STMICROELECTRONICS

COMMON CRITERIA FOR IT SECURITY EVALUATION

TRUSTED PLATFORM MODULE

ST33TPHF2XSPI

TPM FIRMWARE
1.771

ST33TPHF2XI2C

TPM FIRMWARE
2.771

SECURITY TARGET

DOCUMENT REVISION

Version	Date	Author	Modifications
01.00	5/May/2022	Olivier Collart	Firmware 1.769 for ST33HTPH2XSPI Update Strict conformance to Protection profile TPM 2.0 v1.3 with optional package “ECDAA PP-Module” supported
01.01	13/Jun/2022	Olivier Collart	Correct references to TPM 2.0 PP v1.3
01.02	23/Sep/2022	Olivier Collart	Update user documentation reference
02.00	12/May/2026	Olivier Collart	Scope ST33TPHF2XSPI firmware 1.771 and ST33TPHF2XI2C firmware 2.771 CC Conformance claim changed to CC:2022 Protection Profile claim changed to Protection Profile PC Client Specific TPM 2.0, Version 2.0 certified under reference [EUCC-3090-2026-26]
02.01	21/May/2026	Olivier Collart	Include CC Evaluation lab review comments
02.01p	21/May/2026	Olivier Collart	Public release

Table of Contents

List of Tables

List of Figures

1 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 Security target reference

This security target is referenced with the following information:

- Filename: ST33TPHF2X_ST
- Revision: 02.01p
- Internal documentation system reference: SSS_ST33TPHF2X_ST_22_001
- Date: 21 May 2026

The security target describes the target of evaluation (TOE) named ST33TPHF2X and provides a product summary.

1.2 Target of evaluation reference

Table 1: Target of evaluation: ST33TPHF2X reference

Product	Hardware Commercial Product	Hardware maskset & Hardware Version (External/Internal)	Firmware Version Major.Minor (decimal)
ST33TPHF2XSPI	ST33HTPH	Maskset: **K8KA0** External version: A Internal version: C	0x00 0x01.0x03 0x03 (decimal 1.771)
ST33TPHF2XI2C			0x00 0x02.0x03 0x03 (decimal 2.771)

The firmware major and minor versions may be retrieved from the TOE when loaded on hardware platform with the command TPM2_GetCapability [10], in the response field TPM_PT_FIRMWARE_VERSION_1.

The firmware major and minor versions from the TOE in firmware data delivery format can be retrieved from the binary file from the second to the fifth byte.

For both cases, the firmware version is formatted with the value 0x00 0x01 0x03 0x03 or 0x00 0x02 0x03 0x03 according to [14], Table 1.

Some tools may report the version in decimal value. In that case, the version retrieved is 1.771 or 2.771

1.3 Target of evaluation overview

The product ST33TPHF2XSPI and ST33TPHF2XI2C are TPM 2.0 products targeting PC, server platforms and embedded systems.

The product ST33TPHF2XSPI implements an SPI interface as defined in [14].

The product ST33TPHF2XI2C implements an I²C interface as defined in [14].

The product interface is configured by the firmware and is irreversibly locked after the first firmware factory loading.

The TOE implements the functions defined in the TCG Trusted Platform Module Library Specification, version 2.0 revision 1.59, [9], [10], [11], [12] and the PC Client Specific Platform TPM Profile for TPM 2.0 [14]. The TCG Trusted Platform Module Library specification describes the design principles, the TPM structures, the TPM commands and supporting routines for the commands. The PC Client Specific Platform TPM Profile for TPM 2.0 specification describes the additional features and communication interfaces that must be implemented by a TPM for a PC Client platform.

1.3.1 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 correctly performed 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 behaviour 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 misbehaviour 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 ST33TPHF2X is shipped with EKs and for each EK, a Certificate of the Authenticity of this EK is also provided. The EK may be 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 builds 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 a 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

The TOE provides cryptographic services for hashing, asymmetric encryption and decryption, asymmetric signing and signature verification, symmetric encryption and decryption, symmetric signing and signature verification by means of and key generation. Hash functions SHA-1, SHA-256, SHA_384, SHA3_256 and SHA3_384 are provided as cryptographic service to external entities for measurements and used internally for user authentication, signing and key derivation. A TOE is required to implement asymmetric algorithms, where the current specification supports RSA with 2048 and 3072 bits for digital signature, secret sharing and encryption and ECC algorithms with P-256, P-384 and BN-256 curves for digital signatures and secret sharing. The TOE provides symmetric encryption and decryption of AES-128 192 and 256 in CFB, CTR, OFB, CBC and ECB modes. The TOE implements symmetric signing and signature verification by means of HMAC. 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.

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 authorised 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 TSF also includes random number generation, self-test and physical protection.

Generation and import of the Endorsement key pair and certificate

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 three Endorsement keys

- One 2048-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 by a HSM (Hardware Security Module) and then stored encrypted on a key server.

The Endorsement Key certificate is generated also by a HSM that stores the STMicroelectronics intermediate CA (Certification Authority) keys. The certificates are stored on a certificate server. CA keys are stored outside the HSM in backup encrypted with a 3-DES key. This backup key is generated under dual control by 3 different security officers.

The RSA EK is certified by an intermediate CA 2048-bit key.

The ECC_NIST_P256 EKs and ECC_NIST_P384 EKs are certified by two specific intermediate CAs using a NIST_P384 key.

These certificates comply with the templates defined in the TCG specification for TPM 2.0 EK certificates [15].

The importation of the EK and EK certificate in the TOE is done by the personalization infrastructure that requests EK and EK certificate to the key and certificate servers. The personalization infrastructure decrypts the EK private key and writes it encrypted on the chip with the EK certificate.

The key server, certificate server, HSM and the personalization infrastructure are all located within the secure production area of the TOE.

The STMicroelectronics intermediate certificates are described in a document publicly available [44].

Ecosystem integration

The TPM 2.0 is compliant with PC operating systems like Microsoft® Windows® and Linux that support communication drivers and built in use cases for platform integrity and attestation.

A TPM software stack can be also integrated at host level to facilitate the development of application use cases leveraging the TPM services.

The host drivers and software stack needed to build a TCG compliant trustworthy system are not part of the TOE.

1.4 Target of evaluation description

The TOE consists of

- TPM hardware,
- TPM firmware,
- TPM guidance documentation.

