHA3-245, SHA3-384, SHA3-512	Crypto Suite	N/A	[FIPS 202]
Random	Physical RNG PTG.2	Hardware	N/A	[AIS 31]
Random	Hybrid Physical RNG PTG.3	Crypto Suite	128, 256	[AIS 31] [SP 800-90A], ch. 10.2
Random	Deterministic RNG DRG.3	Crypto Suite	128, 256	[AIS 20] [SP 800-90A], ch. 10.2
Random	Hybrid Deterministic RNG DRG.4	Crypto Suite	128, 256	[AIS 31] [SP 800-90A], ch. 10.2
XOF	SHAKE128, SHAKE256	Crypto Suite	N/A	[FIPS 202]
Authenticated encryption	AES CCM	Flash Loader	128	[SP 800-38C]
Key derivation	KDF in counter mode with AES CMAC as PRF	Flash Loader	128	[SP 800-108], ch. 4.1 [SP 800-38B], ch. 6.2
Integrity	AES CMAC mode	Flash Loader	128	[FIPS 197] [SP 800-38B]
Key generation	MLKEM key generation	ML-KEM-512, ML-KEM-768, ML-KEM-1024	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.1
Confidentiality	MLKEM Key encapsulation	ML-KEM-512, ML-KEM-768, ML-KEM-1024	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.2
Confidentiality	MLKEM Key decapsulation	ML-KEM-512, ML-KEM-768, ML-KEM-1024	ML-KEM-512, ML-KEM-768, ML-KEM-1024	[FIPS 203] ch. 6.3

Cryptographic Table

Purpose	Cryptographic operation	Implementation	Key size in bits	Standards
Key generation	MLDSA key generation	ML-DSA-44 ML-DSA-65 ML-DSA-87	ML-DSA-44 ML-DSA-65 ML-DSA-87	[FIPS 204] ch. 5.1
Integrity	MLDSA signature generation	ML-DSA-44 ML-DSA-65 ML-DSA-87	ML-DSA-44 ML-DSA-65 ML-DSA-87	[FIPS 204] ch. 5.3, ch.5.4
Integrity	MLDSA signature verification	ML-DSA-44 ML-DSA-65 ML-DSA-87	ML-DSA-44 ML-DSA-65 ML-DSA-87	[FIPS 204] ch. 5.2, ch.5.4

10 Evaluation Methodology for ATE_SDP.1

10.1 Evaluation sub-activity (ATE_SDP.1)

10.1.1 Objectives

The context of this evaluation method is the evaluation of security ICs that employs SDP functionality. This evaluation method specifically applies to the corresponding SAR ATE_SDP.1.

10.1.2 Input

The evaluation evidence for this sub-activity is:

a) the ST
b) the security architecture specification
c) the test documentation
d) the test environment

10.1.3 Action ATE_SDP.1.1E

ATE_SDP.1.1C: The test documentation shall contain one or more SDP-testing netlists which cover the SDP subsystem.

Work unit ATE_SDP.1-1: The evaluator ***shall check*** that the test documentation contains one or more SDP-testing netlists covering the SDP subsystem.
The set of provided SDP-testing netlists shall contain the sequential logic of the SDP subsystem as present in the TOE's product netlist.

ATE_SDP.1.2C: The test documentation shall contain a rationale demonstrating that the SDP-testing netlists are adequate design representations of the TOE's product netlist. Adequate means that

- the provided SDP-testing netlists may be partly or fully independent of any product cell libraries.
- logic inferred by electronic design automation (EDA), like clock trees and clock gates, reset trees, design for testability (DfT) logic, and power control logic may be omitted.
- logic of memory cells may be replaced by functional equivalent mock-ups.

