**STMICROELECTRONICS**

# COMMON CRITERIA FOR IT SECURITY EVALUATION

TRUSTED PLATFORM MODULE

## ST33TPHF2ESPI MODE TPM 2.0

TPM FIRMWARE 0X47.0X10

## SECURITY TARGET

**DOCUMENT REVISION**

| Version | Date | Author | Modifications |
|---|---|---|---|
| 01-00 | 06/04/2016 | Olivier Collart | Release |
| 01-01 | 06/05/2016 | Olivier Collart | Update including evaluator comments |
| 01-02 | 30/01/2017 | Olivier Collart | Release for TPM Firmware 0x47.0x0C (71.12) |
| 01-03 | 06/11/2019 | Olivier Collart | Release for TPM Firmware 0x47.0x10 (71.16) |
| 01-04 | 11/06/2020 | Olivier Collart | Update after review |
| 01-04p | 11/06/2020 | Olivier Collart | Public release for TPM Firmware 0x47.0x10 (71.16) |

# 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 ST Reference

This security target is referenced with the following information:

- Filename: ST33TPHF2ESPI_M20_ST
- Revision: 01.04p
- Internal documentation system reference: SSS_ST33TPHF2ESPI_M20_ST_16_001
- Date: June 11, 2020

This security target is strictly conformant to the TPM Protection Profile PC Client Specific Trusted Platform Module Family 2.0 level 0 Revision 1.16, Version 1.0, [PP-2015/07] [17].

## 1.2 Purpose

This document presents the Security Target (ST) of the ST33TPHF2ESPI product.

The reference and definition of the TOE are provided in Chapter 2.

A list of acronyms is provided in Chapter 9

# 2 TOE DESCRIPTION

## 2.1 TOE reference

**Table 1: Target of evaluation: ST33TPHF2ESPI reference**

| Product | Hardware | Hardware Version (Ext.Int) | Firmware Version Major.Minor[1] (Major.Minor in decimal format[2]) |
|---|---|---|---|
| ST33TPHF2ESPI | ST33HTPH | A.C | 0x47.0x10 (decimal 71.16) |

The chip packaging is not included in the TOE.

## 2.2 Target of evaluation Overview

The ST33TPHF2ESPI is a hybrid TPM product targeting PC, server platforms and embedded systems.

This product supports two modes exclusively:

- TPM 1.2 mode: the set of TPM 1.2 commands is supported and only TPM 1.2 assets can be accessed
- TPM 2.0 mode: the set of TPM 2.0 commands is supported and only TPM 2.0 assets can be accessed.

The mode can be selected by the platform vendor and locked irreversibly during platform provisioning. The mode can also be left modifiable by the platform firmware after platform provisioning.

The TOE is the product ST33TPHF2ESPI in mode TPM 2.0. The TPM 1.2 mode is not part of the TOE. The same product ST33TPHF2ESPI in mode TPM 1.2 is covered by another security target conformant with the TPM 1.2 protection profile.

The security target describes the target of evaluation (TOE) named ST33TPHF2ESPI mode TPM 2.0 and provides a product summary. In the following sections of this document the expressions ST33TPHF2ESPI or TPM stands for all forms factors of the TOE.

The TOE is a device that implements the functions defined in the TCG Trusted Platform Module Library Specification, version 2.0, [11], [12], [13], [14] and the PC client specific interface specification [15]. The TCG Trusted Platform Module Library specification describes the design principles, the TPM structures, the TPM commands and supporting routines for the commands. The TPM PC client specific interface specification describes the additional features that must be implemented by a TPM for a PC Client platform.

The TOE consists of

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

The TOE components are described in 2.3

[1] The firmware major and minor versions may be retrieved from the TOE with the command TPM2_GetCapability [13], in the response field TPM_PT_FIRMWARE_VERSION_1 and formatted with the value 0x00 0x47 0x00 0x10 according to [15], Table 1.

[2] Some tools may report the version in decimal value. In that case, the version retrieved is 71.16.

### 2.2.1 *TOE Usage and Security Features*

The TPM library specification describes the TPM protections in terms of Protected Capabilities and Protected Objects. A Protected Capability is an operation that must be 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 ST33TPHF2ESPI is shipped with EK and a Certificate of the Authenticity of this EK. The EK is bound to the Platform via Platform Certificate, providing assurance from the certification body of the physical binding and connection through a trusted path between the platform (the RTM) and the genuine TPM (the RTR). The attestation of the EK and the Platform Certificates build the base for attestation of other keys and measurements.

#### Root of Trust for Storage

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

**Platform Key Hierarchy**

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

**Other Security Services and Features**

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 and SHA-256 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 bits for digital signature, secret sharing and encryption and ECC algorithms with P-256 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 mode. 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 two Endorsement keys

- One 2048-bit RSA key pair
- One 256-bit ECC key pair generated with curve TPM_ECC_NIST_P256.

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 are certified by an intermediate CA 2048-bit key.

The ECC_NIST_P256 EK are certified by an intermediate CA using a NIST_P384 key.

Both certificates comply with the templates defined in the TCG specification for TPM 2.0 EK certificates [46].