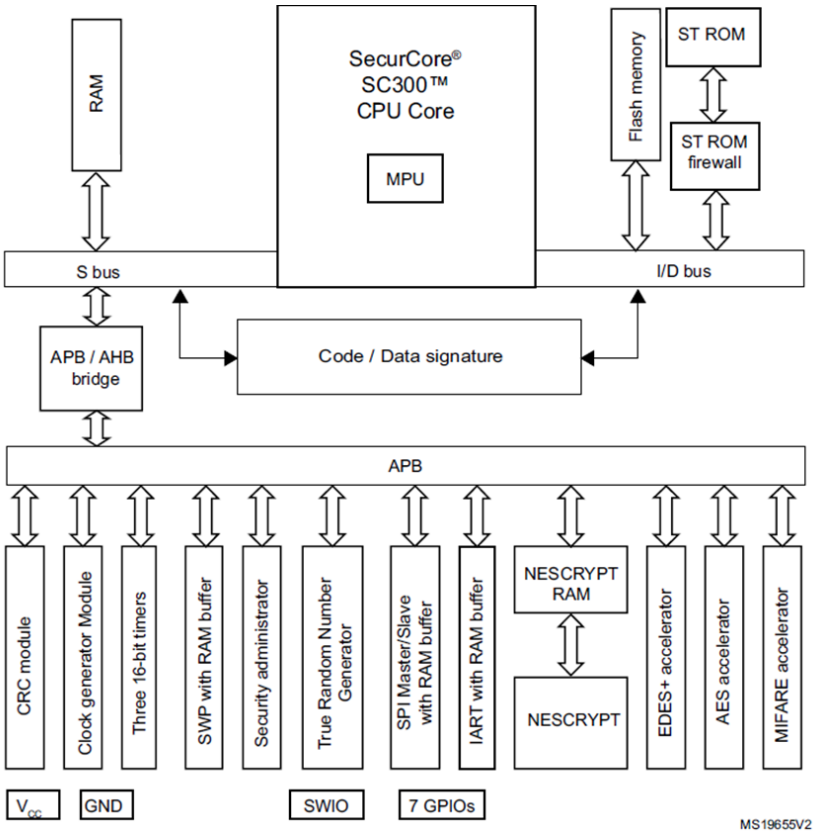
The TOE components are described in the following sections.

1.4.1 TOE hardware description

The TOE includes a hardware called ST33HTPH in this document.

The ST33HTPH is a serial access microcontroller designed for Trusted Platform Module applications that incorporates the generation of ARM processors for embedded secure systems. Its SecurCore® SC300™ 32-bit RISC core is built on the Cortex™ M3 core with additional security features to help to protect against advanced forms of attacks.

Figure 1: ST33HTPH block diagram



The ST33HTPH hardware supports an SPI interface compliant with [PTP 1.05] for integration with controllers and system drivers.

Hardware includes general purpose Input/Output pins to support a driver implementation compliant with the I²C interface and the protocol defined in the TCG standard [PTP 1.05].

Hardware includes the following security features:

- Active shield
- Memory protection unit (MPU)
- Monitoring of environmental parameters through security sensors

- Code/Data Signature for Protection against fault attacks
- ISO 3309 CRC calculation block
- AIS-31 Class PTG2 compliant true random generator (TRNG)
- the EDES peripheral provides a secure DES (Data Encryption Standard) algorithm implementation,
- the AES peripheral provides a secure AES (Advanced Encryption Standard) algorithm implementation, and
- the NESCRYPT crypto-processor efficiently supports the public key algorithm.
- Three timers for TPM Clock and TPM Time management
- The ST ROM is located in non-volatile memory protected by a firewall. This ST firmware includes:
  - A test program used to validate the TOE production (OST)
  - A set of boot and flash management services

1.4.2 *TOE firmware description*

The firmware architecture “F2X” is common to all products included in the TOE. The FW is composed of three independent blocks:

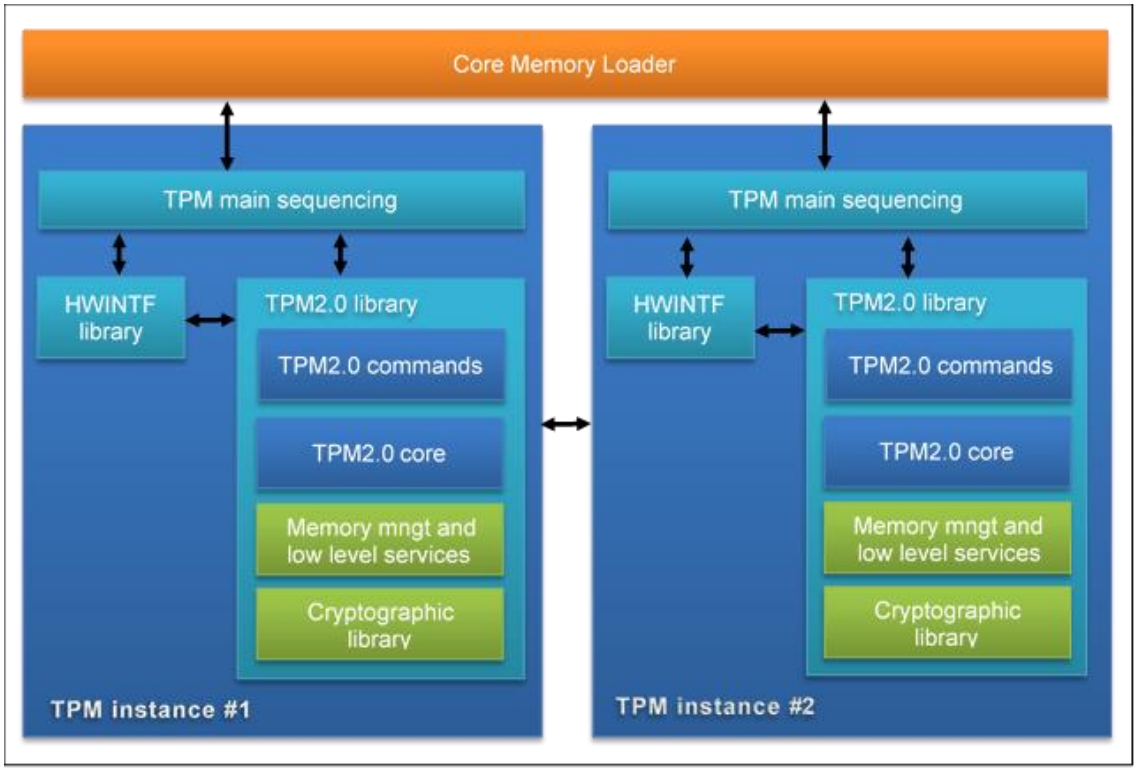
- A non-upgradable code block located in ROM & flash memories (orange box)
  - Core memory loader (CML) in charge of verifying integrity of the TPM instance to be executed.
- Two independent code blocks upgradable via secure field upgrade mechanism (TPM instances #1 and #2). They are composed of:
  - TPM2.0 commands code
  - TPM2.0 core
  - Memory management and low-level services
  - Cryptographic library (NesLib 6.5 for ST33)

From the two code block instances, only one is executed.

The two-instance code architecture provides two resilience features.

- Fault tolerant firmware upgrade: if the firmware loading process is interrupted, the loading instance remains fully functional. The TPM doesn’t enter any limited mode.
- Self-recovery: in case of firmware integrity error of one instance, the second instance becomes active.

Figure 2: F2X firmware block diagram



1.4.3 *TOE guidance documentation*

The following documents must be used by the TOE user in order to configure and operate the TOE.

