Security Target for Trusted Platform Module 2.0

NS350 v30

Version 1.8

Date：2024-07-09

PUBLIC

REVISION HISTORY

Version	Date	Modification	Author
1.0	22 March, 2023	First Release	Xin Liu, Jin Jia
1.1	12 June, 2023	Update life cycle description	Xin Liu, Jin Jia
1.2	03 July, 2023	Includes evaluation comments	Xin Liu, Jin Jia
1.3	29 April, 2024	Includes evaluation comments	Xin Liu, Jin Jia
1.4	29 May, 2024	Includes evaluation comments	Xin Liu, Jin Jia
1.5	17 June, 2024	Includes evaluation comments	Xin Liu, Jin Jia
1.6	02 July, 2024	Includes evaluation comments	Xin Liu, Jin Jia
1.7	05 July, 2024	Includes evaluation comments	Xin Liu, Jin Jia
1.8	09 July, 2024	Includes evaluation comments	Xin Liu, Jin Jia

Contents

1 Security Target Introduction (ASE_INT)

This section contains the necessary information to identify the Security Target (ST). This information may be used to cross-reference this document.

1.1 ST Reference

This security target is referenced with the following information:

- Filename: Security Target for Trusted Platform Module 2.0 NS350 v30
- Revision: v1.8
- Internal documentation reference: NS350-DT-D016
- Date: 09 July, 2024

This security target is strictly conformant to the Protection Profile PC Client Specific Trusted Platform Module Specification Family 2.0 Level 0 Revision 1.59, Version 1.3, [ANSSI-CC-PP-2021/02]

Table 1: Identification

	Version	Date	Registration
Security Target	1.8	July 09, 2024	Security Target for Trusted Platform Module 2.0 NS350 v30
Target of Evaluation	30.30.9220.9488	June 24, 2024	Trusted Platform Module 2.0 NS350 in the delivery format as defined in Section 2.3.4
Protection Profile	Version 1.3	September 29, 2021	Protection Profile PC Client Specific Trusted Platform Module Specification Family “2.0” Level 0 Revision 1.59 ANSSI-CC-PP-2021/02
Guidance documentation	Revision 01.59	November 8, 2019	Trusted Platform Module Library Part 1: Architecture Family “2.0” Level 00 Revision 01.59
	Revision 01.59	November 8, 2019	Trusted Platform Module Library Part 2: Structures Family “2.0” Level 00 Revision 01.59
	Revision 01.59	November 8, 2019	Trusted Platform Module Library Part 3: Commands Family “2.0” Level 00 Revision 01.59 Trusted Platform Module Library
	Revision 01.59	November 8, 2019	Trusted Platform Module Library Part 4: Supporting Routines Family “2.0” Level 00 Revision 01.59

	Version 1.4	January 9, 2023	Errata for TCG Trusted Platform Module Library, Family “2.0” Level 00 Revision 01.59
	Version 1.05 Revision 14	September 4, 2020	TCG PC Client Platform TPM Profile (PTP) for TPM 2.0 Version 1.05 Revision 14
	Version 1.1	November 8, 2021	Errata for PC Client Platform TPM Profile (PTP) for TPM 2.0 Version 1.05 Revision 14
	Revision 2	January 26, 2022	TCG EK Credential Profile for TPM Family 2.0; Level 0 Version 2.5
	Version 1.06	December 4, 2023	TCG PC Client Platform Firmware Profile Specification, Level 0, Version 1.06 Revision 52, December 4, 2023.
	Version 1.0	March 15, 2017	TCG TPM v2.0 Provisioning Guidance, Version 1.0, Revision 1.0, March 15, 2017
	Revision 1.10	May 20, 2024	NS350 TPM 2.0 Datasheet for V30.30.9220.9488
	Version 1.4	July 09, 2024	AGD_OPE
	Version 1.3	July 09, 2024	AGD_PRE
	3.1 Revision 5	April 2017	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2017-04-001 Part 2: Security functional requirements CCMB-2017-04-002 Part 3: Security Assurance Components CCMB-2017-04-003

2 TOE Description

2.1 TOE Reference

The chip packaging is not included in the TOE.

The TOE is a complete solution for Trusted Platform Module 2.0 NS350 v30 with Hardware version 1C, version NS350 v30.30.9220.9488.

2.2 TOE Overview

The security target (ST) describes the target of evaluation (TOE) named Trusted Platform Module 2.0 NS350 v30 and provides a product summary. In the following sections of this document the expressions NS350 or TPM stands for all forms of the TOE.

The NS350 is a single integrated circuit that implements the functions defined in the TCG Trusted Platform Module Library Family “2.0” Level 00 Revision 01.59, and the TCG PC Client Platform TPM Profile (PTP) for TPM 2.0 Version 1.05 Revision 14. The TCG Trusted Platform Module Library specification describes the design principles, the TPM structures, the commands and supporting routines for the commands. The PC Client Specific Platform TPM Profile for TPM 2.0 specification describes the additional features that must be implemented by a TPM for a PC Client platform.

NS350 is also compliant with the PC Client Specific Platform TPM Profile for TPM 2.0 for communication interface to leverage the drivers and software stack.

The TOE consists of TPM hardware, TPM firmware and TPM guidance documentation. The non-TOE component is described in Chapter 3.1.3 of Protection Profile [5].

2.2.1 TOE Usage and Security Features

The TPM library specification describes the TPM protections in terms of Protected Capabilities and Protected Objects. A Protected Capability is an operation that must be performed correctly for a TPM to be trusted and therefore is in the scope of the CC evaluation as part of the TOE security functionality (TSF). A Protected Object is data that must be protected for a TPM operation to be trusted. The TSF performs all operations with Protected Objects inside the TPM. The TSF protects the confidentiality of Protected Objects when exported from the TPM and checks the integrity of Protected objects when imported into the TPM. The TOE provides physical protection for Protected Objects residing in the TPM.

The TPM provides methods for collecting and reporting identities of hardware and software components of a computer system platform. The computer system report generated by the trusted computing base (TCB) the TPM is part of allows determination of expected behavior and from that expectation of trust in the computer system platform.

There are commonly three Roots of Trust in a trusted platform: a root of trust for measurement (RTM), root of trust for reporting (RTR) and root of trust for storage (RTS). In

TCG systems roots of trust are components that must be trusted because misbehavior might not be detected. The RTM is a computing engine capable of making inherently reliable integrity measurements and maintaining an accurate summary of values of integrity digests and the sequence of digests. The RTR is a computing engine capable of reliably reporting information held by the RTM. The RTS provides secure storage for a practically unlimited number of private keys or other data by means of exporting and importing encrypted data.

Support for the Root of Trust for Measurement

The TPM supports the integrity measurement of the trusted platform by calculation and reporting of measurement digests of measured values. Typically, the RTM is controlled by the Core Root of Trust for Measurement (CRTM) as the starting point of the measurement. The measurement values are representations of embedded data or program code scanned and provided to the TPM by the measurement agent. The TPM supports cryptographic hashing of measured values and calculates the measurement digest by extending the value of a PCR with a calculated or provided hash value. The PCRs are shielded locations of the TPM which can be reset by TPM reset or a trusted process, written only through measurement digest extensions and read.

Root of Trust for Reporting

The EK and the corresponding Endorsement Certificates define the trusted platform identities for RTR. The NS350 is shipped with EK and a Certificate of the Authenticity of this EK. The EK is bound to the Platform via Platform Certificate, providing assurance from the certification body of the physical binding and connection through a trusted path between the platform (the RTM) and the genuine TPM (the RTR). The attestation of the EK and the Platform Certificates build the base for attestation of other keys and measurements.

Root of Trust for Storage

The TPM holds the Storage Primary Seed (SPS) and generates Storage Root Keys (SRK) from SPS. The SRK are roots of Protected Storage Hierarchies associated with a TPM. The storage keys in these hierarchies are used for symmetric encryption and signing of other keys and data together with their security attributes. The resulting encrypted file, which contains header information in addition to the data or the key, is called a BLOB (Binary Large Object) and is output by the TPM and can be loaded in the TPM when needed. The private keys generated on the TPM can be stored outside the TPM (encrypted) in a way that allows the TPM to use them later without ever exposing such keys in the clear outside the TPM. The TPM uses symmetric cryptographic algorithms to encrypt data and keys and may implement cryptographic algorithms of equivalent strength.

Platform Key Hierarchy

The TPM may hold an additional Platform Primary Seed (PPS) and generate Platform Keys from PPS. The platform key hierarchy is controlled by the Platform Firmware. The PPS is generated by the TOE.

Other Security Services and Features

- Random number generator (NRBG/DRBG, NRBG denotes entropy source based on a hardware physical source, health test and conditioning component conforms to NIST SP800-90B and the entropy source output bits are used as input to a DRBG algorithm (NIST SP800-90A) based on CTR_DRBG with AES-256 and derivation function using a block cipher algorithm.

- Asymmetric key generation (RSA keys with key length 2048 , 3072 and 4096 bits, ECDSA keys with key length 256 and 384 bits of curve TPM_ECC_NIST_P256 and TPM_ECC_NIST_P384)
- Symmetric key generation (AES keys with 128 and 256 bits, not open to user)
- Asymmetric key functions (RSA algorithm supports secret sharing, RSA algorithm according to RSAES-PKCS1-v1_5 and RSAES-OAEP for encryption and decryption, RSA algorithm according to RSASSA-PSS and RSASSA-PKCS1-v1_5 for signature and verification, ECDSA algorithm supports secret sharing, signature and verification)
- Symmetric key functions (AES supports encryption and decryption with CFB mode, HAMC-SHA1, HAMC-SHA256 and HAMC-SHA384 support symmetric signature and signature verify)
- HMAC function for symmetric signing and signature verification
- Hash functions (SHA1, SHA256 and SHA384)
- Key Derivations (KDFa and KDFe)
- Secure storage (shielded location) and data transmission(encryption)
- Identification and Authorization mechanisms
- Startup self-tests and conditional self-tests by known-answer test or pair-wise consistency test for cryptographic algorithm
- Physical protection

The TPM stores persistent state associated with the TPM in NV memory and provides NV memory as a shielded location for data of external entities. The platform and entities authorized by the TPM owner controls allocation and use of the provided NV memory. The access control may include the need for authentication of the user, delegations, PCR values and other controls.