Work unit ATE_SDP.1-2: The evaluator ***shall examine*** the test documentation to determine that the rationale demonstrates that each SDP-testing netlist contains all relevant sequential cells of the TOE's product netlist.
For each sequential cell in the TOE's product netlist, which belongs to the SDP subsystem, there shall be an equivalent counterpart in at least one SDP-testing netlist. An SDP-testing netlist may in total contain more sequential cells than the TOE's product netlist.
Latches may be replaced by replacement circuits in an SDP-testing netlist, consisting of a register and multiplexer to represent the functionality of both phases, storage phase and transparent phase.
Where necessary, clock gates may be replaced by functional clock gating represented by logic that choses between capturing a new value and storing the old value.
CPU and SDP protected peripherals shall be subject to fault injection together with all connections between those components.
Memory arrays may be replaced by equivalent replacement circuits (mock-ups) and/or may be reduced in size but only to a degree that the corresponding memory control logic can be stimulated qualitatively the same as in the product netlist.

ATE_SDP.1.3C: The test documentation shall contain the test stimulus specifications and expected results.

Work unit ATE_SDP.1-3: The evaluator ***shall check*** that the test documentation contains the test stimulus specifications and expected results. The evaluator may make their decision based on the test procedure descriptions in the test documentation or based on automated test records created during SDP testing (concerning the necessary stimulation, test programs used are of particular importance).

ATE_SDP.1.4C: The SDP testing shall be considered as pass if the fault injection is detected before it causes exploitable failures.

Work unit ATE_SDP.1-4: The evaluator ***shall examine*** the test documentation to determine that the fault injection is detected before it causes exploitable failures.

A functional deviation shall be determined by comparison of fault-free primary output values of the SDP subsystem and corresponding primary output values observed and recorded during SDP testing. The functional deviation must be documented as part of each expected SPD-testing result. A timing deviation due to the fault injection may be tolerated and does not imply a functional deviation.

An actual SDP-testing result shall be considered not as expected, i.e., a fail result, if the fault injection caused a functional deviation while the TOE did not preserve a secure state. Otherwise, an actual SDP-testing result shall be considered as expected, i.e., a pass result.

ATE_SDP.1.5C: The test documentation shall contain a rationale demonstrating that all sequential cells contained in CPU and protected peripherals are covered by each test stimulus.

Work unit ATE_SDP.1-5: The evaluator ***shall examine*** the test documentation to determine if the SDP-testing approach ensures that faults are injected in all sequential cells contained in the SDP-testing netlists. Logic whose outputs are exclusively connected to unprotected hardware components may be omitted.
The memory cells and logic inferred by electronic design automation (EDA) are not subject to fault simulation/emulation testing. For memory cells the integrity protection required by FDP_SDP.1.2 shall be verified according to standard evaluation procedures (e.g. by design review and mathematical arguments showing the claimed EDC strength).
The developer may document the approach of how to make sure that each sequential cell is contained in at least one SDP-testing netlist and are subject to fault-injection during SDP testing.
The evaluator may make his decision based on the documented approach, and whether it is convincing about the fault-injection coverage of sequential cells.

ATE_SDP.1.6C: The test documentation shall contain a mapping demonstrating that all security relevant functions of the SDP subsystem are covered by fault injection tests.

Work unit ATE_SDP.1-6: The evaluator ***shall examine*** the test documentation and the security architecture description to determine that all SFR-enforcing and SFR-supporting modules which belong to the CPU and protected peripherals of the SDP subsystem are covered by at least one fault detection test.

11 Acronyms

Acronym	Description
AES	Advanced Encryption Standard
CSP	Chip Scale Package
DC	Data Cache
ECC	Elliptic Curve Cryptography or Error Correction Code
EDC	Error Detection Code
FFC	Finite Field Cryptography
FRS	Fast Random Source
IC	Instruction Cache
ISS	Instruction Stream Signature
MISE	Masked Instruction Set Extension
MCICE	Memory Cache Confidentiality Integrity Engine
ML	Module Lattice
MLDSA	Module Lattice Digital Signature Algorithm
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
SDP	Self-Secured Digital Protection
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
XOF	eXtensible Output Function