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 the document TPM EK Certificate – Chip and EK authenticity verification [44].

## 2.3 TOE Description

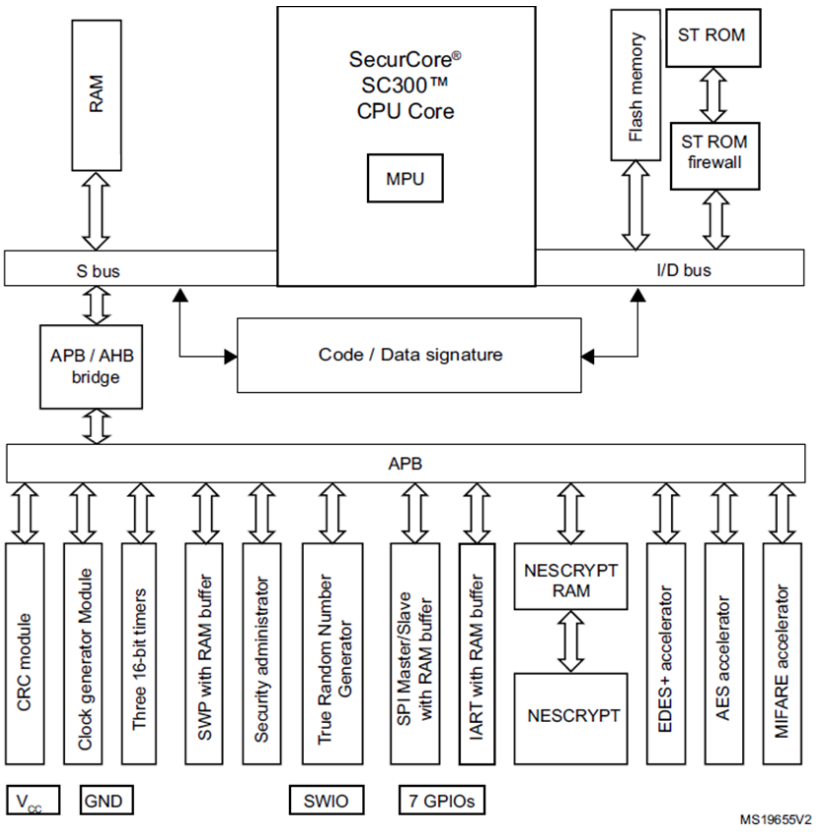
### *2.3.1 TOE hardware description*

The TOE includes the ST33HTPH hardware platform based on the ST33 product family.

The ST33HTPH is a serial access microcontroller designed for Trusted Platform Module applications that incorporates the most recent 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.

The SC300™ core brings great performance and excellent code density thanks to the Thumb®-2 instruction set.

**Figure 1: ST33HTPH block diagram**




The ST33HTPH supports a SPI interface up to 34 Mhz compliant with [9] for integration with controllers for PC Client and Server platforms and system drivers.