The TOE generates two types of keys: Ordinary keys are generated using the random number generator to seed the key computation. Primary Keys are derived from a Primary Seed and key parameters by means of a key derivation function. Derived Keys are derived from the sensitive value of the parent and key parameters by means of a key derivation function.

The Endorsement Key (EK) and associated EK certificate (EK credential) are stored in the TPM during the manufacturing process at the TOE lifecycle phase “Manufacturing”.

Each TOE supports Endorsement keys:

- One 2048-bit RSA key pair
- One 3072-bit RSA key pair
- One 4096-bit RSA key pair
- One 256-bit ECC key pair generated with curve TPM_ECC_NIST_P256
- One 384-bit ECC key pair generated with curve TPM_ECC_NIST_P384

Each Endorsement key is generated in TOE inside.

The Endorsement Key certificate is generated on the Vendor controlled certificate server. Every certificate is stored in TOE inside. The EKs are certified by the intermediate CA keys. The intermediate CA keys are under control by Vendor.

All EK certificates comply with the templates defined in the TCG EK Credential Profile for TPM Family 2.0; Level 0 Version 2.5.

The EK certificate generation and importation infrastructure is located within the secure production area of the TOE.

2.3 TOE Description

2.3.1 TOE Hardware Description

The TOE of the NS350's hardware is a security controller. Basic structure of the TOE hardware is shown in figure 1 below:

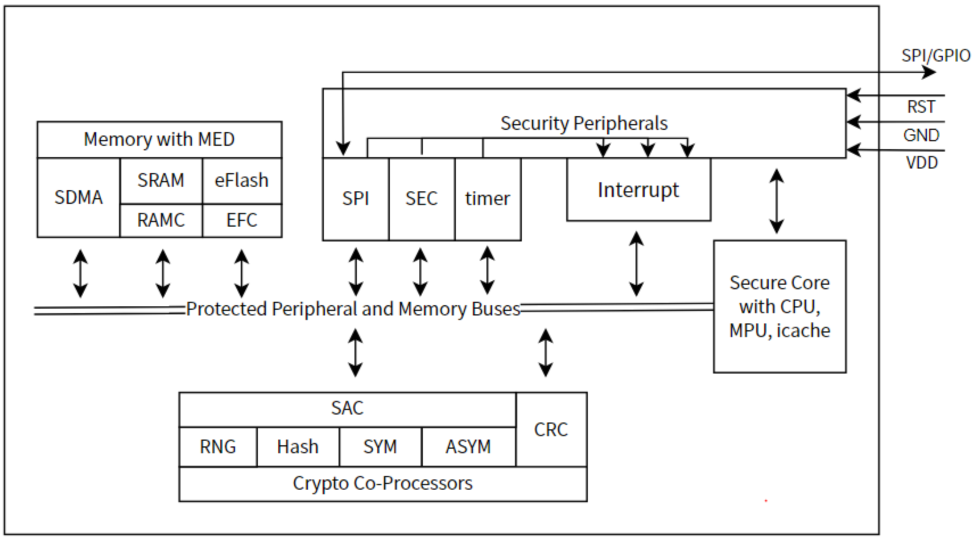

Figure 1: Hardware Block Diagram for NS350

Basic components of the TOE hardware are shown below:

(1) Execution Secure Core

- 32 - bit high performance and low power Security CPU, AMBA architecture, Low power mode: Sleep and Deep Sleep mode
- MPU - Memory Permission Unit, Permission management of FLASH, SRAM and other components
- icache - Instruction Cache, 1KB, to improve the efficiency for the system

(2) Coprocessors

- SAC - Secure Algorithm Co-processor, an integration module of algorithms
- SYM - Symmetric Algorithm Engine for AES
- ASYM - Asymmetric Algorithm Engine, supports RSA 2048/3072/4096 and ECC 256/384
- Hash Algorithm Engine, support SHA-1, SHA-256 and SHA384
- Checksum Module (CRC 16bits)
- Random number generator (RNG)

(3) TOE Memory

- RAMC - Random Access Memory Controller, the inverter of AHB bus and SRAM bus
- SDMA - Secure Direct Memory Access, to move data from/to ARAM(SRAM for algorithms) and can ensure the data security during data moving
- SRAM - Static Random Access Memory, support memory for algorithms
- EFC - Embedded Flash Controller, to generate flash operations including erase/program/read and security control
- eFlash - Embedded Flash, support non-volatile memory
- MED - Memory Encrypt and Decrypt, the data on the bus and data stored in memory are encrypted to ensure data security

(4) Security Peripheral

- SPI - communicate with host using SPI protocol
- SEC - Secure Environment Control Unit, the chip digital system security detection and alarm processing module
- Timer- 4-channel timers, can generate interrupt from time to point and support wake-up function in sleep mode
- Interrupt - completes the management of chip interrupt, using vectorization interrupt, 4 interrupt priorities can be programmed
- Security I/O manager – manager I/O alternation relation, pad control (include pull-down, pull-up, output, input) , and can config GPIO mode or alternative function mode, can remap to some peripheral device, communicate with outside of chip

2.3.2 TOE Firmware Description

The TOE firmware includes the two compiled blocks

- Boot Firmware is an immutable area used to complete chip hardware platform initialization, check TPM firmware integrity and load the TPM firmware, or enter the chip's upgrade/restore mode to upgrade or restore TPM Firmware.
- TPM Firmware is a mutable area that completes the acceptance, processing and response of the TPM 2.0 commands supported by the NS350. Provide security services for the entire chip.

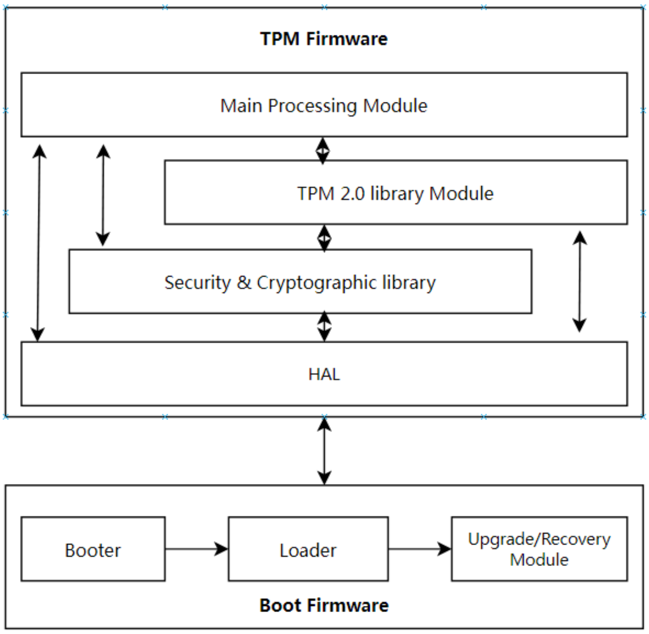

Figure 2: NS350 firmware structure

The TOE firmware consists of several modules:

- Boot: initial the hardware resources;
- Loader: call TPM initial and startup or upgrade;
- Upgrade/Recovery Module: Decrypt the upgraded firmware Block and upgraded that in eflash
- Hardware Abstraction layer (HAL): TPM hardware resources abstraction interface for processing TPM commands, including cryptolib
- Security & Cryptographic Library: Cryptographic Algorithms and general security function Library.
- TPM 2.0 Library Module: The code implementation for supported TPM 2.0 functions
- Main Processing Module: TPM firmware mainstream process and management, including receiving/processing/response all host required.

2.3.3 TOE Guidance Documentation

The guidance documentation consists of a set of information containing the description of all interfaces to operate the TOE. The list of the guidance documentation is given in Table 1.

2.3.4 Forms of delivery

The TOE is delivered in form of complete chips which include the hardware, the firmware, the Endorsement Primary Keys and certificates, and the guidance documentation. The TOE is finished and the extended test features are removed. The TOE is delivered in different package (e.g. QFN). The ordering codes are listed in the document NS350 v30 TPM 2.0 Datasheet for v30.30.9220.9488.

2.3.5 Life Cycle Description

The life cycle of the TOE as part of this evaluation includes

- phase 1 “development” and
- phase 2 “Manufacturing and delivery”

as defined in the PP [5], section 3.1.4 'TPM Life Cycle'.

The Phase 1 that includes TPM hardware and firmware development involves the sites of

- Development - Shenzhen, China

The Phase 2 that includes the die manufacturing and the EK and EK certificate injections involves the sites of

- Silicon production - Tainan, Taiwan
- Test manufacturing and EK/EK certificate injection - Chungli, Taiwan
  & Suzhou, China

3 Conformance Claim (ASE_CCL)

3.1 CC Conformance Claim

This Security Target (ST) and the TOE claim conformance to the Common Criteria version 3.1, Release 5 Part 1 [1], part 2 [2] and part 3 [3].

Conformance of this ST is claimed for Common criteria Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in Protection Profile.

3.2 PP Claim

This Security Target is in **strict conformance** to the Protection Profile PC Client Specific Trusted Platform Module Specification Family 2.0 Level 0 Revision 1.59, Version 1.3, released by Trusted Computing Group dated 29 September 2021.