Table 2: User Documentation

User Documentation	Version	Date	Ref
TPM Library Part 1: Architecture, Specification Version 2.0, Revision 1.59	Revision 1.59	November 8, 2019	[9]
TPM Library Part 2: Architecture, Specification Version 2.0, Revision 1.59	Revision 1.59	November 8, 2019	[10]
TPM Library Part 3: Architecture, Specification Version 2.0, Revision 1.59	Revision 1.59	November 8, 2019	[11]
TPM Library Part 4: Architecture, Specification Version 2.0, Revision 1.59	Revision 1.59	November 8, 2019	[12]
Errata version 1.7 for TCG TPM library version 2.0 revision 1.59	version 1.7	June 10, 2025	[13]
TCG PC Client Specific Platform TPM Profile for TPM 2.0 (PTP), Family “2.0”, Version 1.05 revision 14	Version 1.05	September 4, 2020	[14]
ST33TPHF2XSPI datasheet: Flash-based TPM 2.0 device	Revision 13	February 3, 2026	[43]
ST33TPHF2XI2C datasheet: Flash-based TPM 2.0 device	Revision 6	February 12, 2026	[44]
ST33TPHF2X/ST33GTPMA/ST33GTPMI - Security recommendations	V2.1	September 16, 2022	[46]

1.4.4 *Forms of delivery*

The TOE is delivered in form of firmware binary signed with the TPM firmware production signing key and the guidance documentation. The TOE is finished and the extended test or diagnostic features are irreversibly disabled.

The TOE guidance documentation listed in Table 2 is delivered as electronic file in *.pdf format.

The guidance document released by the Trusted Computing Group are available through download from the website *https://trustedcomputinggroup.org/*

The guidance documents [43], [44] and [46] and the TPM firmware binary are provided by email or secure ftp server upon user request or ST product change notification.

1.5 Target of evaluation lifecycle

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

- phase 1 “Development” and
- phase 2 “Manufacturing”

as defined in the PP [16].

The phase 1 that includes TPM development involves the sites of

- ST ROUSSET (FRANCE)
- ST ANGMO KIO 1 (SINGAPORE)

for the hardware development activities and

- ST ROUSSET (FRANCE)
- ST RENNES (FRANCE)
- ST ZAVENTEM (BELGIUM)

for the embedded software development activities.

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

- ST CROLLES (FRANCE) (Manufacturing)
- ST ROUSSET (FRANCE) (Test Manufacturing and EK/EK certificate injection)
- ST TOA PAYOH (SINGAPORE) (Test Manufacturing and EK/EK certificate injection)

The phase 2 ends with the delivery of the TOE.

2 CONFORMANCE CLAIM (ASE_CCL)

2.1 CC Conformance Claim

This security target is **conformant** to the Common Criteria version CC:2022 Revision 1 as follows.

- Part 2 **conformant**
- Part 3 **conformant and**
- Part 5 **conformant**.

The extended Security Function Requirement is the one defined in the protection profile.

2.2 PP Claim

This security target is in **strict conformance** to the Protection Profile PC Client Specific Trusted Platform Module Specification Family 2.0, Version 2.0 [16].

That protection profile is compliant with Common Criteria CC:2022 and requires EAL4+ assurance level. The assurance level is augmented with assurance components AVA_VAN.4, ALC_FLR.2 and ALC_DVS.2.

The protection profile is registered and certified by the “Agence Nationale de la Sécurité des Systèmes d’Information” (ANSSI) under the reference [EUCC-3090-2026-26].

2.3 Package claim

2.3.1 Assurance package

This ST is conforming to the following assurance packages defined in CC Part 5 [5]

- EAL package EAL4+ augmented with
  - ALC_DVS.2
  - ALC_FLR.2 and
  - AVA_VAN.5

2.3.2 Functional package

This security target claims conformance to an optional functional package “ECDAA” defined in the PP [16].

The functional package has been directly merged in this ST: specific requirements are not explicitly referenced as being part of the optional functional package “ECDAA”.

2.4 Conformance Rationale

This security target claims **strict conformance** to only one PP.

The Target of Evaluation (TOE) is a complete solution implementing the TCG Trusted Platform Module main specifications Version 2.0 level 0 revision 1.59 ([9], [10], [11] and[12]) and the TCG PC Client Specific Platform TPM Profile Specification, Version 1.05 [14] as defined in the PP [16] section 2.2.1. So, the TOE is **consistent** with the **TOE type** in the PP[16].

The **security problem** definition of this security target is **consistent** with the statement of the security problem definition in the PP [16], as the security target claims strict conformance to the PP [16] and no other threats, organizational security policies and assumptions are added.

The **security objective**s of this security target are **consistent** with the statement of the security objectives in the PP as the security target claims strict conformance to the PP and no other security objectives are added.

The **security requirements** of this security target are **consistent** with the statement of the security requirements in the PP [16] as the security target claims strict conformance to the PP [16]. All assignments and selections of the security functional requirements are done in the PP [16] and in this security target section 6.2.

2.5 Application notes

The evidences that the PP [16] is compliant with the application note [42] released by the ANSSI (French CC Certification scheme) and defining security requirements for post-delivery code loading are provided in this security target.

The functional requirement FCS_RNG.1 is a refinement of the FCS_RNG.1 defined in the PP [16] according to —Anwendungshinweise und Interpretationen zum Schema (AIS) respectively - Functionality classes for random number generators [40].

3 SECURITY PROBLEM DEFINITION (ASE_SPD)

The contents of the PP [16] applies to this chapter without any restriction or addition.

3.1 Assets

The assets of the TOE are defined in the PP [16] sections 5.1 and 9.3.1 Assets. These assets have to be protected while being executed as well as when the TOE is not in operation.

3.2 Threats

The threats to security are defined in the PP [16], sections 5.2 and 9.3.2 Threats.

The following threats are added in the security target. Those threats are thwarted by the security objectives defined in the ANSSI Note 6 [42]

Table 3: Threats added in the SPD

Threats	Description
T.Update-Forging	An attacker forges a rogue malicious update that is installed or processed by the TOE, altering the intended TSF functionality.
T.Update-Rollback	An attacker uses the update mechanism to install an update containing a version of the TOE code that is older than the one installed in the initial TOE. This could potentially allow an attacker to replace the initial TOE code with older and vulnerable firmware data.

3.3 Organisational Security Policies

The organisational security policies are defined in the PP [16], sections 5.3 and 9.4 Organisational Security Policies, no other organisational security policies are added

3.4 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 TOE assumptions to the IT environment are defined in the PP [16], section 5.4 and 9.5 Assumptions, no other assumptions are added.

4 SECURITY OBJECTIVES

This section shows the security objectives which are relevant for the TOE. For this section the PP [16] can be applied completely.

4.1 Security Objectives for the TOE

The security objectives of the TOE are defined and described in the PP [16], sections 6.1 and 9.6.1 Security Objectives for the TOE.

The following security objectives from the Note 6, “Security requirements for post-delivery code loading” [42] released by ANSSI are added in the TOE security objectives.

Table 4: ANSSI Note 6 Security objectives

Security Objectives Note 6	Description
O.Secure_Load_ACode	The Loader of the Initial TOE shall check an evidence of authenticity and integrity of the loaded Additional Code. The Loader enforces that only the allowed version of the Additional Code can be loaded on the Initial TOE. The Loader shall forbid the loading of an Additional Code not intended to be assembled with the Initial TOE. During the Load Phase of an Additional Code, the TOE shall remain secure.
O.Secure_AC_Activation	Activation of the Additional 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 Final TOE shall be completed before activation. If the Atomic Activation is successful, then the resulting product is the Final TOE, otherwise (in case of interruption, or incident which prevents the forming of the final TOE), the Initial TOE shall remain in its initial state or fail secure
O.TOE_Identification	The Identification Data identifies the Initial TOE and Additional Code. The TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data. After Atomic Activation of the Additional Code, the Identification Data of the Final TOE allows identifications of the initial TOE and Additional Code. The user shall be able to uniquely identify Initial TOE and Additional Code which are embedded in the Final TOE.