The ST33HTPH 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

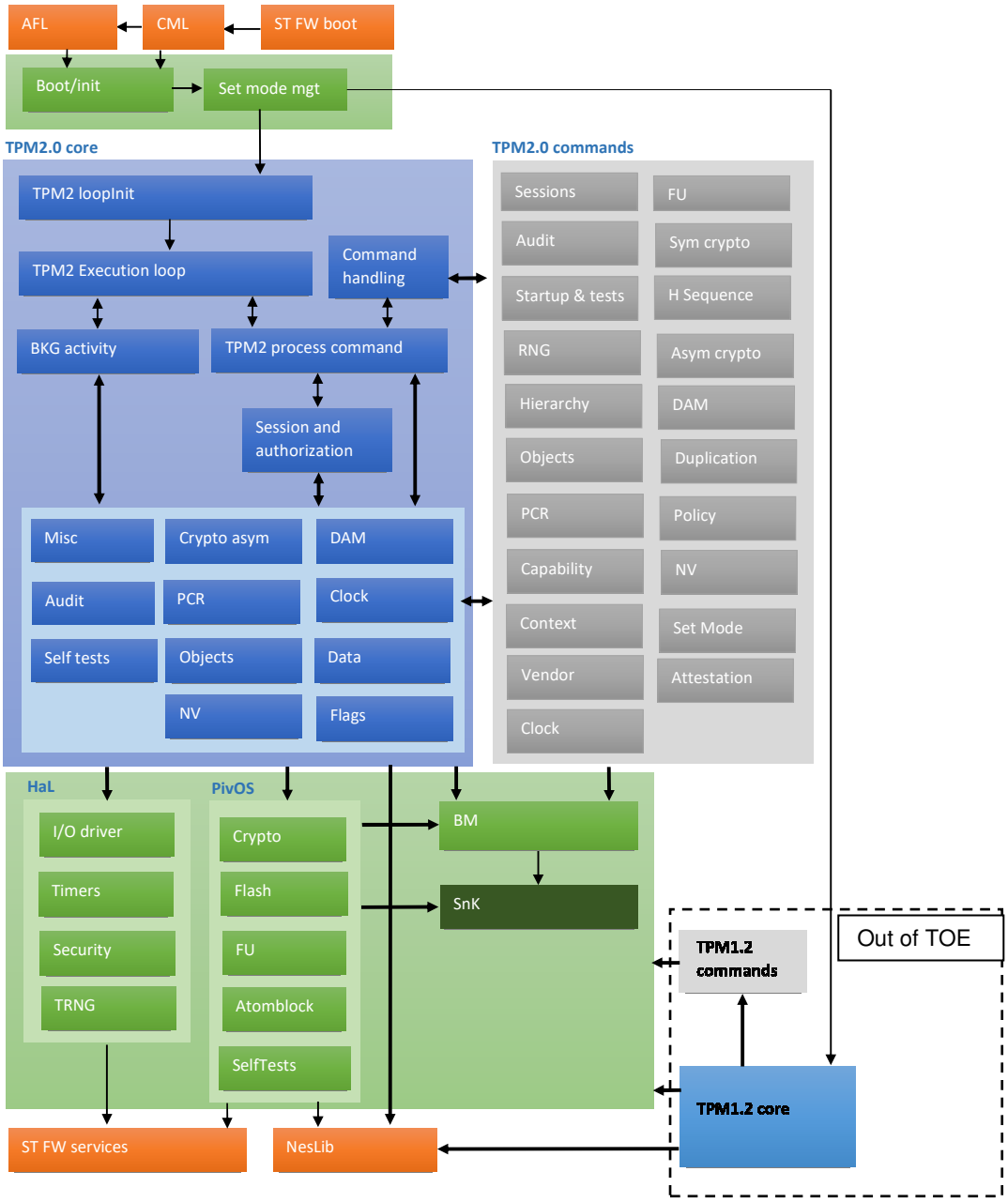
### 2.3.2 *TOE firmware description*

The TOE firmware “F2E” is divided in four compiled blocks

- TPM firmware: dynamic block supporting the TPM 2.0 commands.
- Application Flash Loader (AFL): dynamic block responsible to manage the loading of a new firmware
- Code Memory Loader (CML): static block responsible to verify the integrity of the dynamic blocks and to launch one of the dynamic blocks depending on the TPM state machine
- NesLib: static cryptographic library providing high-level crypto services to the dynamic blocks
  - NesLib 4.2.9 for ST33 is integrated into the TOE

The TPM firmware, the CML and the AFL use all the services of the ST Firmware.

**Figure 2: F2E firmware block diagram**



The TPM firmware is divided into several modules

- PivOS: module supporting a set of low-level services built with these submodules:
  - Flash, FU, AtomBlock, SelTests and Crypto
- Hardware abstraction layer (Hal) supporting a set of services provided by the hardware platform built with these submodules
  - I/O driver, Timers, Security and TRNG
- Block Manager (BM): module supporting heap services for data storage
- Secure nano kernel (Snk) supports low-level services for symmetric cryptography macrocells and for atomic transactions
  - Snk 2.20 for ST33 is integrated into the TOE
- TPM2.0 Core: module supporting
  - Command processing
  - Backgroung processing for RSA key generation and flash management
  - Command authorization
  - Data management: Crypto asym, PCR, Objects, NV, DAM, Clock, Data, Flags, Misc, Audit and SelfTests.
- TPM 2.0 Commands: module supporting TPM2_ commands implementation
  - Sessions, Audit, Startup&SelfTests, RNG, Hierarchy, Objects, PCR, Capability, Context, Vendor, Clock, FU, Sym-Crypto, H-Sequence, Asym-Crypto, DAM, Duplilcation, Policy, NV, SetMode, Attestation.
- The modules “TPM1.2 commands” and “TPM1.2 Core” are included in the product and may be activated with the command TPM2_SetMode. These modules are non-interfering with the TOE since TPM 1.2 and TPM 2.0 modes are exclusive and data accessed in these two modes are completely segregated.

### 2.3.3 *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 1.16, October 30 2014 | Revision 1.16 | October 30 2014 | [11] |
| TPM Library Part 2: Architecture, Specification Version 2.0, Revision 1.16, October 30 2014 | Revision 1.16 | October 30 2014 | [12] |
| TPM Library Part 3: Architecture, Specification Version 2.0, Revision 1.16, October 30 2014 | Revision 1.16 | October 30 2014 | [13] |
| TPM Library Part 4: Architecture, Specification Version 2.0, Revision 1.16, October 30 2014 | Revision 1.16 | October 30 2014 | [14] |
| Errata version 1.3 June 16 2015 for TCG TPM library version 2.0 revision 1.16 October 2014 | version 1.3 | June 16 2015 | [15] |
| TCG PC Client Specific Platform TPM Profile for TPM 2.0 (PTP), Family “2.0”, Level 00 Revision00.43, August 4, 2014 | Revision 00.43 | August 4, 2014 | [16] |
| TCG EK credential profile for TPM Family 2.0 Level 0. Specification Version 2.0 Revision 14, November 4 2014 | Revision 14 | November 4 2014 | [46] |
| ST33TPHF2ESPI datasheet V11, Firmware 0x47.0x0C | V11 | March 10 2017 | [48] |
| Technical note Addendum to the ST33TPHF2ESPI datasheet V11 for Firmware 0x47.0x10 | V2 | November 6, 2019 | [49] |
| TPM EK Certificate – Chip and EK authenticity verification | 2.0 | March 2016 | [44] |
| ST33TPHF20SPI - Security recommendations | 1.3 | January 2020 | [45] |

### 2.3.4 *Forms of delivery*

The TOE is delivered in form of complete chips which include the hardware, the firmware, the Endorsement Primary Keys and certificates, and the guidance documentation. The TOE is finished and the extended test features are removed. The TOE is delivered in different packages (e.g. TSSOP and VQFN). The ordering codes are listed in the document *ST33TPHF2ESPI:– Datasheet* [48] and Datasheet addendum [49].