The protection profile is registered and certified by the "Agence Nationale de la Sécurité des Systemes d´information" (ANSSI) under the reference [ANSSI-CC-PP-2021/02, dated 2021-11-30].

3.3 Package Claim

This Security Target does not claim conformance to a package of the PP [5]. The assurance level for this TOE is EAL4 augmented with ALC_FLR.1 and AVA_VAN.4 defined in CC part 3 [3].

3.4 Conformance Claim Rationale

This Security Target claims strict conformance to the PP [5].

The Target of Evaluation (TOE) is a complete solution implementing the TCG Trusted Platform Module main specification Version 2.0, Level 0 Revision 1.59 ([6], [7], [8] and [9]) and the TCG PC Client Specific Platform TPM Profile (PTP) Specification [11], so the TOE is consistent with the TOE type defined in the PP [5].

The **security problem** definition of this security target is consistent with the statement of the security problem definition of the PP [5], as the security target claims strict conformance to the PP [5] and no other threats, organisational security policies and assumptions are added.

The **security objectives** of this security target are consistent with the statement of the security objectives of the PP [5], as the security target claimed strict conformance to the PP [5]. In order to align with ANSSI Application Note 6 [28], three new security objectives mentioned in Note 6 have been added. The new security objectives do not cause any disruption to the existing security objectives in the PP [5].

The **security requirements** of this security target are consistent with the statement of the security requirements of the PP [5], as the security target claimed strict conformance to the PP [5]. All assignments, selections and iterations of the security functional requirements are done in the PP [5] and in this security target at section 7.2.

4 Security Problem Definition (ASE_SPD)

The content of the PP [5] applies to this chapter completely.

4.1 Assets

The assets of the TOE are defined in the PP [5], section 5.1, Assets. No other assets are added. These assets have to be protected while being executed as well as when the TOE is not in operation.

4.2 Threats

The threats of security are defined in the PP [5], section 5.2, Threats. No other threats are added.

4.2.1 Compliance to ANSSI Note 6

The threats in additional code loading described in ANSSI Note 6 can be categorized into the existing threats specified in PP.

4.3 Organizational Security Policies

The organizational security policies are defined in the PP [5], section 5.3, Organizational Security Policies (OSPs). No other OSPs are added.

4.4 Assumptions

The TOE environment is highly variable. In general, the TOE is assumed to be in an uncontrolled environment with no guarantee of the TOE's physical security.

The assumptions of this Security Target are defined in the PP [5], section 5.4, Assumptions. No other assumptions are added.

5 Security Objectives (ASE_OBJ)

This section shows the security objectives which are relevant for the TOE. For this section the PP [5] can be applied completely.

5.1 Security Objectives for the TOE

The security objectives of the TOE are defined and described in the PP [5], section 6.1.

5.2 Security Objectives for the Operational Environment

The security objectives for the operational environment of this TOE are described in the PP [5], section 6.2, Security Objectives for the Operational Environment, no other security objectives for the operational environment are added.

5.3 Security Objectives Rationale

The security objectives rationale described in the PP [5], section 6.3, Security Objective Rationale remains fully valid.

6 Extended Components Definition (ASE_ECD)

The extended component “FCS_RNG Generation of random numbers” is defined in the PP [5], section 7.2. No other extended component definitions are added in this security target.

7 Security Requirements (ASE_REQ)

7.1 Security Functional Requirements Listed by the TPM 2.0 Protection Profile

The security functional requirements (SFRs) for the TOE are defined in the PP [5] section 8.1 and chapter 9. All the assignments and selections of the Security Functional Requirements are done in the PP with the exception of the following SFRs that required to be completed in the security target. The operations completed in the ST are marked in *italic* font.

7.2 Security Functional Requirements for the TOE

FMT_MSA.2 Secure security attributes

Hierarchical to:	No other components
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for: *security attributes of keys, PCR, NV storage areas, monotonic counters and field upgrade data.*

FCS_CKM.1/PKRSA Cryptographic key generation (RSA primary keys)

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/PKRSA The TSF shall generate cryptographic **primary *RSA*** keys in accordance with a specified cryptographic key generation algorithm *RSA key generator* and specified cryptographic key sizes *2048*, *3072 and 4096 bits* that meet the following: *TPM library specification [6], [7], [8], in combination with RFA 3447 [27] and FIPS 186-4 [14].*

FCS_CKM.1/PKECC Cryptographic key generation (ECC primary keys)

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/PKECC The TSF shall generate cryptographic **primary *ECC*** keys in accordance with a specified cryptographic key generation algorithm *ECC key generator* and specified cryptographic key sizes *256 and 384 bits* that meet the following: *TPM library specification [6], [7], [8]*, and

ECC key generation:

1. *According to ISO/IEC 15946-1 [24] section 8.2 "Elliptic curve key generation" with curves*

- *ECC_NIST_P256*
- *ECC_NIST_P384*

FCS_CKM.1/PKSYM Cryptographic key generation (SYM primary keys)

Hierarchical to: No other components.

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

FCS_CKM.1.1/PKSYM The TSF shall generate cryptographic **primary *symmetric*** keys in accordance with a specified cryptographic key generation algorithm *AES key generator* and specified cryptographic key sizes *128 bits and 256 bits* that meet the following: *TPM library specification [6], [7], [8], and*

AES key generation:

1. *The AES key is a 128 and 256 bit random number is recommended according to NIST SP800-133 [18] section 6;*
2. *and using key Derivation function as the pseudorandom Functions according to NIST SP800-133 [18] section 5.*

FCS_CKM.1/RSA Cryptographic key generation (RSA keys)

Hierarchical to: No other components

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

FCS_CKM.1.1/RSA The TSF shall generate cryptographic ***RSA*** keys in accordance a specified cryptographic key generation algorithm *RSA key generator* and specified cryptographic key size *2048*, *3072 and 4096* bits *that meet the following: TPM library specification [6] [7] [8], and RFA 3447 [27] and FIPS 186-4 [14].*

FCS_CKM.1/ECC Cryptographic key generation (ECC keys)

Hierarchical to: No other components.

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

FCS_CKM.1.1/ECC The TSF shall generate cryptographic **ECC** keys in accordance with a specified cryptographic key generation algorithm *ECC key generator* and specified cryptographic key sizes *256 and 384 bits* that meet the following: TPM library specification [6], [7], [8], and

ECC key generation:

1. *According to ISO/IEC 15946-1 [24] section 8.2 "Elliptic curve key generation" with curves*

- *ECC_NIST_P256*
- *ECC_NIST_P384*

FCS_CKM.1/SYMM Cryptographic key generation (symmetric keys)

Hierarchical to: No other components.

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

FCS_CKM.1.1/SYMM The TSF shall generate cryptographic **symmetric** keys in accordance with a specified cryptographic key generation algorithm *AES key generator* and specified cryptographic key sizes *128 and 256 bits* that meet the following: *TPM library specification [6], [7], [8], and NIST Special Publication 800-133 [18].*

FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

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

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method: *key zeroise method that meets the following:*

according to ISO/IEC 19790:2012 [26] section 7.9.7 "Sensitive security parameter zeroisation"

FCS_COP.1/AES Cryptographic operation (symmetric encryption/decryption)

Hierarchical to: No other components.

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

FCS_COP.1.1/AES The TSF shall perform *symmetric encryption and decryption* in accordance with a specified cryptographic algorithm *AES* in the mode *CFB* , and cryptographic key sizes *128 and 256 bits* that meet the following: *[SP800-38A][16] or [ISO 10116:2006][22] or [ISO 18033-3][25].*

FCS_COP.1/SHA **Cryptographic operation (hash function)**

Hierarchical to: No other components

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

FCS_COP.1.1/SHA The TSF shall perform hash value calculation in accordance with a specified cryptographic algorithm *SHA-1, SHA-256 and SHA-384* and cryptographic key sizes ***none*** that meet the following: *FIPS 180-4[13].*

FCS_COP.1/HMAC **Cryptographic operation (HMAC calculation)**

Hierarchical to: No other components

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

FCS_COP.1.1/HMAC The TSF shall perform HMAC value generation and verification in accordance with a specified cryptographic algorithm *HMAC with SHA-1, SHA-256 and SHA384* and cryptographic key sizes 160 bits, *256 bits and 384 bits* that meet the following: *FIPS 198-1 [15] or ISO/IEC 9797-2 [21].*

FCS_COP.1/RSAED **Cryptographic operation (asymmetric encryption/decryption)**

Hierarchical to: No other components

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

FCS_COP.1.1/RSAED The TSF shall perform *asymmetric encryption and decryption* in accordance with a specified cryptographic algorithm *RSA without padding,* RSAES-PKCS1-v1_5, RSAES-OAEP and cryptographic key sizes *2048 bits, 3072 bits and 4096 bits* that meet the following: *RFC3447 PKCS#1 v2.1 [27].*

FCS_COP.1/RSASign: **Cryptographic operation (RSA signature generation/verification)**

Hierarchical to: No other components

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

FCS_COP.1.1/RSASign The TSF shall perform *signature generation and verification* in accordance with a specified cryptographic algorithm RSASSA-PKCS1-v1_5, RSASSA-PSS and cryptographic key sizes *2048 bit, 3072 bits and 4096 bits* that meet the following: *RFC3447 PKCS#1 v2.1 [27].*

FCS_COP.1/ECDSA Cryptographic operation (ECC signature generation/verification)

Hierarchical to: No other components
Dependencies: [FDT_ITC.1 Import of user data without security attributes,
or FDP_ITC.2 Import of user data with security attributes,
or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/ECDSA The TSF shall perform *signature generation and verification* in accordance with a specified cryptographic algorithm *ECDSA with curve TPM_ECC_NIST_P256 and TPM_ECC_NIST_P384* and cryptographic key sizes *256 bits and 384 bits* that meet the following: ISO/IEC 14888-3 [23] and FIPS PUB 186-4 [14].