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are described in the PP [16], sections 6.2 and 9.6.2 Security Objectives for the Operational Environment, no other security objectives for the operational environment are added

4.3 Security Objective Rationale

The security objectives rationale is described in the PP [16], sections 6.3 and 9.6.3 Security Objective Rationale. The ANSSI Note 6 security objectives rationale is described in the following table.

Table 5: ANSSI Note 6 Security objectives rationale

	O.Secure_Load_ACode	O.Secure_AC_Activation	O.TOE_Identification
T.Update-Forging	X		
T.Update-Rollback	X	X	X

The threat **T.Update-Forging** is countered by the following security objectives:

- O.Secure_Load_ACode: the check of the integrity and authenticity of the loaded additional code thwarts the possibility for an attacker to forge a malicious Additional Code that could be loaded on a genuine initial TOE.

The threat **T.Update-Rollback** is countered by the following security objectives:

- O.Secure_Load_ACode: this objective enforces that only the allowed versions of the Additional Code can be loaded on the Initial TOE preventing rollback to versions not allowed and potentially with known vulnerabilities.
- O.Secure_AC_Activation: this objective enforces that if the activation is successful and atomic, the resulting product is the Final TOE, or (in case of interruption, or incident which prevents the forming of the final TOE), the Initial TOE that remains in its initial state or fail secure. This prevents to bypass the checks enforced by O.Secure_Load_ACode.
- O.TOE_Identification: this objective enforces that the check on the versions of the Additional code and the initial TOE is reliable. It also enforces that the state of the final TOE can be reported after additional code loading to confirm whether the activation is successful or not.

5 EXTENDED COMPONENTS DEFINITION (ASE_ECD)

No other extended component is added in this security target.

6 SECURITY REQUIREMENTS (ASE_REQ)

6.1 Security Functional Requirements listed by the TPM 2.0 Protection Profile

The security functional requirements (SFRs) for the TOE are defined in the PP [16] section 8.1. All 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 following convention has been followed for **assignment** operation used to assign a specific value to an unspecified parameter and for selection operation used to select one or more options:

- Underlined text shows assignment or selection done in the protection profile
- *Underlined text italicised* shows assignment or selection done in the security target.

The refinement operation used to add detail to a requirement is denoted with **bold text**.

6.2 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, PCRs, NV storage areas, counters and firmware*.

FCS_RNG.1 Random number generation

Hierarchical to: No other components.
Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a *deterministic*[1] random number generator that implements: *NIST SP 800-90A Hash_DRBG*. [33][2].

FCS_RNG.1.2 The TSF shall provide octets of bits[3] that meet: *Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG*[4].

In order to comply with the requirements defined in the standard AIS 20 [40], a refinement of the SFR FCS_RNG is provided below:

FCS_RNG.1 Random number generation

Hierarchical to: No other components

Dependencies: No dependencies

FCS_RNG.1.1 The TSF shall provide a *deterministic* random number generator *AIS20 Class DRG.3 according* to [40] that implements:

[1] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]
[2] [assignment: *list of security capabilities*]
[3] [selection: bits, octets of bits, numbers [assignment: format of the numbers]]
[4] [assignment: *a defined quality metric*]

(DRG.3.1) if initialized with a random seed using a PTRNG of class PTG.2 as random source, the internal state of the RNG shall have at least 100 bit of min-entropy and implements NIST SP 800-90A *Hash_DRBG* [33] and FIPS 180-4 [26].

(DRG.3.2) The RNG provides forward secrecy

(DRG.3.3) The RNG provides backward secrecy even if the current internal state is known

FCS_RNG.1.2 The TSF shall provide random numbers that meet

(DRG.3.4) *The RNG initialized with a random seed before the first use of the RNG after each product power up and reseeded after 2^{20} requests generates output for more than 2^{34} strings of bit length 128 that are mutually different with probability of $w>1-2^{-16}$*

(DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass FIPS 140-3 statistical test suite.

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

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_CKM.5 Cryptographic key derivation, or
FCS_COP.1 Cryptographic operation]
[FCS_RBG.1 Random bit generation, or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of 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 and 3072 bits* that meet the following: TPM library specification *[9], [10], [11], [12] in combination with [SP800-108], and [IEEE1363], [RFC 3447]*.

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

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_CKM.5 Cryptographic key derivation, or
FCS_COP.1 Cryptographic operation]
[FCS_RBG.1 Random bit generation, or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of 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 *[9], [10], [11], [12], in combination with [SP800-108]*.

FCS_CKM.1/PKAES Cryptographic key generation (primary keys)

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_CKM.5 Cryptographic key derivation, or
FCS_COP.1 Cryptographic operation]
[FCS_RBG.1 Random bit generation, or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of cryptographic key destruction]

FCS_CKM.1.1/PKAES 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, 192 & 256 bits,* that meet the following: TPM library specification *[9], [10], [11], [12] in combination with [SP800-108]*, .

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

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_CKM.5 Cryptographic key derivation, or
FCS_COP.1 Cryptographic operation]
[FCS_RBG.1 Random bit generation, or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of cryptographic key destruction]

FCS_CKM.1.1/RSA The TSF shall generate cryptographic **RSA** keys in accordance with a specified cryptographic key generation algorithm *RSA key generator* and specified cryptographic key sizes *2048 and 3072 bits* that meet the following: TPM library specification *[9], [10], [11], [12], [RFC 3447] and [IEEE1363]*.

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

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_CKM.5 Cryptographic key derivation, or
FCS_COP.1 Cryptographic operation]
[FCS_RBG.1 Random bit generation, or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of 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 *[9], [10], [11], [12]*.

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

Hierarchical to: No other components.
Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_CKM.5 Cryptographic key derivation, or
FCS_COP.1 Cryptographic operation]
[FCS_RBG.1 Random bit generation, or
FCS_RNG.1 Generation of random numbers]
FCS_CKM.6 Timing and event of 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, 192 & 256 bits* that meet the following: *TPM library specification [9], [10], [11], [12]*.

FCS_CKM.6 Timing and event of 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.6.1 The TSF shall destroy *cryptographic keys* when no longer needed.

FCS_CKM.6.2 The TSF shall destroy *cryptographic keys and key material* specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method: *key overwriting and NV memory zeroization* that meets the following: *none*.

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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of 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, *CTR, OFB, CBC and ECB* and cryptographic key sizes 128*, 192 and* 256 bits that meet the following: [FIPS 197] and [SP 800-38A]

FCS_COP.1/SHA Cryptographic operation (hash function)

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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1/SHA The TSF shall perform hash value calculation in accordance with a specified cryptographic algorithm SHA-256, SHA-384 and *SHA-1*[5] cryptographic key sizes none that meet the following: ISO/IEC 10118-3 [21] and FIPS 180-4 [25].

FCS_COP.1/SHA3 Cryptographic operation (hash function)

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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1/SHA3 The TSF shall perform *hash value calculation* in accordance with a specified cryptographic algorithm *SHA3-256 and SHA3-384* and cryptographic key sizes *none* that meet the following: *FIPS 202*.