The firmware is described in the document [49].

## 2.4 TOE 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 [17].

The phase 1 that includes TPM development involves the sites of

- ST ROUSSET (FRANCE)

- ST ANGMO KIO (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 TPM 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.

# 3 CONFORMANCE CLAIM (ASE_CCL)

## 3.1 CC Conformance Claim

This security target is **conformant** to the Common Criteria version 3.1 R4.

This security target claims to be Common Criteria version 3.1 R4

- Part 1 **conformant**,
- Part 2 **extended** and
- Part 3 **conformant**.

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

## 3.2 PP Claim

This security target is in **strict conformance** to the PC Client Specific Trusted Platform Module Family 2.0 level 0 Revision 1.16, Version 1.0, released by the Trusted Computing Group dated December 10th 2014.

The protection profile is registered and certified by the “Agence Nationale de la Sécurité des Systèmes d’Information” (ANSSI) under the reference PP-2015/07, dated May 6th 2015.

## 3.3 Package claim

This security target does not claim conformance to a package of the PP [17].

This ST is conforming to assurance package EAL4 augmented with

- ALC_FLR.1 and
- AVA_VAN.4

defined in CC Part 3.

## 3.4 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 1.16 ([11], [12] ,[13] and [14]) and the TCG PC Client Specific Platform TPM Profile Specification, Version 2.0 Revision 1.16 [17] as defined in the PP [17] section 2.2.1. So the TOE is **consistent** with the **TOE type** in the PP [17].

The **security problem** definition of this security target is **consistent** with the statement of the security problem definition in the PP [17], as the security target claims strict conformance to the PP [17] 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 [17] as the security target claims strict conformance to the PP [17]. All assignments and selections of the security functional requirements are done in the PP [17] and in this security target section 7.1.

## 3.5 Application notes

The evidences that the PP [17] 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 [17] according to —Anwendungshinweise und Interpretationen zum Schema (AIS) respectively - Functionality classes for random number generators [40].

# 4 SECURITY PROBLEM DEFINITION (ASE_SPD)

The contents of the PP [17] applies to this chapter without any restriction or addition.

## 4.1 Assets

The assets of the TOE are defined in the PP [17] section 4.1 Assets. These assets have to be protected while being executed as well as when the TOE is not in operation.

## 4.2 Threats

The threats to security are defined in the PP [17], section 4.2 Threats. No other threats are added.

## 4.3 Organisational Security Policies

The organisational security policies are defined in the PP [17], section 4.3 Organisational Security Policies, no other organisational security policies are added

### *4.3.1 Compliance to ANSSI note 6*

The organisational security policy *OSP.FieldUpgrade* defined in Application Note – Security requirements for post-delivery code loading, Version 2.0 [42] is covered by the OSP defined in TPM 2.0 PP [17] in Table 2.

## 4.4 Assumptions

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

The TOE assumptions to the IT environment are defined in the PP [17], section 4.4 Assumptions, no other assumptions are added.

# 5 SECURITY OBJECTIVES

This section shows the security objectives which are relevant for the TOE. For this section the PP [17] can be applied completely.

## 5.1 Security Objectives for the TOE

The security objectives of the TOE are defined and described in the PP [17], section 5.1 Security Objectives for the TOE, no other security objectives are added

## 5.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are described in the PP [17], section 5.2 Security Objectives for the Operational Environment, no other security objectives for the operational environment are added

### *5.2.1 Compliance to ANSSI note 6*

The security objective for the operational environment *OE.FieldUpgradeInfo* defined in Application Note – Security requirements for post-delivery code loading, Version 2.0 [42] is covered by the security objective for the operational environment *OE.FieldUpgradeInfo* defined in TPM 2.0 PP [17] in Table 5.

## 5.3 Security Objective Rationale

The security objectives rationale is described in the PP [17], section 5.3 Security Objective Rationale. No other security objectives rationale are added.

## 5.4 ANSSI note 6 Security Objectives Equivalence

**Table 3: ANSSI Note 6 Security objectives rationale**

| Objectives Note 6 | Description | Security Objective or SFR equivalence |
|---|---|---|
| O.Secure_Load_ACode | The Loader of the Initial TOE shall check an evidence of authenticity and integrity of the loaded Additional Code.<br>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. | Covered by SFR<br>FDP_ACF.1.2/States, iteration 2 from PP [17]<br>Covered by SFR<br>FDP_ACF.1.3/States iterations 1 & 2 from this security target |
| 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.<br>All the operations needed for the code to be able to operate as in the Final TOE shall be completed before activation.<br>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 of fail secure. | Covered by SFR<br>FDP_ACF.1.2/States iteration 3 |
| 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.<br>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(s) which are embedded in the Final TOE. | Covered by SFR<br>FCO_NRO.1.2/M&R iteration 6 |

# 6 EXTENDED COMPONENTS DEFINITION (ASE_ECD)

The extended component “FCS_RNG Generation of random numbers” is defined in the PP [17], section 6.1. No other extended component are added in this security target.