FCS_COP.1/ECDEC Cryptographic operation (decryption)

Hierarchical to: No other components
Dependencies: [FDT_ITC.1 Import of user data without security attributes,
or FDP_ITC.2 Import of user data with security attributes,
or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/ECDEC The TSF shall perform *decryption of ECC key* in accordance with a specified cryptographic algorithm *ECDH with curve TPM_ECC_NIST_P256, TPM_ECC_NIST_P384 and none* and cryptographic key sizes *256 bits and 384 bits and none* that meet the following: *TPM library specification [6] and NIST Special Publication 800-56A [17] and ISO/IEC 15946-1 [24].*

FIA_UID.1: Timing of identification

Hierarchical to: No other components
Dependencies: No dependencies

FIA_UID.1.1 The TSF shall allow

(1) to execute indication _TPM_Hash_Start, _TPM_Hash_Data and _TPM_Hash_End,
(2) to execute commands that do not require authentication,
(3) to access objects where the entity owner has defined no authentication requirements (authValue, authPolicy),
(4) None

on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user, e.g. self-test.

FPT_TST.1 TSF testing

Hierarchical to:	No other components
Dependencies:	No dependencies

FPT_TST.1.1 The TSF shall run a suite of self-tests

(1) at the request of the authorized user "World"
 (a) the TPM2_SelfTest command and of selected algorithms using the TPM2_IncrementalSelfTest command,
(2) at the conditions
 (a) Initialization state after reset and before the reception of the first command,
 (b) prior to execution of a command using a not self-tested function,
(3) None

to demonstrate the correct operation of sensitive parts of the TSF.

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of *TSF data.*

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of the TSF.

FPT_FLS.1/FS Failure with preservation of secure state (fail state)

Hierarchical to:	No other components
Dependencies:	No dependencies

FPT_FLS.1.1/FS The TSF shall preserve a secure state by entering the Fail state when the following types of failures occur:

(1) If during TPM Restart or TPM Resume, the TPM fails to restore the state saved at the last Shutdown(STATE), the TPM shall enter Failure Mode and return TPM_RC_FAILURE.
(2) failure detected by TPM2_ContextLoad when the decrypted value of sequence is compared to the stored value created by TPM2_ContextSave(),
(3) failure detected by self-test according to FPT_TST.1.
(4) *None.*

FPT_PHP.3 Resistance to physical attack

Hierarchical to:	No other components

Dependencies: No dependenciess

FPT_PHP.3.1 The TSF shall resist physical manipulation and physical probing to the TSF by responding automatically such that the SFRs are always enforced.

FDP_ACF.1/States: Security attribute based access control (operational states)

Hierarchical to: No other components

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/States The TSF shall enforce the TPM State Control SFP to objects based on the following

Subjects as defined in Table 7*, [5]*:

(1) Platform firmware with the security attributes platformAuth, platformPolicy and physical presence if supported by the TOE,

(2) all other subjects; their security attributes are irrelevant for this SFP,

Objects as defined in Table 8 and Table 9*, [5]*:

(1) Shutdown BLOB with the security attribute validation status,

(2) Firmware update data with security attributes signature of the TPM manufacturer and digest,

(3) all other objects; their security attributes are irrelevant for this SFP.

FDP_ACF.1.2/States The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) The *Admin* is authorized to change the TPM state to FUM if the authenticity of the first digest or the signature could be successfully verified.

(2) While in FUM state the Platform firmware is authorized to import or activate firmware data only after successful verification of its integrity and authenticity (see FDP_UIT.1/States).

(3) The FUM state shall only be left when *code activation is completed.*

(4) In the Init state the subject "World" is authorised to execute the commands TPM2_Startup and the sequence _TPM_Hash_Start, _TPM_Hash_Data, and _TPM_Hash_End.

(5) In the Init state every subject is authorized to process the Resume operation on the Shutdown BLOB with state transition to Operational.

(6) In the Init state every subject is authorized to process the Restart operation on the Shutdown BLOB with state transition to Operational.

(7) In the Init state, if no Shutdown BLOB was generated or if the Shutdown BLOB is invalid (see attribute "Validation status") every subject is authorized to process the TPM2_Startup command. In the case of the parameter TPM_SU_CLEAR the TPM shall change the state to Operational and initialize its internal operational variables to default initialization values (Reset), otherwise the TPM shall return an error and stay in the same state.

(8) In the Operational state, nobody is authorized to execute the command TPM2_Startup. For all other subjects, objects and operations, the access control rules of the Access Control SFP shall apply (see FDP_ACF.1/AC).

(9) The Operational state shall change to Self-Test state if one of the commands TPM2_Selftest or TPM2_IncrementalSelfTest is executed or when a test of a dedicated functionality is required (see FPT_TST.1). In the Self-Test state, nobody is authorized to execute any other TPM command.

(10) The Self-Test state shall be left only after finishing the intended test of the dedicated functionality. In the case of a successful test result the state shall change to Operational, otherwise to Fail.

(11) In the Fail state, every subject is authorized to execute the commands TPM2_GetTestResult and TPM2_GetCapability.

(12) In the Fail state the subject World is authorized to send a _TPM_Init indication with state change to Init.

(13) Any subject is authorized to prepare the TPM for a power cycle using the TPM2_Shutdown command and to create a shutdown BLOB by TPM2_Shutdown(TPM_SU_STATE).

FDP_ACF.1.3/States The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

(1) the TPM authorises to enter FUM state if the firmware update data major version is equal to the major version of the loaded firmware

(2) the TPM authorises to enter FUM state if the firmware update data minor version is bigger than or equal to the minor version of the loaded firmware

(3) the TOE authorises to enter FUM state if the upgrade counter is strictly lower than the limit upgrade counter

FDP_ACF.1.4/States The TSF shall explicitly deny access of subjects to objects based on the following additional rules:

(1) Once the TPM receives a TPM2_SelfTest command and before completion of all tests, the TPM shall return TPM_RC_TESTING for any command that uses a command that requires a test.

FDP_UIT.1/States Data exchange integrity (operational states)

Hierarchical to:	No other components
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]

FDP_UIT.1.1/States The TSF shall enforce the TPM state control SFP to receive firmware update data in a manner protected from modification, deletion, insertion, replay errors.

FDP_UIT.1.2/States The TSF shall be able to determine on receipt of firmware update data, whether *modification, deletion, insertion, replay* has occurred.

FDP_ACF.1/AC Security attribute based access control (access control)

Hierarchical to: No other components

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1/AC The TSF shall enforce the Access Control SFP to objects based on the following

Subjects:

(1) Platform firmware with security attribute authorisation state gained by authentication with platformAuth, platformPolicy or physical presence if supported by the TOE,

(2) Platform owner with security attribute authorisation state gained by authentication with ownerAuth or ownerPolicy,

(3) Privacy administrator with security attribute authorisation state gained by authentication with endorsementAuth or endorsementPolicy,

(4) Lockout administrator with security attribute authorisation state,

(5) USER with authentication state gained with userAuth or authPolicy,

(6) DUP with authentication state gained with authPolicy,

(7) ADMIN with authentication state gained with userAuth or authPolicy,

(8) World with no security attributes,

Objects:

(1) User key with security attributes TPM_ALG_ID, TPMA_OBJECT,

(2) TPM objects,

(3) Clock with security attributes: resetCount, restartCount, safe-flag,

(4) Data with security attribute “externally provided”

FDP_ACF.1.2/AC The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

(1) The Platform firmware authorized with platformAuth, platformPolicy or with physical presence if supported by the TOE and the Platform owner are authorised to control the persistence of loadable objects in TPM memory (TPM2_EvictControl). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_EvictControl command.

(2) The Platform firmware platformAuth, platformPolicy or with physical presence if supported by the TOE and the Platform owner are authorised to advance the value and to adjust the rate of advance of the TPMs clock (TPM2_ClockSet, TPM2_ClockRateAdjust). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_ClockSet respective TPM2_ClockRateAdjust command.

(3) Any subject is authorised to get the current value of time, clock, resetCount, restartCount and safe (TPM2_ReadClock).

(4) A subject with the role USER endorsed by the Privacy administrator and the keyHandle identifier of a loaded key that can perform digital signatures is authorised to get the current value of time and clock (TPM2_GetTime).

(5) No subject is authorised to set the clock to a value less than the current value of clock using the TPM2_ClockSet command.

(6) No subject is authorised to set the clock to a value greater than its maximum value (0xFFFF000000000000) using the TPM2_ClockSet command.

(7) A subject with the role USER is authorised to generate digital signatures using the command TPM2_Sign for externally provided data (hash). The user authorisation shall be done based on the required authorisation of the key that will perform signing. The key attributes shall allow the signing operation for externally provided data.

(8) Any subject is authorised to verify digital signatures using the command TPM2_VerifySignature.

(9) Any subject is authorised to request data from the random number generator using the command TPM2_GetRandom.

(10) Any subject is authorised to add additional information to the state of the random number generator using the command TPM2_StirRandom.

(11) Any subject is authorised to perform RSA encryption using the command TPM2_RSA_Encrypt for externally provided data. The key attributes shall allow the encrypt operation for externally provided data.

(12) A subject with the role USER is authorised to perform RSA decryption using the command TPM2_RSA_Decrypt for externally provided data. The user authorisation shall be done based on the required authorisation of the key that will be used for decryption. The key attributes shall allow the decrypt operation for externally provided data.

(13) Any subject is authorised to generate ECC ephemeral key pairs using the command TPM2_ECDH_KeyGen.