FCS_COP.1/HMAC Cryptographic operation (HMAC calculation)

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 or

[5] [assignment: *list of additional algorithms*]

FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of 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-256, SHA-384 and *SHA1*[6] and cryptographic key sizes *160, 256 and 384 bits* that meet the following: [FIPS 198-1] [28].

FCS_COP.1/HMAC/SHA3 Cryptographic operation (HMAC calculation)

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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1/HMAC/SHA3 The TSF shall perform *HMAC value generation and verification* in accordance with a specified cryptographic algorithm *HMAC with SHA3-256 and SHA3-384* and cryptographic key sizes *256 and 384 bits* that meet the following*: [FIPS 198-1]* [28].

FCS_COP.1/RSAED Cryptographic operation (asymmetric 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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of 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 bit and 3072 bit that meet the following: PKCS#1v2.1 [39].

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

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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1/RSASign The TSF shall perform signature generation and verification in accordance with a specified cryptographic algorithm RSASSA_PKCS1v1_5, RSASSA_PSS and cryptographic key sizes 2048 bit and 3072 bit that meet the following: PKCS#1v2.1 [RFC 3447].

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

[6] [assignment: *list of additional algorithms*]

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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of 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 curves TPM_ECC_NIST_P256, TPM_ECC_NIST_P384 and *TPM_ECC_BN_P256* and cryptographic key sizes 256 and 384 bits that meet the following: FIPS PUB 186-5 [26] and NIST SP800-186 5.

FCS_COP.1/ECDAA Cryptographic operation (ECDAA commit)

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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_COP.1.1/ECDAA The TSF shall perform signature generation in accordance with a specified cryptographic algorithm ECDAA with curve *TPM_ECC_NIST_P256,* TPM_ECC_BN_P256 *and TPM_ECC_NIST_P384* and cryptographic key sizes 256 *and 384* that meet the following: [FIPS 186-5] for curves *TPM_ECC_NIST_P256 and TPM_ECC_NIST_P384 and* [ISO/IEC 15946-5] for curve *TPM_ECC_BN_P256.*

FCS_COP.1/ECDEC Cryptographic operation (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, or
FCS_CKM.5 Cryptographic key derivation]
FCS_CKM.6 Timing and event of 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 TPM_ECC_BN_P256* and cryptographic key sizes 256 and 384 bits that meet the following: TPM library specification *[9], [10], [11], [12] and* [SP 800-56A] *[32].*

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 authorised user “World”
   a) the TPM2_SelfTest command and of selected algorithms using the TPM2_IncrementalSelfTest command,

2. at the conditions
   a) Initialisation 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 authorised users with the capability to verify the integrity of *TSF data*.

FPT_TST.1.3 The TSF shall provide authorised 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) *failure of execution flow control and hardware failure*

FPT_PHP.3 Resistance to physical attack
Hierarchical to: No other components.
Dependencies: No dependencies.

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_ACC.2/States Complete access control (operational states)
Hierarchical to: FDP_ACC.1 Subset access control
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.2.1/States The TSF shall enforce the TPM State Control SFP on all subjects and objects and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2/States 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.

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 initialisation

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 [7] :
(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[8]:
(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 *Platform firmware* is authorised 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 authorised 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 *the TOE is reset after successful loading of the firmware data*.
(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 authorised to process the Resume operation on the Shutdown BLOB with state transition to Operational.
(6) In the Init state every subject is authorised 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 authorised to process the TPM2_Startup command. In case of the parameter TPM_SU_CLEAR the TPM shall change the state to Operational and initialise its internal operational variables to default initialisation values (Reset), otherwise the TPM shall return an error and stay in the same state.
(8) In the Operational state, nobody is authorised 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 authorised 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 case of a successful test result the state shall change to Operational, otherwise to Fail.
(11) In the Fail state, every subject is authorised to execute the commands TPM2_GetTestResult and TPM2_GetCapability.
(12) In the Fail state the subject World is authorised to send a _TPM_Init indication with state change to Init.
(13) Any subject is authorised 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).

[7] See Table 7 in Protection Profile [15]
[8] See Table 8 and 9 in Protection Profile [15]

FDP_ACF.1.3/States The TSF shall explicitly authorise 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

(4) the TOE authorises to enter FUM state if the internal failure counter is strictly lower than the limit failure counter

(5) the TOE resets the upgrade counter once a firmware with a strictly higher version is loaded successfully

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.

FMT_MSA.1/States Management of security attributes (operational states)

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

FMT_MSA.1.1/States TSF shall enforce the TPM state control SFP to restrict the ability to modify the security attributes TPM state

(1) FUM to Platform firmware,
(2) other than FUM to any role.

FMT_MSA.3/States Static attribute initialisation (operational states)

Hierarchical to: No other components.
Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1/States The TSF shall enforce the TPM state control SFP to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/States The TSF shall allow nobody to specify alternative initial values to override the default values when an object or information is created.

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_ACC.1/Hier Subset access control (object hierarchy)

Hierarchical to: No other components.
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP on

Subjects

(1) Platform firmware,
(2) Platform Owner,
(3) Privacy administrator,
(4) Lockout administrator,
(5) USER,
(6) World

Objects

(5) PPS,
(6) EPS,
(7) SPS,
(8) PPO,
(9) EK,
(10) SRK
(11) Null Seed,
(12) object in a TPM hierarchy

Operations

(1) TPM2_CreatePrimary,
(2) TPM2_CreateLoaded
(3) TPM2_HierarchyControl,
(4) TPM2_Clear,
(5) TPM2_ClearControl,
(6) TPM2_HierarchyChangeAuth,
(7) TPM2_SetPrimaryPolicy,
(8) TPM2_Load,
(9) TPM2_LoadExternal,
(10) TPM2_ReadPublic,
(11) Use
(12) TPM2_ChangeEPS
(13) TPM2_ChangePPS
(14) *TPM2_RestoreEK*

FDP_ACF.1/Hier Security attribute based access control (object hierarchy)

Hierarchical to: No other components.
Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP to objects based on the following:

Subjects:

(1) Platform firmware with security attribute authorisation state gained by authentication with platformAuth or platformPolicy,
(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 authValue or authPolicy,
(6) World with no security attributes,

Objects:

(1) EPS,
(2) PPS,
(3) SPS,
(4) EK,
(5) PPO,

(6) SRK,
(7) Null Seed,
(8) object in a TPM hierarchy with security attributes: state of the hierarchy, fixedParent, fixedTpm