# 7 SECURITY REQUIREMENTS (ASE_REQ)

## 7.1 Security Functional Requirements for the TOE

### *7.1.1 Security Functional Requirements listed by the TPM 2.0 Protection Profile*

The security functional requirements (SFRs) for the TOE are defined in the PP [17] section 7.1. 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.

**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_CKM.1/PKRSA Cryptographic key generation (primary keys)**

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

FCS_CKM.1.1/PKRSA The TSF shall generate cryptographic **primary *RSA*** keys in accordance with a specified cryptographic key generation algorithm *RSA key generator* and specified cryptographic key sizes *2048 bits* that meet the following: *TPM library specification [11], [12], [13]* 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_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/PKECC The TSF shall generate cryptographic **primary *ECC*** keys in accordance with a specified cryptographic key generation algorithm *ECC key generator* and specified cryptographic key sizes *256 bits,* that meet the following: *TPM library specification [11], [12], [13]*, 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_COP.1 Cryptographic operation]
FCS_CKM.4 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 [11], [12], [13]* 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_COP.1 Cryptographic operation]
FCS_CKM.4 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 *1024 and 2048 bits* that meet the following: *TPM library specification [11], [12], [13]*, [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_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/ECC The TSF shall generate cryptographic **ECC** keys in accordance with a specified cryptographic key generation algorithm *ECC key generator* and specified cryptographic key sizes *224 and 256 bits* that meet the following: *TPM library specification [11], [12], [13]*.

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

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

FCS_CKM.1.1/SYMM The TSF shall generate cryptographic **symmetric** keys in accordance with a specified cryptographic key generation algorithm *AES key generator* and specified cryptographic key sizes *128, 192 & 256 bits* that meet the following: *TPM library specification [11], [12], [13]*.

**FCS_CKM.4 Cryptographic key destruction**

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

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *key 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]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/AES The TSF shall perform symmetric encryption and decryption in accordance with a specified cryptographic algorithm AES in the mode CFB and cryptographic key sizes 128*, 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]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SHA The TSF shall perform hash value calculation in accordance with a specified cryptographic algorithm SHA-1 and SHA-256 and cryptographic key sizes none that meet the following: FIPS 180-4 48.

**FCS_COP.1/HMAC/SHA1 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]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HMAC The TSF shall perform HMAC value generation and verification in accordance with a specified cryptographic algorithm HMAC with SHA-1 and cryptographic key sizes *160 bits* that meet the following: [FIPS 198-1] [29].

**FCS_COP.1/HMAC/SHA256 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]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/HMAC The TSF shall perform HMAC value generation and verification in accordance with a specified cryptographic algorithm HMAC with SHA-256 and cryptographic key sizes *256 bits* that meet the following: [FIPS 198-1] [29].

**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]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/RSASign The TSF shall perform signature generation and verification in accordance with a specified cryptographic algorithm RSASSA_PKCS1v1_5, RSASSA_PSS and cryptographic key sizes *1024* and 2048 bit that meet the following: PKCS#1v2.1 [RFC 3447].

**FCS_COP.1/ECDSA Cryptographic operation (ECC 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]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/ECDSA The TSF shall perform signature generation and verification in accordance with a specified cryptographic algorithm ECDSA with curve TPM_ECC_NIST_P256 and cryptographic key sizes 256 bit *and with curve TPM_ECC_NIST_P224 and cryptographic key sizes 224 bit* that meet the following: FIPS PUB 186-4 [27].

**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]
FCS_CKM.4 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* and *TPM_ECC_BN_P256* and cryptographic key sizes 256 that meet the following: TPM library specification [13].

**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]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/ECDEC The TSF shall perform decryption of ECC key in accordance with a specified cryptographic algorithm ECDH with curve *TPM_ECC_NIST_P256*, *TPM_ECC_NIST_P224* and cryptographic key sizes *224 and* 256 bit that meet the following: TPM library specification *[11], [12], [13] 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

- at the request of the authorised user "World"
  (1) the TPM2_SelfTest command and of selected algorithms using the TPM2_IncrementalSelfTest command,
- at the conditions
  (1) Initialisation state after reset and before the reception of the first command,
  (2) prior to execution of a command using a not self-tested function,
- *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 [3] :

(1) Platform firmware with the security attributes platformAuth 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[4]:

(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_HashSequenceStart, TPM2_SequenceUpdate, TPM2_EventSequence-Complete, TPM2_SequenceComplete, TPM2_PCR_Extend, TPM2_Startup, TPM2_SelfTest, TPM2_GetRandom, TPM2_HierarchyControl, TPM2_Hierarchy-ChangeAuth, TPM2_SetPrimaryPolicy, TPM2_GetCapability, TPM2_NV_Read, 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.