(14) A subject with the role USER is authorised to recover a value that is used in ECC based key sharing protocols using the command TPM2_ECDH_ZGen. The user authorisation shall be done based on the required authorisation of the involved private key.

(15) Any subject is authorised to request the parameters of an identified ECC curve using the command TPM2_ECC_Parameters.

(16) The subject USER is authorised to start a HMAC sequence using the command TPM2_HMAC_Start.

(17) The subject World is authorised to start a hash or event sequence using the command TPM2_HashSequenceStart.

(18) The subject USER is authorised to add data to a hash, event or HMAC sequence using the command TPM2_SequenceUpdate.

(19) The subject USER is authorised to add the last part of data (if any) to a hash or HMAC sequence using the command TPM2_ SequenceComplete.

(20) The subject USER is authorised to add the last part of data (if any) to an event sequence using the command TPM2_EventSequenceComplete.

(21) Any subject is authorised to perform hash operations on a data buffer using the command TPM2_Hash.

(22) A subject with the role USER is authorised to perform HMAC operations on a data buffer. The user authorisation shall be done based on the required authorisation of the involved symmetric key.

(23) A subject with the role USER is authorised to generate HMACs using the command TPM2_HMAC for externally provided data (hash). The user authorisation shall be done based on the required authorisation of the key that will perform the HMAC. The key attributes shall allow the signing operation for externally provided data.

FDP_ACF.1.3/AC The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *none.*

FDP_ACF.1.4/AC The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*

7.2.1 Extended Component FCS RNG.1

The PP [5] defines the extended family Random Number Generation (FCS_RNG) of the class FCS (cryptographic support). This family describes the functional requirements for random number generation used for cryptographic purposes.

FCS_RNG.1/DRBG Deterministic random number generation according to NIST SP800-90A [19]

Hierarchical to: No other components

Dependencies: No dependencies

FCS_RNG.1.1/DRBG: The TSF shall provide a *deterministic* Random Number Generator that implements: *NIST SP800-90A CTR_DRBG with AES-256 and* derivation function using a block cipher algorithm.

FCS_RNG.1.2/DRBG: The TSF shall provide random numbers that meet: *Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG.*

FCS_RNG.1/NRBG: *Non-deterministic* random number generation according to NIST SP800-90B [20]

Hierarchical to: No other components

Dependencies: No dependencies

FCS_RNG.1.1/NRBG: The TSF shall provide a *Non-deterministic* Random Bits Generator that implements: *an entropy source based on a hardware physical source, health test and conditioning* *component conforms to NIST SP800-90B. The output of entropy source is used as the input of the deterministic* random number generator *with NIST SP800-90A.*

FCS_RNG.1.2/NRBG: The TSF shall provide random numbers that meet: *Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG.*

7.3 TOE Security Assurance Requirement

The Security Assurance Requirements (SAR) for the TOE are the assurance components of Evaluation Assurance Level 4 (EAL4) as defined in CC part 3 and augmented with ALC_FLR.1 and AVA_VAN.4.

The security assurance requirements defined in Table 3 are defined in section 8.2 of the PP [5].

Table 5: Security Assurance Requirements for the TOE

Assurance Class	Assurance Family
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance Documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-Cycle Support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_FLR.1 Basic flow remediation - **augmented**
	ALC_TAT.1 Well-defined development tools
ASE: Security Target Evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: security enforcing modules
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.4 Methodical vulnerability analysis - **augmented**

7.4 Security Requirement Rationale

The security Requirement Rationale of the TOE is defined and described in the PP [5], section 8.3. Security Requirement rationale.

7.4.1 Sufficiency of SFR

The sufficiency of SFR in this ST is consistent with PP chapter 8.3.1. The mapping of added iterations and security objectives is as follows:

The SFRs FCS_CKM.1/PKRSA, FCS_CKM.1/PKECC and FCS_CKM.1/PKSYM fulfil the same objectives as the SFR FCS_CKM.1/PK defined in the PP [5] Table 11.

The SFRs FCS_RNG.1/DRBG and FCS_RNG.1/NRBG fulfil the same objectives as the SFR FCS_RNG.1 defined in the PP [5] Table 11.

7.4.2 Dependency rationale

The majority of the dependency rationale is detailed in Chapter 8.3.2 of the PP [5]. This chapter serves as a supplement, introducing iterations of existing SFRs.

Table 7: SFR Dependency Rationale of iterations added in ST

SFR	Dependency	Rationale / fulfilled by
FCS_CKM.1/PKRSA	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/RSAED, FCS_COP.1/RSASign, FCS_CKM.4
FCS_CKM.1/PKECC	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/ECDEC, FCS_COP.1/ECDSA, FCS_CKM.4
FCS_CKM.1/PKSYM	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	Fulfilled by FCS_COP.1/AES, FCS_CKM.4
FCS_RNG.1/DRBG	No dependencies	n. a.
FCS_RNG.1/NRBG	No dependencies	n. a.

For existing SFRs:
The dependency of FCS_CKM.4 is also fulfilled by FCS_CKM.1/PKRSA, FCS_CKM.1/PKECC and FCS_CKM.1/PKSYM.
The dependency of FCS_COP.1/RSAED is also fulfilled by FCS_CKM.1/PKRSA.
The dependency of FCS_COP.1/RSASign is also fulfilled by FCS_CKM.1/PKRSA.
The dependency of FCS_COP.1/AES is also fulfilled by FCS_CKM.1/PKSYM.

The dependency of FCS_COP.1/ECDEC is also fulfilled by FCS_CKM.1/PKECC.

7.5 Security Assurance Rationale

The security Requirement Rationale of the TOE is defined and described in the PP [5], section 8.3.3 Assurance rationale.

8 TOE summary specification (ASE_TSS)

The product overview is given in section 2.2. In the following the security functionality and the assurance measures of the TOE are described.

8.1 TOE security features

This section contains the definition and description of the security features (SF) of the TOE. The TOE provides five security features to meet the security functional requirements. The security figures are:

SF_CRY: Cryptographic Support

SF_I&A: Identification and Authentication

SF_G&T: General and Test

SF_OBH: Object Hierarchy

SF_TOP: TOE Operation

8.1.1 SF_CRY: Cryptographic Support

There are several functions within the TOE related to cryptographic support: generation of random numbers, generation of asymmetric key pairs, RSA and ECC digital signature (generation and verification), RSA and AES data encryption and decryption, *decryption of ECC key* in accordance with a specified cryptographic algorithm *ECDH,* key destruction, the generation of hash values and the generation and verification of MAC values.

The TOE supports the generation of cryptographic keys in accordance with the specified cryptographic key generation algorithm RSA key generator and ECC key generator and specified cryptographic key sizes RSA 2048 bits, 3072 and 4096 bits that meet the following: RFC3447 PKCS #1 v2.1 [27], FIPS PUB 186-4 [14] and ECDSA with key sizes of 256 bits and 384 bits that meet ISO/IEC 15946-1 [24]. The source of randomness is the internal random generator.

The covered security functional requirements are FCS_CKM.1/PKRSA, FCS_CKM.1/PKECC, FCS_CKM.1/RSA and FCS_CKM.1/ECC.

The TOE supports the generation of symmetric cryptographic keys in accordance with the specified cryptographic key generation algorithm AES key generator and specified cryptographic key sizes 128 bits and 256 bits that meet NIST Special Publication 800-133 [18].

The covered security functional requirements are FCS_CKM.1/PKSYM and FCS_CKM.1/SYMM.

The TOE supports the destruction of cryptographic keys by erasure of volatile memory areas containing cryptographic keys in accordance with ISO/IEC 19790:2012 [26].

The covered security functional requirement is FCS_CKM.4.

The TOE performs the encryption and decryption in accordance with the specified cryptographic algorithm AES in the CFB mode and cryptographic key size of 128 bits and 256 bits that meet [SP800-38A][16] or [ISO10116:2006][22] or [ISO 18033-3][25].

The covered security functional requirement is FCS_COP.1/AES.

The TOE performs the hash value calculation in accordance with the specified cryptographic algorithm SHA-1, SHA-256 and SHA-384 (cryptographic key sizes not available) that meets FIPS 180-4 [13].

The covered security functional requirement is FCS_COP.1/SHA.

The TOE performs HMAC value calculation and verification in accordance with the specified cryptographic algorithm HMAC with SHA-1, SHA-256 and SHA-384 and cryptographic key sizes160 bits, 256 bits, 384bits and 512 bits that meets FIPS 198-1[15] or ISO/IEC 9797-2 [21].

The covered security functional requirement is FCS_COP.1/HMAC.

The TOE performs asymmetric encryption and decryption in accordance with the specified cryptographic algorithm RSA without padding, RSAES-PKCS1-v1_5, RSAES-OAEP and cryptographic key sizes 2048 bits, 3072 and 4096 bits that meet RFC3447 PKCS#1 v2.1 [27].

The covered security functional requirements are FCS_COP.1/RSAED.

The TOE performs signature generation and signature verification in accordance with the specified cryptographic algorithm RSASSA-PKCS1v1_5, RSASSA_PSS and cryptographic key sizes 2048 bits, 3072 and 4096 bits that meet RFC3447 PKCS#1v2.1 [27].

The covered security functional requirement is FCS_COP.1/RSASign.

The TOE performs signature generation and signature verification in accordance with the specified cryptographic algorithm ECDSA with curve TPM_ECC_NIST_P256 and TPM_ECC_NIST_P384 and cryptographic key sizes 256 bits and 384 bits that meet TPM library specification [6] section C.4.

The covered security functional requirement is FCS_COP.1/ECDSA.