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

(1) The subject World is authorised to create an EPS whenever the TPM is powered on and no EPS is present.
(2) The subject World is authorised to create an PPS whenever the TPM is powered on and no PPS is present.
(3) The subject World is authorised to create an SPS whenever the TPM is powered on and no SPS is present.
(4) The subject World is authorised to create a Null Seed whenever the TPM is reset.
(5) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth is authorised to change the SPS to a new value from the RNG (TPM2_Clear). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_Clear command.
(6) The Platform firmware is authorised to create a Platform Primary Object under PPS. The physical presence is not required if it is not if supported by the TOE or disabled for TPM2_CreatePrimary or TPM2_CreateLoaded command.
(7) The Platform Owner is authorised to create a primary object (SRK) under SPS.
(8) The privacy administrator is authorised to create a primary object (EK) under EPS.
(9) The subject World is authorised to create temporary objects for no hierarchy (using the Null Seed).
(10) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth are authorised to remove all TPM context associated with a specific owner (TPM2_Clear). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_ClearControl command.
(11) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth are authorised to disable and enable the execution of TPM2_Clear by the command TPM2_ClearControl. The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_ClearControl command.
(12) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE, the Platform Owner, the privacy administrator and the lockout administrator are authorised to change the authorisation secret for a hierarchy or lockout (TPM2_HierarchyChangeAuth). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyChangeAuth command.
(13) The Platform firmware with platformAuth, platformPolicy or physical presence, if supported by the TOE the Platform Owner and the privacy administrator are authorised to set the authorisation policy for the platform hierarchy (platformPolicy), the storage hierarchy (ownerPolicy) and the endorsement hierarchy (endorsementPolicy) using the command TPM2_SetPrimaryPolicy. The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_SetPrimaryPolicy command.
(14) The Platform firmware is authorized to replace the current EPS with a value from RNG, disable EKs loaded by the TPM Vendor and to set the endorsement hierarchy controls to their default values (TPM2_ChangeEPS).
(15) The Platform firmware is authorized to replace the current PPS with a value from RNG and to set the platformPolicy to the default value (TPM2_ChangePPS)
(16) The Platform firmware is authorized to replace the current EPS with a value from RNG, to restore the EKs loaded by the TPM vendor and to set the endorsement hierarchy controls to their default values (TPM2_RestoreEK). The EKs are restored from the EKs values loaded by the TPM vendor in phase 2 (manufacturing and

delivery) defined for case 1 in the Protection Profile [15] . The restored values are used to generate the EKs when the command TPM2_CreatePrimary uses the default creation templates defined in the TOE user guidance

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

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

(1) No subject is authorised to use any object of a hierarchy if the corresponding hierarchy is disabled (i.e phEnable for platform hierarchy is CLEAR, shEnable for Storage hierarchy is CLEAR, ehEnable for EPS hierarchy is CLEAR).

FMT_MSA.1/Hier Management of security attributes (object hierarchy)

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

FMT_MSA.1.1/Hier TSF shall enforce the TPM Object Hierarchy SFP to restrict the ability to modify the security attributes fixedTPM and fixedParent to nobody.

FMT_MSA.3/Hier Static attribute initialisation (object hierarchy)

Hierarchical to: No other components.
Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1/Hier The TSF shall enforce the TPM Object Hierarchy SFP to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Hier The TSF shall allow 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.

FMT_MSA.4/Hier Security attribute value inheritance (hierarchy)

Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FMT_MSA.4.1/Hier The TSF shall use the following rules to set the value of security attributes:

(1) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE is authorised to enable and to disable the use of the platform hierarchy and its associated NV storage (TPM2_HierarchyControl changing phEnable or phEnableNV). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyControl command.

(2) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and Platform Owner with ownerAuth or ownerPolicy are authorised to enable and to disable the use of a Storage hierarchy (TPM2_HierarchyControl changing shEnable). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyControl command.

(3) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and privacy administrator with endorsementAuth or endorsementPolicy are authorised to enable and to disable the use of a Endorsement hierarchy (TPM2_HierarchyControl changing ehEnable). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyControl command.

(4) The only way to enable platform hierarchy is power-on of the TPM.

(5) The Platform firmware with platformAuth, platformPolicy, or physical presence if supported by the TOE is authorised to enable the use of the Endorsement hierarchy and the Storage hierarchy (TPM2_HierarchyControl). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyControl command

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 initialisation

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 firmware 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 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 and restartCount and safe (TPM2_ReadClock).
(4) A subject with the role USER endorsed by the Privacy administrator or 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 authorise 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*.

6.3 Security assurance requirements

The security assurance requirements defined in Table 6 are defined in section 8.2 of the PP [15] with the exception of the vulnerability assessment assurance component augmented to AVA_VAN.5 whereas the PP [15] mandates AVA_VAN.4.

Table 6: Security assurance requirements for the TOE

Assurance Class	Assurance components
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.2 Suffiency of security measures- **augmented**
	ALC_LCD.1 Developer defined life-cycle model
	ALC_FLR.2 Flaw reporting procedures - **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.5 Advanced vulnerability analysis - **augmented**

6.4 Security Requirements rationale

The security requirements rationale of the TOE is defined and described in the PP [15], sections 8.3 and 9.8 Security Requirements rationale.

6.4.1 Sufficiency of SFR

The SFRs FCS_CKM.1/PKRSA, FCS_CKM.1/PKECC and FCS_CKM.1/PKAES fulfil the same objectives as the SFR FCS_CKM.1/PK defined in the PP [15] Table 11.

The SFR FCS_COP.1/SHA3 fulfils the same objectives as the SFR FCS_COP.1/SHA defined in the PP [15] Table 11.

The SFR FCS_COP.1/HMAC/SHA3 fulfils the same objectives as the SFR FCS_COP.1/HMAC defined in the PP [15] Table 11.

For the additional security objectives described in Table 4, the following table provides the security requirements rationale.

Table 7: Security requirements rationale for Security objectives from [42]

Security Objectives	SFR
O.Secure_Load_ACode	Covered by SFR FDP_ACF.1.2/States, iteration 2 from PP [16] Covered by SFR FDP_ACF.1.3/States iterations 1 & 2 from this security target
O.Secure_AC_Activation	Covered by SFR FDP_ACF.1.2/States iteration 3
O.TOE_Identification	Covered by SFR FCO_NRO.1.2/M&R iteration 6

6.4.2 Dependency rationale

The SFRs FCS_CKM.1/PKRSA, FCS_CKM.1/PKECC and FCS_CKM.1/PKAES fulfil the same dependency rationale as the SFR FCS_CKM.1/PK defined in the PP [15] Table 12.

The SFR FCS_COP.1/SHA3 fulfils the same dependency rationale as the SFR FCS_COP.1/SHA defined in the PP [15] Table 12.

The SFR FCS_COP.1/HMAC/SHA3 fulfil the same dependency rationale as the SFR FCS_COP.1/HMAC defined in the PP [15] Table 12.

6.5 Security Assurance rationale

The security assurance requirements rationale of the TOE is defined and described in the section 8.3.3 Assurance rationale of the PP [15].

For this evaluation, the evaluation vulnerability assessment assurance component has been augmented to AVA_VAN.5 whereas the PP [15] mandates AVA_VAN.4. This vendor choice is a willingness to increase the resistance of the TOE to attacks categorized at level “High attack potential” compared to the level “Moderate attack potential” required by AVA_VAN.4.

7 TOE SUMMARY SPECIFICATION

The product overview is described in section 1.3.

In the following section, the security functionality and the assurance measures of the TOE are described.

7.1 TOE Security Features

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

7.1.1 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, ECC and AES data encryption and decryption, 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 and 3072 bits that meet the following: [37] and optional [35] and ECC with key sizes of 256 and 384 bits that meet *[9], [10], [11], [12]*,and optional [35].