[3] See Table 7 in Protection Profile [17]
[4] See Table 8 and 9 in Protection Profile [17]

(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 TPM_RC_FAILURE 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).

(14) The Platform firmware is authorised to change the library mode (TPMlib mode) to TPM 1.2 mode

(15) The Platform firmware is authorised to lock permanently the TPM library mode (TPMLibLock mode)

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 strictly bigger than the minor version of the loaded firmware*

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_ACF.1/ACSecurity 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 (TPM2_ReadClock).

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

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

(6) 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.

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

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

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

(10) 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.

(11) 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.

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

(13) 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.

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

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

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

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

(18) 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.

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

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

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*.

*7.1.2* *Extended component FCS_RNG.1*

The protection profile [17] defines the extended family Random Number Generation (FCS_RNG) of the class FCS (Cryptographic support) in order to describe the generation of random numbers for cryptographic purposes.

**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 that implements: NIST SP 800-90A *Hash_DRBG*. [33]

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

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:

(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 [27].

(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 $2^{32}$ 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 140-2 statistical test suite.

## 7.2 Security assurance requirements

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

The security assurance requirements defined in Table 4 are defined in section 7.2 of the PP [17].

**Table 4: 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.1 Identification of security measures |
| | ALC_LCD.1 Developer defined life-cycle model |
| | ALC_FLR.1 Basic flow remediation - **augmented** |
| | ALC_TAT.1 Well-defined development tools |
| ASE: Security Target evaluation | ASE_CCL.1 Conformance claims |
| | ASE_ECD.1 Extended components definition |
| | ASE_INT.1 ST introduction |
| | ASE_OBJ.2 Security objectives |
| | ASE_REQ.2 Derived security requirements |
| | ASE_SPD.1 Security problem definition |
| | ASE_TSS.1 TOE summary specification |
| ATE: Tests | ATE_COV.2 Analysis of coverage |
| | ATE_DPT.1 Testing: basic design |
| | ATE_FUN.1 Functional testing |
| | ATE_IND.2 Independent testing - sample |
| AVA: Vulnerability assessment | AVA_VAN.4 Methodical vulnerability analysis - **augmented** |

## 7.3 Security Requirements rationale

The security requirements rationale of the TOE are defined and described in the PP [17], section 7.3 Security Requirements rationale.

### *7.3.1 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 [17] Table 11.

The SFRs FCS_COP.1/HMAC/SHA1 and FCS_COP.1/HMAC/SHA256 fulfil the same objectives as the SFR FCS_COP.1/HMAC defined in the PP [17] Table 11.

In addition to the rationale provided by the PP [17], FCS_COP.1/ECDSA fulfils the security objectives O.MessageNR and O.Reporting.

### *7.3.2 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 [17] Table 12.

The SFRs FCS_COP.1/HMAC/SHA1 and FCS_COP.1/HMAC/SHA256 fulfil the same dependency rationale as the SFR FCS_COP.1/HMAC defined in the PP [17] Table 12.

## 7.4 Security Assurance rationale

The security assurance requirements rationale of the TOE are defined and described in the PP [17], section 7.3.3 Assurance rationale.

# 8 TOE SUMMARY SPECIFICATION

The product overview is given in section 2.1. In the following the security functionality and the assurance measures of the TOE are described.

## 8.1 TOE Security Features

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

### *8.1.1 SF_CRY - Cryptographic Support*

There are several functions within the TOE related to cryptographic support: generation of random numbers, generation of asymmetric key pairs, RSA and ECC digital signature (generation and verification), RSA, 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 1024 and 2048 bits that meet the following: [37] and optional [35] and ECC with key sizes of 224 and 256 bits that meet *[11], [12], [13],*and optional [35].

*RSA key generator*:

- Endorsement Key generated with default template defined in [46] is securely written in the TOE during the manufacturing process
- Other keys are generated according to *[11], [12], [13] using the DRBG as random generator*

*ECC key generator*

- Endorsement Key generated with default template defined in [46] is securely written in the TOE during the manufacturing process
- Other keys are generated according to *[11], [12], [13] 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 *[11], [12], [13]* and optional [35].

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

The TOE supports the destruction of cryptographic keys by erasure of volatile memory areas containing cryptographic keys in accordance with FIPS PUB 140-2 [25].

The covered security functional requirement is FCS_CKM.4.

The TOE performs the encryption and decryption in accordance with the specified cryptographic algorithm AES in the CFB mode and cryptographic key size of 128, 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 and SHA-256 (cryptographic key sizes not available) that meets [FIPS 180-4].

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 and SHA-256 and cryptographic key sizes 160 and 256 bits that meets [FIPS 198-1] and [FIPS 180-4]

The covered security functional requirements are FCS_COP.1/HMAC/SHA1 and FCS_COP.1/HMAC/SHA256.