The TOE performs decryption of ECC key in accordance with the specified cryptographic algorithm ECDH with curve TPM_ECC_NIST_P256, TPM_ECC_NIST_P384 and none and cryptographic key sizes 256 bits and 384 bits and none that meet the following: TPM library specification [6], [7], [8], and NIST Special Publication 800-56A [17].

The covered security functional requirement is FCS_COP.1/ECDEC.

The TOE provides a deterministic random number generator (DRBG) including a true random generator, which is used for the seeding of the DRBG, to provide the random numbers. The TOE provides deterministic random numbers that fulfils the requirements of NIST Special Publication 800-90A [19].

The TOE provides an entropy source based on a hardware physical source, health test and conditioning component compliance with NIST Special Publication 800-90B [20] , and the output of hardware entropy source is used as the input of the deterministic random number generator with NIST Special Publication 800-90A. The TOE uses the internal true random generator as the source for any randomness that the processes defined in SF_CRY may require.

The covered security functional requirement is FCS_RNG.1.

The SF_CRY "Cryptographic Support" covers the following security functional requirements: FCS_CKM.1/PKRSA, FCS_CKM.1/PKECC, FCS_CKM.1/PKSYM, FCS_CKM.1/RSA, FCS_CKM.1/ECC, FCS_CKM.1/SYMM, FCS_CKM.4, FCS_COP.1/AES, FCS_COP.1/SHA, FCS_COP.1/HMAC, FCS_COP.1/RSAED, FCS_COP.1/RSASign, FCS_COP.1/ECDSA, and FCS_RNG.1.

8.1.2 SF_I&A: Identification and Authentication

The TPM provides two mechanisms for the identification and authentication capability to authorize the use of the Protected Object and Protected Capability. Note that the TCG TPM Library specification refers to the identification and authentication process and access control as authorization. The first authentication mechanisms is the prove of knowledge of a shared secret (password or secret for HMAC) assigned to the entity as authValue. The second mechanism is the authentication of the user and verification of an intended state of the TPM and its environment encoded in authPolicy and assigned to the entity.

The TOE provides a mechanism to generate secrets that meet uniform distribution of random variable generating the value, and is able to enforce the use of TSF generated secrets for nonce values for authorization sessions unknown authValues.

The covered security functional requirement is FIA_SOS.2.

The TOE use different rules to set value of security attributes.

The covered security functional requirement is FMT_MSA.4/AUTH.

The TOE provides the management functionality of the TSF data by user authorization.

The covered security functional requirement is FMT_MTD.1/AUTH.

The TOE detects when the maximal tries of unsuccessful authentication attempts occur for objects and NV Index where DA is active and blocks the authorizations for a defined time.

The covered security functional requirement is FIA_AFL.1/Recover.

The TOE detects when an unsuccessful authentication attempts occur using lockoutAuth in the command TPM2_DictionaryAttackLockReset and blocks the TPM2_DictionaryAttackLockReset command for a defined time.

The covered security functional requirement is FIA_AFL.1/Lockout.

The TOE detects when a defined number of successful authentication events exceeds pinLimit for an NV index with the attribute TPM_NT_PIN_PASS and blocks further authorization events.

The covered security functional requirement is FIA_AFL.1/PINPASS.

The TOE detects when a defined number of unsuccessful authentication events exceeds pinLimit for an NV index with the attribute TPM_NT_PIN_FAIL and blocks further authorization events.

The covered security functional requirement is FIA_AFL.1/PINFAIL.

The TOE allows access to a defined number of commands and objects for the user to be performed before the user is authenticated/identified.

The covered security functional requirements are FIA_UID.1 and FIA_UAU.1.

The TOE provides different authentication mechanisms to support user authentication and authenticate any users' claimed identity according to the different rules. The TOE provides re-authentication of the user for multiple command processing.

The covered security functional requirements are FIA_UAU.5 and FIA_UAU.6.

The TOE associate security attributes with subject acting on the behalf of that user. The TOE enforces different rules on the initial association of user security attributes with subjects acting on the behalf of users and enforces different rules governing changes to the user security attributes associated with subjects acting on the behalf of users.

The covered security functional requirement is FIA_USB.1.

The SF_I&A “Identification and Authentication” covers the following security functional requirements, FIA_SOS.2, FIA_MSA.4/AUTH, FMT_MTD.1/AUTH, FIA_AFL.1/Recover, FIA_AFL.1/Lockout, FIA_AFL.1/PINPASS, FIA_AFL.1/PINFAIL, FIA_UID.1, FIA_UAU.1, FIA_UAU.5, FIA_UAU.6 and FIA_USB.1.

8.1.3 SF_G&T: General and Test

The TOE provides the roles: Platform firmware, Platform owner, Privacy Administrator, Lockout Administrator, User, Admin, DUP and World and associates users with roles. The roles are enforced within the TOE because there are specific commands and specific keys bond to different token.

The covered security functional requirement is FMT_SMR.1.

The TOE performs different management functions.

The covered security functional requirement is FMT_SMF.1.

The TOE ensures that only secure values are accepted for security attributes.

The covered security functional requirement is FMT_MSA.2.

The TOE provides reliable time stamps as number of milliseconds the TOE has been powered since initialization of the Clock value.

The covered security functional requirement is FPT_STM.1.

The TOE ensures that any previous information content of a resource is made unavailable upon the deal location of the resource from defined objects.

The covered security functional requirement is FDP_RIP.1.

The TOE supports a suite of self tests during startup and at the request of an authorized user world to demonstrate the correct operation of sensitive parts of the TSF and to verify the integrity of stored TSF executable code and parts of TSF data.

The covered security functional requirement is FPT_TST.1.

The TOE preserves a secure state by entering the Fail state when a failure during TPM Restart or Resume occurs, a failure is detected by TPM2_ContecxtLoad or the self test, of any crypto operations including RSA encryption, RSA decryption, AES encryption, AES decryption, SHA-1, RNG, RSA signature generation, HMAC generation or failure of any commands or internal operations and authorization occurs.

The covered security functional requirement is FPT_FLS.1/FS.

The TOE preserves a secure state by shutdown, when detecting a physical attack or an environmental condition which is out of spec value.

The covered security functional requirement is FPT_FLS.1/SD.

The TOE resists physical manipulation and physical probing to the TSF by responding automatically such that the SFRs are always enforced.

The TOE supports the following functions for protection against and detection of physical manipulation and probing:

- The correct function of the TOE is only given in the specific range of the environmental operating parameters. To prevent an attack exploiting those circumstances the external voltage conditions, the temperature and electro-magnetic radiation (e.g. light) are observed to detect if the specified range is left. 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 control signals of flash are automatically monitored by the glue logic, and the data in the flash are checked by the ECC. Once the control signals of flash or the data written to the flash are tampered, an automatic reset of the TOE will be generated.

- Several mechanisms protect the TOE against snooping the design or the user data during operation and even if it is out of operation (power down). There are topological design measures for disguise, such as the protection of security critical lines by specific intelligent and intrinsic shielding including secure wiring of security critical signals. The entire design is kept in a non standard way to prevent attacks using standard analysis methods.

- The readout of data can be controlled with the use of encryption. An attacker can not use the data obtained by espionage due to their encryption. The memory contents of the TOE are encrypted on chip to protect against data analysis on stored data as well as on internally transmitted data.

- The virtual physical address mapping together with the memory management unit (MMU) gives the operating system the possibility to define different access rights for memory areas. In case of an access violation the MMU will generate a non maskable interrupt (NMI) and an interrupt service routine react on the access violation.

The covered security functional requirement is FPT_PHP.3, FDP_ITT.1 and FPT_ITT.1.

The SF_G&T “General and Test” covers the following security functional requirements, FMT_SMR.1, FMT_SMR.1, FMT_SMF.1, FMT_MSA.2, FPT_STM.1, FDP_RIP.1, FPT_TST.1, FPT_FLS.1/FS, FPT_FLS.1/SD, FPT_PHP.3, FDP_ITT.1 and FPT_ITT.1.

8.1.4 SF_OBH: Object Hierarchy

The TOE supports different states during his life-cycle as described in [5] section 8.1.4.1 “TPM Operational States” in detail

The TOE enforces the TPM State Control SFP on all subjects and objects and all operations among subjects and objects covered by the SFP. The TOE 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 and enforces different access control rules on controlled subjects and objects.

The covered security functional requirements are FDP_ACC.2/States and FDP_ACF.1/States.

The TOE enforce the TPM state control SFP to restrict the ability to modify the security attributes TPM state and to provide restrictive default values for security attributes that are used to enforce the SFP. The TOE enforce the TPM state control SFP to receive firmware update data in a manner protected from errors and determines on receipt of firmware update data, whether error has occurred.

The covered security functional requirements are FMT_MSA.1/States, FMT_MSA.3/States and FDP_UIT.1/States.

The TOE supports three different hierarchies, the platform hierarchy, the storage hierarchy and the endorsement hierarchy. The root of each TPM hierarchy is defined by a primary seed which is a random value persistently stored in the TOE. A hierarchy may be disabled.

The TOE monitors user data stored in containers controlled by the TSF for data modifications and modification of hierarchy on all objects, based on the different attributes.

The covered security functional requirement is FDP_SDI.1.

The TOE enforces the TPM Object Hierarchy SFP on defined subjects, objects and operations and enforces different rules to determine if an operation among controlled subjects and controlled objects is allowed and deny access of subjects to objects based on different rules.

The covered security functional requirements are FDP_ACC.1/Hier and FDP_ACF.1/Hier.

The TOE enforces the TPM Object Hierarchy SFP to not allow the modification of the security attributes fixedTPM and fixedParent.

The covered security functional requirement is FMT_MSA.1/Hier.

The TOE enforces the TPM Object Hierarchy SFP to provide restrictive default values for security attributes that are used to enforce the SFP and allows the creator of an object in a TPM hierarchy to specify alternative initial values to override the default values when an object or information is created.