RSA key generator:

- Endorsement Key generated with default template defined in [15] is securely written in the TOE during the manufacturing process
- Other keys are generated according to *[9], [10], [11], [12] using the DRBG as random generator*

ECC key generator

- Endorsement Key generated with default template defined in [15] is securely written in the TOE during the manufacturing process
- Other keys are generated according to *[9], [10], [11], [12] using the DRBG as 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, 192 and 256 bits that meet *[9], [10], [11], [12]* and optional [35].

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

The TOE supports the destruction of cryptographic keys by erasure of volatile memory areas containing cryptographic keys.

The covered security functional requirement is FCS_CKM.6.

The TOE performs the encryption and decryption in accordance with the specified cryptographic algorithm AES in the CFB, CTR, OFB, CBC, ECB modes and cryptographic key size of 128, 192 and 256 bits that meet [FIPS 197] and [SP 800-38A].

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 that meets [FIPS 180-4] beside SHA3-256 and SHA3-384 that meet [FIPS 202] .

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, SHA-384, SHA3-256 and SHA3-384 and cryptographic key sizes 160, 256 and 384 bits that meet [FIPS 198-1] and [FIPS 180-4]

The covered security functional requirements are FCS_COP.1/HMAC and FCS_COP.1/HMAC/SHA3.

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 and 3072 bits that meet [RFC 3447].

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

The TOE performs signature generation and signature verification in accordance with the specified cryptographic algorithm RSASA_PKCS1v1_5, RSASSA_PSS and cryptographic key sizes 2048 and 3072 bits that meet [RFC 3447].

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 curves TPM_ECC_NIST_P256, TPM_ECC_NIST_P384 and TPM_ECC_BN_P256 and cryptographic key sizes 256 and 384 bits that meet TPM library specification [9] section C.4.

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

The TOE performs signature generation in accordance with the specified cryptographic algorithm ECDAA with curves TPM_ECC_NIST_P256, TPM_ECC_NIST_P384 and TPM_ECC_BN_P256 and cryptographic key sizes 256 and 384 bits that meet TPM library specification, [9] section C4.2.

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

The TOE performs decryption of ECC key in accordance with the specified cryptographic algorithm ECDH with curves TPM_ECC_BN_P256, TPM_ECC_NIST_P256 and TPM_ECC_NIST_P384 and cryptographic key sizes 256 and 384 bits that meet TPM library specification *[9], [10], [11], [12] and* [SP 800-56A] section 6.1.1.2.

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 random numbers that fulfils the requirements from the functional class DRG.3 of [AIS 20] and [SP 800-90Ar1]. The DRBG is based on a HASH_DRBG with SHA256.

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/PKAES,
- FCS_CKM.1/RSA,
- FCS_CKM.1/ECC,

- FCS_CKM.1/SYMM,
- FCS_CKM.6,
- FCS_COP.1/AES, FCS_COP.1/SHA,
- FCS_COP.1/HMAC,
- FCS_COP.1/HMAC/SHA3
- FCS_COP.1/RSAED,
- FCS_COP.1/RSASign,
- FCS_COP.1/ECDSA,
- FCS_COP.1/ECDAA,
- FCS_COP.1/ECDEC and
- FCS_RNG.1.

7.1.2 SF_I&A - Identification and Authentication

The TPM provides two mechanisms for the identification and authentication capability to authorize the use of a 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 proof 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 the 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.

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 one unsuccessful authentication attempt occurs 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 user's 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 subjects 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.

7.1.3 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 deallocation 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_ContextLoad 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:

- Protection by an active shield that commands an automatic reaction on die integrity violation detection.
- Preventative mechanisms are implemented in order to mitigate the risk of information disclosure or unauthorized modification
  - Bus encryption
  - Memories scrambling and encryption
  - Mechanisms for operation execution concealment
  - Clock frequency modification and jittering
- Intrinsic countermeasures for cryptographic algorithm against side channel attacks like timing attacks (TA), SPA and DPA.
- Detection of abnormal behavior of the following operational conditions:
  - High voltage supply
  - Glitches
- Detection of abnormal TOE behavior
  - MPU error
  - TRNG failure

The covered security functional requirements are 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_SMF.1,
- FMT_MSA.2,
- FPT_STM.1,
- FDP_RIP.1,
- FPT_TST.1,
- FPT_FLS.1/FS,
- FPT_FLS.1/SD and

- FPT_PHP.3
- FDP_ITT.1
- FPT_ITT.1

7.1.4 SF_OBH - Object Hierarchy

The TOE supports different states during his lifecycle as described in [16] 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

7.1.5 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

7.1.6 *Assignment of Security Functional Requirements*

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	X				
FCS_CKM.1/PKRSA	X				
FCS_CKM.1/PKECC	X				
FCS_CKM.1/PKAES	X				
FCS_CKM.1/RSA	X				
FCS_CKM.1/ECC	X				
FCS_CKM.1/SYMM	X				
FCS_CKM.6	X				
FCS_COP.1/AES	X				
FCS_COP.1/SHA	X				
FCS_COP.1/SHA3	X				
FCS_COP.1/HMAC	X				
FCS_COP.1/HMAC/SHA3	X				
FCS_COP.1/RSAED	X				
FCS_COP.1/RSASign	X				
FCS_COP.1/ECDSA	X				
FCS_COP.1/ECDAA	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_AFL.1/PINPASS		X			
FIA_AFL.1/PINFAIL		X			
FIA_UID.1		X			

FIA_UAU.1		X			
FIA_UAU.5		X			
FIA_UAU.6		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
FTP_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

8 ACRONYMS

For the purposes of this document, the acronyms given in CC Parts 2 and 3 and the following apply.

Acronym	Description
AFL	Application Flash Loader
AuthData	Authentication Data or Authorisation Data, depending on the context
CA	Certificate Authority
CFB	Cipher Feedback mode
CML	Code Memory Loader
CRTM	Core Root of Trust for Measurement
CTR	Counter-mode encryption
DA	Dictionary Attack
DAA	Direct Autonomous Attestation
DRBG	Deterministic Random Bit Generator
EAL	evaluated assurance level
ECB	Electric Cookbook
ECC	Elliptic Curve Cryptography
ECDAA	ECC-based Direct Anonymous Attestation
ECDH	Elliptic Curve Diffie-Hellman
EK	Endorsement Key
EPS	Endorsement Primary Seed
FIPS	Federal Information Processing Standard
FU	Field Upgrade
FUM	Field Upgrade mode
HMAC	Hash Message Authentication Code
HW	Hardware Interface
I/O	Input/Output
IV	Initialisation Vector
KDF	key derivation function
MMIO	Memory Mapped I/O
MPU	Memory Protecting Unit
NIST	National Institute of Standards and Technology
NV	Non-volatile
NVM	Non-Volatile Memory
OAEP	Optimal Asymmetric Encryption Padding
PCR	platform configuration register(s)
PK	Primary Key
PP	Physical Presence, Protection Profile
PPO	Platform Primary Object
PPS	Platform Primary Seed
PRIVEK	Private Endorsement Key
PRNG	Pseudo Random Number Generator
PUBEK	Public Endorsement Key
RNG	Random Number Generator
RSA	Algorithm for public-key cryptography. The letters R, S, and A represent the initials of the first public describers of the algorithm Rivest, Shamir and Adleman.