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 1024 bits and 2048 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 1024 bits and 2048 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 curve TPM_ECC_NIST_P256 and cryptographic key sizes 256 bits that meet TPM library specification [TPM2.0 Part1 r116] 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 curve TPM_ECC_NIST_P256 and TPM_ECC_BN_P256 and cryptographic key sizes 256 bits that meet TPM library specification [TPM2.0 Part1 r116], 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 curve TPM_ECC_NIST_P256 and cryptographic key sizes 256 bits that meet TPM library specification *[11], [12], [13] 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/PKSYMM,
- FCS_CKM.1/RSA,
- FCS_CKM.1/ECC,

- FCS_CKM.1/SYMM,
- FCS_CKM.4,
- FCS_COP.1/AES,
- FCS_COP.1/SHA,
- FCS_COP.1/HMAC/SHA1,
- FCS_COP.1/HMAC/SHA256,
- FCS_COP.1/RSAED,
- FCS_COP.1/RSASign,
- FCS_COP.1/ECDSA,
- FCS_COP.1/ECDAA,
- FCS_COP.1/ECDEC and
- FCS_RNG.1.

### *8.1.2 SF_I&A - Identification and Authentication*

The TPM provides two mechanisms for the identification and authentication capability to authorize the use of 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 detect 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 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_UID.1,
- FIA_UAU.1,
- FIA_UAU.5,
- FIA_UAU.6 and
- FIA_USB.1.

### *8.1.3 SF_G&T - General and Test*

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

The covered security functional requirement is FMT_SMR.1. The TOE performs different management functions.

The covered security functional requirement is FMT_SMF.1.

The TOE ensures that only secure values are accepted for security attributes. The covered security functional requirement is FMT_MSA.2.

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

The covered security functional requirement is FPT_STM.1

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

The covered security functional requirement is FDPT_RIP.1.

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

The covered security functional requirement is FPT_TST.1.

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

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

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

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

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

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

- 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
- 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 requirement is FPT_PHP.3

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

### *8.1.4 SF_OBH - Object Hierarchy*

The TOE supports different states during his life-cycle as described in [TPM2.0 PP] section 7.1.4.1 -TPM Operational States in detail.

The TOE enforces the TPM State Control SFP on all subjects and objects and all operations among subjects and objects covered by the SFP. The TOE ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP and enforces different access control rules on controlled subjects and objects.

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

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

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

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

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

The covered security functional requirement is FDP_SDI.1.

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

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

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

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

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

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

The TOE enforces different rules to set the value of security attributes. The covered security functional requirement is FMT_MSA.4/Hier.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The SF_OBH - Object Hierarchy covers the following security functional requirements:

- FDP_ACC.2/States,
- FDP_ACF.1/States,
- FMT_MSA.1/States,
- FMT_MSA.3/States,
- FDP_UIT.1/States,
- FDP_SDI.1,
- FDP_ACC.1/Hier,
- FDP_ACF.1/Hier,
- FMT_MSA.1/Hier,
- FMT_MSA.3/Hier,
- FMT_MSA.4/Hier,
- FDP_ACC.1/ExIm,
- FDP_ACF.1/ExIm,
- FMT_MSA.1/ExIm,
- FMT_MSA.3/ExIm,
- FDP_ETC.2/ExIm,
- FDP_ITC.2/ExIm,
- FDP_UCT.1/ExIm,

- FDP_UIT.1/ExIm,
- FDP_ACC.1/M&R,
- FDP_ACF.1/M&R,
- FMT_MSA.1/M&R,
- FMT_MSA.3/M&R and
- FCO_NRO.1/M&R

*8.1.5 SF_TOP - TOE Operation*

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

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

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

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

The TOE enforces the Access Control SFP to transmit user data in a manner protected from unauthorised disclosure. The TOE provides a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure. The TOE initiates communication via the trusted channel and permits another trusted IT product to initiate communication via the trusted channel.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

The covered security functional requirement is FCO_NRO.1/Cre

The SF_TOE - TOE Operation‖ covers the following security functional requirements:

- FDP_ACC.1/AC,
- FDP_ACF.1/AC,
- FMT_MSA.1/AC,
- FMT_MSA.3/AC,
- FDP_UCT.1/AC,
- FTP_ITC.1/AC,
- FMT_MOF.1/AC,
- FDP_ACC.1/NVM,
- FDP_ACF.1/NVM,
- FMT_MSA.1/NVM,
- FMT_MSA.3/NVM,
- FMT_MSA.4/NVM,
- FMT_MTD.1/NVM,
- FDP_ITC.1/NVM,
- FDP_ETC.1/NVM,
- FDP_ACC.1/Cre,
- FDP_ACF.1/Cre,
- FMT_MSA.1/Cre,
- FMT_MSA.3/Cre and
- FCO_NRO.1/Cre

### *8.1.6 Assignment of Security Functional 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/PKSYM | X | | | | |
| FCS_CKM.1/RSA | X | | | | |
| FCS_CKM.1/ECC | X | | | | |
| FCS_CKM.1/SYMM | X | | | | |
| FCS_CKM.4 | X | | | | |
| FCS_COP.1/AES | X | | | | |
| FCS_COP.1/SHA | X | | | | |
| FCS_COP.1/HMAC/SHA1 | X | | | | |
| FCS_COP.1/HMAC/SHA256 | 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_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_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 |