The covered security functional requirement is FMT_MSA.3/Hier.

The TOE enforces different rules to set the value of security attributes.

The covered security functional requirement is FMT_MSA.4/Hier.

The TOE allows the import and export of data as an object of a hierarchy.

The TOE enforces the Data Export and Import SFP on subjects, objects and operations. The Data Export and Import SFP enforce different rules to determine if an operation between a controlled subject and controlled object is allowed.

The covered security functional requirements are FDP_ACC.1/ExIm and FDP_ACF.1/ExIm.

The TOE enforce the Data Export and Import SFP to restrict the ability to use the security attribute authorization data to every subject, to provide restrictive default values for security attributes that are used to enforce the SFP and to prevent to override the default values when an object or information is created.

The covered security functional requirements are FMT_MSA.1/ExIm and FMT_MSA.3/ExIm.

The TOE enforces the Data Export and Import SFP when exporting user data, controlled under the SFP(s), outside of the TOE and to export the user data with the user data's associated security attributes. The TOE ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data and different rules are enforced when user data is exported from the TOE.

The covered security functional requirement is FDP_ETC.2/ExIm.

The TOE enforces the Data Export and Import SFP when importing user data, controlled under the SFP(s), outside of the TOE. The correct interpretation, association and use of the security attributes associated with the imported user data are ensured and different rules are enforced when user data is imported from outside the TOE.

The covered security functional requirement is FDP_ITC.2/ExIm.

The TOE enforces the Data Export and Import SFP to transmit user data in a manner protected from unauthorised disclosure and to transmit and receive user data in a manner protected from modification errors. The TOE is able to determine on receipt of user data, whether modification has occurred.

The covered security functional requirements are FDP_UCT.1/ExIm and FDP_UIT.1/ExIm.

The TOE enforces the Measurement and Reporting SFP on subjects, objects and operations. The Measurement and Reporting SFP enforce different rules to determine if an operation among controlled subjects and controlled objects is allowed.

The covered security functional requirements are FDP_ACC.1/M&R and FDP_ACF.1/M&R.

The TOE enforces the Measurement and Reporting SFP to restrict the ability to modify the security attributes PCR attributes, PCR extension algorithm and used hash algorithm to the subject Platform firmware, to provide restrictive default values for security attributes that are used to enforce the SFP, and to prevent to override the default values when an object or information is created.

The covered security functional requirements are FMT_MSA.1/M&R and FMT_MSA.3/M&R.

The TOE is able to generate evidence of origin for transmitted attestation structure and object creation tickets at the request of the originator and provide a capability to verify the evidence of origin of information to recipient given as soon as the recipient can verify the signature and has confidence to the key that is used to sign.

The covered security functional requirement is FCO_NRO.1/M&R.

The SF_OBH "Object Hierarchy" covers the following security functional requirements: FDP_ACC.2/States, FDP_ACF.1/States, FMT_MSA.1/States, FMT_MSA.3/States, FDP_UIT.1/States, FDP_SDI.1, FDP_ACC.1/Hier, FDP_ACF.1/Hier, FMT_MSA.1/Hier, FMT_MSA.3/Hier, FMT_MSA.4/Hier, FDP_ACC.1/ExIm, FDP_ACF.1/ExIm, FMT_MSA.1/ExIm, FMT_MSA.3/ExIm, FDP_ETC.2/ExIm, FDP_ITC.2/ExIm, FDP_UCT.1/ExIm, FDP_UIT.1/ExIm, FDP_ACC.1/M&R, FDP_ACF.1/M&R, FMT_MSA.1/M&R, FMT_MSA.3/M&R and FCO_NRO.1/M&R.

8.1.5 SF_TOP: TOE operation

The TOE enforces the Access Control SFP on different subjects, objects and operations and enforces different rules to determine if an operation among controlled subjects and controlled objects is allowed. The TOE explicitly authorize access of subjects to objects based on different additional rules and explicitly deny access of subjects to objects based on the different additional rules.

The covered security functional requirements are FDP_ACC.1/AC and FDP_ACF.1/AC.

The TOE enforces the Access Control SFP to restrict the ability to query and modify different security attributes to specific subjects, to provide restrictive default values for security attributes that are used to enforce the SFP and to specify alternative initial values to override the default values when an object or information is created.

The covered security functional requirements are FMT_MSA.1/AC and FMT_MSA.3/AC.

The TOE enforces the Access Control SFP to transmit user data in a manner protected from unauthorised disclosure. The TOE provides a communication channel between itself and another trusted IT product 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. The TOE initiates communication via the trusted channel and permits another trusted IT product to initiate communication via the trusted channel.

The covered security functional requirements are FDP_UCT.1/AC and FTP_ITC.1/AC.

The TSF shall restrict the ability to disable and enable the functions TPM2_Clear to the subjects Platform firmware and Lockout administrator.

The covered security functional requirement is FMT_MOF.1/AC.

The TSF shall enforce the NVM SFP on different subjects, objects and operations and enforces different rules to determine if an operation among controlled subjects and controlled objects is allowed.

The covered security functional requirements are FDP_ACC.1/NVM and FDP_ACF.1/NVM.

The TOE enforces the NVM SFP to restrict the ability to query and modify the security attribute NV index attributes to the authorized role of the subject that executes the NVM related command and to provide restrictive default values when an object or information is created. The TOE prohibits to override the default values with alternative initial values when an object or information is created. The TOE enforces different rules to set the value of security attributes and restrict the ability to modify the authorization secret (authValue) for a NV index to the subject ADMIN.

The covered security functional requirements are FMT_MSA.1/NVM, FMT_MSA.3/NVM, FMT_MSA.4/NVM and FMT_MTD.1/NVM.

The TOE enforces the NVM SFP when importing user data, controlled under the SFP, and ignores any security attributes associated with the user data when imported from outside the TOE. Additionally the TOE enforces different rules when importing user data controlled under the SFP from outside the TOE. The TOE enforces the NVM SFP when exporting user data, controlled under the SFP(s), outside of the TOE.

The covered security functional requirements are FDP_ITC.1/NVM and FDP_ETC.1/NVM.

The TOE enforces the Credential SFP on different subjects, objects and operations and enforces different rules to determine if an operation among controlled subjects and controlled objects is allowed.

The covered security functional requirements are FDP_ACC.1/Cre and FDP_ACF.1/Cre.

The TOE enforces the Credential SFP to provide restrictive default values for security attributes that are used to enforce the SFP and prevents to override the default values when an object or information is created. The TOE enforces the Credential SFP to restrict the ability to use the security attributes HMAC in the credential BLOB to the subject USER.

The covered security functional requirements are FMT_MSA.1/Cre and FMT_MSA.3/Cre.

The TOE generates evidence of origin for transmitted TPM objects at the request of the originator and relates the information whether the object is resident in an authentic TPM of the originator of the information, and the name and the public area of the TPM object of the information to which the evidence applies. The TOE provides a capability to verify the evidence of origin of information to the initiator given based on a credential BLOB that was generated by the credential provider.

The covered security functional requirement is FCO_NRO.1/Cre.

The SF_TOE "TOE Operation" covers the following security functional requirements: FDP_ACC.1/AC, FDP_ACF.1/AC, FMT_MSA.1/AC, FMT_MSA.3/AC, FDP_UCT.1/AC, FTP_ITC.1/AC, FMT_MOF.1/AC, FDP_ACC.1/NVM, FDP_ACF.1/NVM, FMT_MSA.1/NVM, FMT_MSA.3/NVM, FMT_MSA.4/NVM, FMT_MTD.1/NVM, FDP_ITC.1/NVM, FDP_ETC.1/NVM, FDP_ACC.1/Cre, FDP_ACF.1/Cre, FMT_MSA.1/Cre, FMT_MSA.3/Cre and FCO_NRO.1/Cre.

8.1.6 Assignment of Security Functional Requirement

The justification of the mapping between security functional requirements and the security features is given in sections 8.1.1 – 8.1.5. The results are shown at following table.

Table 8: Assignment security functional requirement to security features

Security Functional Requirement	SF_CRY	SF_I&A	SF_G&T	SF_OBH	SF_TOP
FMT_SMR.1			X		
FMT_SMF.1			X		
FMT_MSA.2			X		
FPT_STM.1			X		
FDP_RIP.1			X		
FCS_RNG.1/DRBG	X				
FCS_RNG.1/NRBG	X				
FCS_CKM.1/PKRSA	X				
FCS_CKM.1/PKECC	X				
FCS_CKM.1/PKSYM	X				
FCS_CKM.1/RSA	X				
FCS_CKM.1/ECC	X				
FCS_CKM.1/SYMM	X				
FCS_CKM.4	X				
FCS_COP.1/AES	X				
FCS_COP.1/SHA	X				
FCS_COP.1/HMAC	X				
FCS_COP.1/RSAED	X				
FCS_COP.1/RSASign	X				
FCS_COP.1/ECDSA	X				
FCS_COP.1/ECDEC	X				