12 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
[ANSSI]	ANSSI: "Avis relatif aux paramètres de courbes elliptiques définis par l'Etat français" in: Journal Officiel de la République Française (JORF)
[Armv8-M]	Arm® v8-M Architecture Reference Manual, ARM part number: AR100-DA-78000-r0p1-10eac0
[CC1]	Common Criteria for Information Technology Security Evaluation, CC:2022, revision 1, November 2022 — Part 1: Introduction and general model
[CC2]	Common Criteria for Information Technology Security Evaluation, CC:2022, revision 1, November 2022 — Part 2: Security functional components
[CC3]	Common Criteria for Information Technology Security Evaluation, CC:2022, revision 1, November 2022 — Part 3: Security assurance components
[CC5]	Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements, November 2022, CC:2022, Revision 1
[FIPS 180-4]	NIST: FIPS publication 180-4: Secure Hash Standard (SHS), August 2015
[FIPS 186-5]	NIST: FIPS publication 186-5: Digital Signature Standard (DSS), February 3, 2023
[FIPS 197]	Federal Information Processing Standards Publication, Advanced Encryption Standard (AES), FIPS PUB 197, as of 05/09/23
[FIPS 198-1]	Federal Information Processing Standards Publication, FIPS PUB 198-1, July 2008
[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
[FIPS 204]	Module-Lattice-Based Digital Signature Standard, FIPS 204, August 13, 2024
[ICAO]	ICAO Doc 9303, Machine Readable Travel Document, 8th edition, 2021, Part 11: Security Mechanisms for MRTDs
[ISO 15946]	ISO/IEC 15946-5:2022

References

[ISO 9797-1]	ISO/IEC 9797-1: 2011 - Information Technology - Security techniques - Message Authentication Codes - Part 1: Mechanisms using block cipher
[ISO 9798-2]	ISO/IEC 9798-2:2019 - Information Technology - Security techniques - Entity authentication - Part 2: Mechanisms using authenticated encryption. Fourth edition 2019-06
[ISO 14888-3]	ISO/IEC 14888-3:2018 IT Security techniques Digital signatures with appendix Part 3: Discrete logarithm based mechanisms
[JIL]	Joint Interpretation Library, Application of Attack Potential to Smartcards, Version 3.2.1 February 2024
[PKCS#1]	PKCS #1: RSA Cryptography Standard, v2.2, October 27, 2012, RSA Laboratories
[PP0084]	Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, 13.01.2014, BSI-CC-PP-0084-2014
[RFC 5639]	IETF: RFC 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, March 2010, http://www.ietf.org/rfc/rfc5639.txt
[RFC 5114]	IETF: RFC 5114, Additional Diffie-Hellman Groups for Use with IETF Standards, January 2008
[RFC 3526]	RFC 3526, More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), May 2003
[RFC 7748]	Internet Research Task Force (IRTF), Request for Comments: 7748, January 2016
[SEC2]	SEC 2: Recommended Elliptic Curve Domain Parameters Certicom Research, January 27, 2010, Version 2.0
[SP 800-108]	National Institute of Standards and Technology (NIST), Recommendation for Key Derivation Using Pseudorandom Functions, NIST SP 800-108r1, August 2022
[SP 800-186]	NIST SP 800-186 Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters, February 2023
[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 Commerce, Special Publication 800-38A, Edition 2001, Recommendation for Block Cipher Modes of Operation: Methods and Techniques
[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 800-38D]	NIST SP 800-38D, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007

References

[SP 800-56A]	National Institute of Standards and Technology (NIST), Special Publication 800-56A, Revision 3, April 2018
[SP 800-56B]	National Institute of Standards and Technology (NIST), Special Publication 800-56B, Revision 2, March 2019
[SP 800-67]	NIST SP 800-67 Rev. 2, Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[SP 800-90A]	NIST Special Publication 800-90A Revision 1 Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015
[TR-03111]	Technical Guideline BSI TR-03111 Elliptic Curve Cryptography Version 2.10, 2018-06-01
[CCErrata]	Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), V1.1, 2024-07-22
[CCTrans]	CCMC-2023-04-001, Transition Policy to CC:2022 and CEM:2022, 2023-04-20
[PDCL]	Joint Interpretation Library, Security requirements for post-delivery code loading, Version 2.0, September 2024

Revision history

Version	Date	Description
Rev 1.0	2025-10-31	Release Version

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

Edition 2025-10-31

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about this document?

Email:
csscustomerservice@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).

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.