Acronym	Description
RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
SHA	Secure Hash Algorithm
SPS	Storage Primary Seed
SRK	Storage Root Key
TCB	Trusted Computing Base
TCG	Trusted Computing Group
TOE	Target of Evaluation
TPM	Trusted Platform Module
TPM_	Prefix for a command defined in TPM 1.2 library specifications
TPM2_	Prefix for a command defined in TPM 2.0 library specifications
UTC	Universal Time Clock

Appendix A REFERENCES

The following materials are to be used in conjunction with or are referenced by this document.

[1] [CCMB-2022-11-001]

Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CC:2022, Revision 1, November 2022

[2] [CCMB-2022-11-002]

Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; CC:2022, Revision 1, November 2022

[3] [CCMB-2022-11-003]

Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; CC:2022, Revision 1, November 2022

[4] [CCMB-2022-11-004]

Common Methodology for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, CC:2022, Revision 1, November 2022

[5] [CCMB-2022-11-005]

Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements; CC:2022, Revision 1, November 2022

[6] [CC2022-ERR-R1-V1.2]

Common Criteria for Information Technology Security Evaluation, Errata and Interpretation for CC:2022 (Release 1) Version 1.2

[7] [CCMB-2024-02-001-v1.0]

Common Criteria for Information Technology Security Evaluation, CEM:2022 (Release 1), February 2024

[8] [TCG Glossary]

http://www.trustedcomputinggroup.org/developers/glossary

[9] [TPM2.0 Part1 r159]

TPM Library Part 1: Architecture, Specification Version 2.0, Revision 1.59, November 8, 2019, Trusted Computing Group Incorporated

[10] [TPM2.0 Part2 r159]

TPM Library Part 2: TPM Structures, Specification Version 2.0, Revision 1.59, November 8, 2019, Trusted Computing Group Incorporated

[11] [TPM2.0 Part3 r159]

TPM Library Part 3: Commands, Specification Version 2.0, Revision 1.59, November 8, 2019, Trusted Computing Group Incorporated

[12] [TPM2.0 Part4 r159]

TPM Library Part 4: Supporting Routines, Specification Version 2.0, Revision 1.59, November 8, 2019, Trusted Computing Group Incorporated

[13] [TPM2.0 rev159 Err 1.7]

Errata version 1.7 for TCG TPM library Family “2.0” level 0 revision 1.59, June 10, 2025, Trusted Computing Group Incorporated.

[14] [PTP 1.05]

TCG PC Client Specific Platform TPM Profile for TPM 2.0 (PTP), Family “2.0”, Level 00 Version 1.05 Revision 14, September 4, 2020, Trusted Computing Group Incorporated

[15] [TPM2.0 EK CERT]

TCG EK credential profile for TPM Family 2.0 Level 0. Specification Version 2.3 Revision 2, July 23, 2020, Trusted Computing Group, Incorporated

[16] [TPM2.0 PP]

Protection Profile PC Client Specific Trusted Platform Module Specification Family 2.0, Version 2.0, December 3, 2025, Trusted Computing Group Incorporated Certified under reference [EUCC-3090-2026-26]

[17] [IEEE P1363-2000]

Standard Specifications for Public Key Cryptography, Institute of Electrical and Electronics Engineers, Inc. (note reaffirmation PAR is actual running)

[18] [ISO/IEC 9796-2]

ISO/IEC 9796-2, Information technology – Security techniques – Digital signature scheme giving message recovery – Part 2: Integer factorization-based mechanisms, ISO, 2002.

[19] [ISO/IEC 9797-2]

ISO/IEC 9797-2, Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function

[20] [ISO/IEC 10116]

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

[21] [ISO/IEC 10118-3]

ISO/IEC 10118-3, Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash function

[22] [ISO/IEC 14888-3]

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

[23] [ISO/IEC 18033-3]

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

[24] [FIPS 140-3]

FIPS Publication 140-3, Security requirements for cryptographic modules, March 22, 2019

[25] [FIPS 180-4]

FIPS Publication 180-4, Secure Hash standard, NIST, August 1, 2002

[26] [FIPS 186-5]

FIPS Publication, Digital Signature Standard (DSS), February 3, 2023

[27] [FIPS 197]

FIPS Publication, Advanced Encryption Standard (AES), November 26, 2001

[28] [FIPS 198-1]

FIPS Publication, The Keyed-Hash Message Authentication Code (HMAC), July 2008

[29] [FIPS 202]

FIPS Publication, SHA-3 Standard: Permutation-Based hash and Extendable-Output Functions, August 2015

[30] [SP 800-17]

NIST Special Publication 800-17: Modes of Operation Validation System (MOVS): Requirements and Procedures, February 1998

[31] [SP 800-38A]

NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation. December 2001

[32] [SP 800-56A]

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

[33] [SP 800-90Ar1]

Recommendation for random number generation using deterministic random bit generators, NIST, June 2015

[34] [SP800-107]

NIST Special Publication 800-107: Recommendation for Applications Using Approved Hash Algorithms. August 2012

[35] [SP800-108]

NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions. October 2009

[36] [RFC 2104]

RFC2104 - HMAC: Keyed-Hashing for Message Authentication

[37] [RFC 3447]

IETF RFC 3447, Public key Cryptography Standard, PKCS#1
PKCS#1: v2.0 RSA Cryptography Standard, RSA Laboratories, October 1, 1998
PKCS#1: v2.1 RSA Cryptography Standard, RSA Laboratories, June 14, 2002

[38] [IEEE1363]

IEEE Std1363 – 2000 Standard Specifications for Public Key Cryptography
IEEE Std1363a – 2004 Standard Specifications for Public Key Cryptography

[39] [PKCS#1]

PKCS#1: v2.0 RSA Cryptography Standard, RSA Laboratories, October 1, 1998

[40] [AIS 20]

A proposal for Functionality classes for random number generators Version 3.0 BSI

[41] [RGS B1]

Référentiel Général de Sécurité, version 2.0 Annexe B1. Mécanismes cryptographiques version 2.03 (21/02/2014)

[42] [ANSSI N6]

Application Note 6 – Security requirements for post-delivery code loading, Version 2.0, January 23rd, 2015, ANSSI

[43] [DS ST33TPHF2XSPI]

ST33TPHF2XSPI datasheet: Flash-based TPM 2.0 device with an SPI interface, Firmware 1.771, v13, February 3, 2026, STMicroelectronics

[44] [DS ST33TPHF2XI2C]

ST33TPHF2XI2C datasheet: Flash-based TPM 2.0 device with an I2C interface, Firmware 2.771, v6, February 12, 2026, STMicroelectronics

[45] [EK CERT]

https://www.st.com/resource/en/technical_note/dm00711714-st-trusted-platform-module-tpm-endorsement-key-ek-certificates-stmicroelectronics.pdf

[46] [SCY REC]

ST33TPHF2X/ST33GTPMA/ST33GTPMI - Security recommendations (V2.1), STMicroelectronics

[47] [ISO/IEC 15946-5]

ISO/IEC 15946-5, Information technology — Security techniques – Cryptographic techniques based on elliptic curves – Part 5: Elliptic curve generation; Clause 7.3 (Barreto –Naehrig (BN) elliptic curve)

[48] [ANSSI GMC]

https://www.ssi.gouv.fr/uploads/2021/03/anssi-guide-mecanismes_crypto-2.04.pdf

IMPORTANT NOTICE – PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to *www.st.com/trademarks*. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.