FIA_SOS.2		X			
FMT_MSA.4/AUTH		X			
FMT_MTD.1/AUTH		X			
FIA_AFL.1/Recover		X			
FIA_AFL.1/Lockout		X			
FIA_UID.1		X			
FIA_UAU.1		X			
FIA_UAU.5		X			
FIA_UAU.6		X			
FIA_AFL.1/PINPASS		X			
FIA_AFL.1/PINFAIL		X			
FIA_USB.1		X			
FPT_TST.1			X		
FPT_FLS.1/FS			X		
FPT_FLS.1/SD			X		
FPT_PHP.3			X		
FDP_ITT.1			X		
FPT_ITT.1			X		
FDP_ACC.2/States				X	
FDP_ACF.1/States				X	
FMT_MSA.1/States				X	
FMT_MSA.3/States				X	
FDP_UIT.1/States				X	
FDP_SDI.1				X	
FDP_ACC.1/Hier				X	
FDP_ACF.1/Hier				X	
FMT_MSA.1/Hier				X	
FMT_MSA.3/Hier				X	
FMT_MSA.4/Hier				X	
FDP_ACC.1/ExIm				X	
FDP_ACF.1/ExIm				X	
FMT_MSA.1/ExIm				X	
FMT_MSA.3/ExIm				X	
FDP_ETC.2/ExIm				X	
FDP_ITC.2/ExIm				X	
FDP_UCT.1/ExIm				X	
FDP_UIT.1/ExIm				X	
FDP_ACC.1/M&R				X	
FDP_ACF.1/M&R				X	
FMT_MSA.1/M&R				X	
FMT_MSA.3/M&R				X	
FCO_NRO.1/M&R				X	
FDP_ACC.1/AC					X

FDP_ACF.1/AC					X
FMT_MSA.1/AC					X
FMT_MSA.3/AC					X
FDP_UCT.1/AC					X
FDP_ITC.1/AC					X
FMT_MOF.1/AC					X
FDP_ACC.1/NVM					X
FDP_ACF.1/NVM					X
FMT_MSA.1/NVM					X
FMT_MSA.3/NVM					X
FMT_MSA.4/NVM					X
FMT_MTD.1/NVM					X
FDP_ITC.1/NVM					X
FDP_ETC.1/NVM					X
FDP_ACC.1/Cre					X
FDP_ACF.1/Cre					X
FMT_MSA.1/Cre					X
FMT_MSA.3/Cre					X
FCO_NRO.1/Cre					X

9 Reference

9.1 Acronym

Acronym	Description
AMBA	Advanced Microcontroller Bus Architecture
BLOB	Binary Large Object
CC	Common Criteria
CP	Chip Probing
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRTM	Core Root of Trust for Measurement
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
EFC	Embedded Flash Controller
EK	Endorsement Key
EPS	Endorsement Primary Seed
FIPS	Federal Information Processing Standard
FT	Final Test
FU	Field Upgrade
FUM	Field Upgrade mode
HAL	Hardware Abstraction Layer
HMAC	Hash Message Authentication Code
NV	Non-Volatile
NVM	Non-Volatile Memory
PCR	Platform Configuration Register
PK	Primary Key
PP	Protection Profile
PPS	Platform Primary Seed
PTP	Platform TPM Profile
RAMC	Random Access Memory Controller
SDMA	Secure Direct Memory Access
SRAM	Static Random Access Memory
RNG	Random Number Generator
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SAC	Security Algorithm Co-processor
SFR	Security Functional Requirement
SPI	Serial Peripheral Interface
SPS	Storage Primary Seed
SPK	Storage Root Keys
SHA	Secure Hash Algorithm

ST	Security Target
TCB	Trusted Computing Base
TCG	Trusted Computing Group
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSF	TOE Security Functionality

9.2 Glossary

Term	Definition
AES	Advanced encryption standard and cryptographic primitives with encryption schemes
RSA	Rivest, Shamir and Adleman and Cryptographic primitives with encryption and signature generation schemes
ECC	Elliptic curve cryptography and cryptographic primitives with encryption and signature generation schemes
HAMC	Hash message authentication code and cryptographic primitives with message authentication using cryptographic hash functions
SHA	Secure hash standard and cryptographic primitives with SHA1,SHA256 and SHA384
HASH	Hash Crypto Engine
ASYM	Asymmetric Crypto Engine
SYM	Symmetric Crypto Engine
KDF	Key derivation function and cryptographic primitives
KDFa	The counter mode KDF, from SP800-108, uses HMAC as the pseudo-random function
KDFe	Producing a symmetric encryption key for an ECC-protected object uses “One-Pass Diffie-Hellman, C(1, 1, ECC CDH)” from SP800-56A, 6.2.2.2
DRBG	Deterministic random bit generators for the generation of random bits using deterministic methods
NRBG	Non-Deterministic random bit generators for the generation of random bits using physical entropy source methods

9.3 Literature

[1] [CCMB-2017-04-001]

Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 5, April 2017

[2] [CCMB-2017-04-002]

Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 3.1, Revision 5, April 2017

[3] [CCMB-2017-04-003]

Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 5, April 2017

[4] [CCMB-2017-04-004]

Common Methodology for Information Technology Security Evaluation (CEM), Part 2: Evaluation Methodology; Version 3.1, Revision 5, April 2017

[5] [TPM2.0 PP]

Protection Profile PC client specific TPM, TPM library specification family 2.0 Level 0 revision 1.59, Version 1.3, 29 September 2021

[6] [TPM2.0 Part1 r1.59]

TPM Library Part 1: Architecture, Specification Version 2.0, Revision 1.59, November. 2019, Trusted Computing Group Incorporated

[7] [TPM2.0 Part2 r1.59]

TPM Library Part 2: TPM Structures, Specification Version 2.0, Revision 1.59, November 2019, Trusted Computing Group Incorporated

[8] [TPM2.0 Part3 r1.59]

TPM Library Part 3: Commands, Specification Version 2.0, Revision 1. 59, November 2019, Trusted Computing Group Incorporated

[9] [TPM2.0 Part4 r1.59]

TPM Library Part 4: Supporting Routines, Specification Version 2.0, Revision 1. 59, November 2019, Trusted Computing Group Incorporated

[10] [TPM2.0 rev1.59 Err1.4]

Errata version 1.4 January 9, 2023 for TCG Trusted Platform Module Library, Family “2.0” Level 00 Revision 01.59, November 2019,, Trusted Computing Group Incorporated

[11] [PTP 1.05]

TCG PC Client Platform TPM Profile (PTP) Specification, Family “2.0”, Level 00

Version 1.05 Revision 14, September 4, 2020, Trusted Computing Group Incorporated

[12] [PTP 1.05 Err1.1]

Errata version 1.1 November 2019 for PC Client Platform TPM Profile (PTP) for TPM 2.0 Version 1.05 Revision 14, September 4, 2020, Trusted Computing Group Incorporated

[13] [FIPS 180-4]

FIPS180-4, Federal Information Processing Standard 180-4 Secure Hash Standard (SHS)

[14] [FIPS 186-4]

FIPS PUB 186-4, Federal Information Processing Standards Publication Digital Signature Standard (DSS), National Institute of Standards and Technology. July, 2013

[15] [FIPS 198-1]

FIPS 198-1 Federal Information Processing Standards Publication, The keyed-Hash Message Authentication Code (HMAC), July 2008.

[16] [SP800-38A]

NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation Methods and Techniques, December 2001.

[17] [SP800-56A]

NIST Special Publication 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptology. March 2007.

[18] [SP800-133]

NIST Special Publication 800-133, revision 2: Recommendation for Cryptographic Key Generation. July 2020.

[19] [SP800-90A]

NIST Special Publication 800-90A, revision 1: Recommendation for Random Number Generation Using Deterministic Random Bit Generators; June 2015

[20] [SP800-90B]

NIST Special Publication 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018

[21] [ISO/IEC 9797-2]

ISO/IEC 9797-2, Information technology -- Security techniques -- Message

Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash function

[22] [ISO/IEC 10116]

ISO/IEC 10116:2006, Information technology — Security techniques — Modes of operation for an n-bit block cipher

[23] [ISO/IEC 14888-3]

ISO/IEC 14888-3, Information technology -- Security techniques -- Digital signature with appendix -- Part 3: Discrete logarithm based mechanisms

[24] [ISO/IEC 15946-1]

ISO/IEC 15946-1, Information technology – Security techniques – Cryptographic techniques based on elliptic curves – Part 1: General. July 2016.

[25] [ISO/IEC 18033-3]

ISO/IEC 18033-3, Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers

[26] [ISO/IEC 19790]

ISO/IEC 19790:2012(E), Information technology — Security techniques — Security requirements for cryptographic modules

[27] [RFC 3447]

IETF RFC 3447, Public-Key Cryptography Standards PKCS#1:
RSA Cryptography Specifications Version v2.1; June 14, 2002
RSA Cryptography Specifications Version v2.0; October 1, 1998

[28] [ANSSI N6]

Application Note – Security requirements for post-delivery code loading, Version 2.0, January 23, 2015, ANSSI.

IMPORTANT NOTICE – PLEASE READ CAREFULLY

NSING Technologies Pte. Ltd. (“NSING”) can change, modify, enhance and improve its products and/or this document at any time without notice. It is advisable for purchasers to ensure they have the latest information about NSING's products before placing orders. When purchasing NSING's products, the responsibility solely lies on the purchaser to choose, select, and use the products, and NSING assumes no liability for any such responsibilities. NSING does not grant any license, whether express or implied, to any intellect property rights. If any purchaser resells NSING's products with provisions that differ from the information stated in this document, such a resale shall void any warranty granted by NSING for the product. NSING and the NSING logo are their trademarks, and for more information on NSING's trademarks, please see www.nsing.com.sg. All other product or service names belong to their respective owners. The information contained in this document supersedes and replaces the information supplied in any previous versions of the document.

NSING's Products are intended solely for use in general-purpose electronic equipment and are not recommended, authorized, or warranted for use in military, aircraft, space, life-saving, or life-sustaining applications, nor in products or systems where failure or malfunction could result in personal injury, death, or significant property or environmental damage. NSING's Products that are not specifically designated as "automotive grade" may be used in automotive applications only at the user's own risk. Overall, it is important to use NSING's Products only in the manner specified in the product documentation and as explicitly approved by an authorized NSING's representative in writing.