## 9 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-2012-09-001]

Common Criteria for Information Technology Security Evaluation, Version 3.1, Part 1: Introduction and general model, Revision 4, September 2012

**[2]** [CCMB-2012-09-002]

Common Criteria for Information Technology Security Evaluation, Version 3.1, Part 2: Security functional components, Revision 4, September 2012

**[3]** [CCMB-2012-09-003]

Common Criteria for Information Technology Security Evaluation, Version 3.1, Part 3: Security assurance components, Revision 4, September 2012

**[4]** [CCMB-2012-09-04]

Common Methodology for Information Technology Security Evaluation (CEM) Evaluation Methodology, Version 3.1, Rev 4, September 2012

**[5]** [TCG Glossary]

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

**[6]** [TPM Part1 116]

TPM Main, Part 1, Design principles, Version 1.2 Level 2, revision 116, 1 March 2011, Trusted Computing Group, Incorporated

**[7]** [TPM Part2 116]

TPM Main, Part 2, TPM Structures, Version 1.2 Level 2, revision 116, 1 March 2011, Trusted Computing Group, Incorporated

**[8]** [TPM Part3 116]

TPM Main, Part 3, Commands, Version 1.2 Level 2, revision 116, 1 March 2011, Trusted Computing Group, Incorporated

**[9]** [PC Client TIS 1.3]

TCG PC Client Specific TPM Interface Specification (TIS) Version 1.3 – 21 March 2013

**[10]** [TPM1.2 PP rev116]

Trusted Computing Group Protection Profile PC Client Specific TPM, Family 1.2 Level 2 Revision 116 (Version 1.3), 14 July 2014, TCG

**[11]** [TPM2.0 Part1 r116]

TPM Library Part 1: Architecture, Specification Version 2.0, Revision 1.16, October 30 2014, Trusted Computing Group, Incorporated

**[12]** [TPM2.0 Part2 r116]

TPM Library Part 2: TPM Structures, Specification Version 2.0, Revision 1.16, October 30 2014, Trusted Computing Group, Incorporated

**[13]** [TPM2.0 Part3 r116]

TPM Library Part 3: Commands, Specification Version 2.0, Revision 1.16, October 30 2014, Trusted Computing Group, Incorporated

**[14]** [TPM2.0 Part4 r116]

TPM Library Part 4: Supporting Routines, Specification Version 2.0, Revision 1.16, October 30 2014, Trusted Computing Group, Incorporated

**[15]** [TPM2.0 rev116 Err 1.3]

Errata version 1.3 June 16 2015 for TCG TPM library version 2.0 revision 1.16 October 2014.

**[16]** [PTP 0.43]

TCG PC Client Specific Platform TPM Profile for TPM 2.0 (PTP), Family “2.0”, Level 00 Revision00.43 August 4, 2014

**[17]** [TPM2.0 PP]

PC Client Specific Trusted Platform Module Family 2.0 level 0 Revision 1.16, Version 1.0 - [PP-2015/07]

**[18]** [IEEE P1363-2000]

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

**[19]** [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.

**[20]** [ISO/IEC 9797-2]

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

**[21]** [ISO/IEC 10116]

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

**[22]** [ISO/IEC 10118-3]

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

**[23]** [ISO/IEC 14888-3]

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

**[24]** [ISO/IEC 18033-3]

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

**[25]** [FIPS 140-2]

FIPS Publication 140-2

**[26]** [FIPS 180-4]

FIPS Publication, Secure Hash standard, NIST, 2002 August 1

**[27]** [FIPS 186-4]

FIPS Publication, Digital Signature Standard (DSS)

**[28]** [FIPS 197]

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

**[29]** [FIPS 198-1]

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

**[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 – Security requirements for post-delivery code loading, Version 2.0, January 23rd 2015, ANSSI.

**[43]** [DS ST33TPHF2ESPI V4]

ST33TPHF2ESPI datasheet V4, Firmware 0x47.0x00, STMicroelectronics

**[44]** [EK CERT]

TPM EK Certificate – Chip and EK authenticity verification (2.0), STMicroelectronics

**[45]** [SCY REC]

ST33TPHF20SPI - Security recommendations (1.3), STMicroelectronics

**[46]** [TPM2.0 EK CERT]

TCG EK credential profile for TPM Family 2.0 Level 0. Specification Version 2.0 Revision 14, November 4 2014, Trusted Computing Group, Incorporated

**[47]** [DS ST33TPHF2ESPI V5]

ST33TPHF2ESPI datasheet V5, Firmware 0x47.0x04, STMicroelectronics

**[48]** [DS ST33TPHF2ESPI V11]

ST33TPHF2ESPI datasheet V11, Firmware 0x47.0x0C, STMicroelectronics

**[49]** [TN ST33TPHF2ESPI V11 ADD]

Technical note Addendum to the ST33TPHF2ESPI datasheet V11 for Firmware 0x47.0x10, V2, STMicroelectronics

**Please Read Carefully:**

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

**UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.**

**UNLESS EXPRESSLY APPROVED IN WRITING BY AN AUTHORIZED ST REPRESENTATIVE, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.**

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.



STMicroelectronics group of companies Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

**www.st.com**