NUVOTON TECHNOLOGY CORPORATION

NPCT7xx TPM2.0 rev 1.59 configuration ver 1.4.4.4 Security Target Lite

Version:	**1.2**
Date:	**December 23, 2025**
Product:	**NPCT7xx TPM2.0 rev 1.59 configuration ver 1.4.4.4**
Manufacturer:	**Nuvoton Technology Corporation (NTC)**

Revision History

Version	Date	Description
1.0	July 4, 2025	Changes from ST version 1.1, configuration version 1.4.2.2: Updated configuration version to 1.4.4.4. Updated Upgradable SW version to: External: 7.2.4.1, Internal: 2.1.3.30. Updated References Section in Appendix 2.
1.1	August 28, 2025	Changed document title to *NPCT7xx TPM2.0 rev 1.59 configuration ver 1.4.4.4 Security Target Lite*.
1.2	December 23, 2025	Fixed an incorrect configuration version in Table 1.1 and in Section 2.2.

Table of Contents

1 Introduction

This section contains document management and overview information. The Security Target (**ST**) identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference an ST. The ST overview summarizes the ST and provides sufficient information for a potential user to determine whether the ST is of interest. The overview can also be used as a standalone abstract for ST catalogues and registers.

1.1 Security Target (ST) and Target of Evaluation (TOE) Identification

The Target of Evaluation (**TOE**) is a TPM (Trusted Platform Module) device, which implements the Trusted Computing Group (TCG) specifications for PC-Client TPM.

It is compound of Hardware and Firmware parts, described in Table 1.1.

In order to clarify the various usages of the words Software and Firmware in this doc, the following definitions are used:

- **Non-Upgradable Software**: The TOE software that cannot be upgraded; this includes the following components: **Booter**, **BootLoader** and **Cryplib**
- **Upgradable Software**: The TOE software that can be upgraded by applying a Field Upgrade process (see section 7.1.8); this includes the “applicative” part of the TPM firmware and contains the implementation of the TCG commands
- **TPM Firmware**: The entire TOE/TPM software, compound of both **Upgradable Software** and **Non-Upgradable Software**
- **Platform firmware**: defined in [PP], section 8.1.1; it is not part of the TOE

The code name for the TOE is **NPCT7xx TPM2.0 rev 1.59 configuration ver 1.4.4.4**.

The configuration version format (X.Y.Z.W) is as follows:

- X: changes following a change in the Hardware, Booter or Cryplib
- Y: changes following a change in the BootLoader
- Z: changes following a change in the Upgradable Software
- W: changes following a change in the guidance documentation

The title of this document is: “**NPCT7xx TPM2.0 rev 1.59 configuration ver 1.4.4.4 Security Target Lite**” and its version is 1.2.

Table 1.1 – TPM Hardware and Firmware, configuration ver 1.4.4.4

TOE Component	Description	Version	Reference
Hardware	The Hardware part of the TPM chip	VID = 1050h DID = 00FCh RID = 01h	[ERT] section 1.
Booter	The first code that runs after power up, resides in the ROM	2.0.7	Derived from HW version (can't be changed or obtained by the TOE user).
Cryplib	The cryptographic library, resides in the ROM	2.0.18	Derived from HW version (can't be changed or obtained by the TOE user).
BootLoader	The code that is measured by the Booter and loads the Upgradable Software, resides in the Non-Volatile Memory	2.0.0.32	[ERT] section 1. The BootLoader version is derived from the preloaded Upgradable Software.
Upgradable Software	The TPM library code, resides in the Non-Volatile Memory	External: 7.2.4.1 Internal: 2.1.3.30	[ERT] section 1. External: TPM_PT_FIRMWARE_VERSION_1/2 Internal: TPM_PT_VENDOR_STRING_3

The Security Target is based on [PP].

The Protection Profile and the Security Target are built with Common Criteria V3.1 Release 5.

1.2 TOE Global Overview

This security target describes the TOE, which is called “TPM2.0”, and gives a short summary specification.

The TPM2.0 is a single electronic device Trusted Platform Module (**TPM**). The TPM2.0 implements the following TCG documentation:

- TPM Main specification documents (sometimes referred as [TCG-x]): [TCG-1], [TCG-2], [TCG-3] and [TCG-4]
- [TIS]
- [PTP]

The TPM2.0 is designed to reduce system boot time and Trusted OS loading time. It provides a solution for PC security over a wide range of PC applications.

The TPM2.0 may interface with the host platform via SPI interface or I2C interface. The TPM2.0 implements the SPI and I2C interfaces as defined in [TIS] and [PTP]. The I2C interface is supported by TIS emulation over the I2C physical bus interface. The TPM2.0 is Microsoft® Windows® compliant and is supported by Linux kernel v4.0 and higher.

The following is a summary of the TPM2.0 main features:

- Single-chip TPM solution
- Three package options: TSSOP28, QFN32, UQFN16
- TCG compliance: [TCG-x], [TIS] and [PTP]
- Cryptographic operations:
    - Asymmetric (public key) cryptography: RSA digital signature generation and verification, RSA encryption and decryption, ECC digital signature generation and verification, ECC key agreement, and key derivation
    - Symmetric key cryptography: AES encryption and decryption and HMAC signatures
    - Hash generation
- Random Number Generator (RNG)
- Cryptographic hardware accelerators for AES, SHA, RSA and ECC
- Field Upgrade
- EK certification support
- Secure General-Purpose I/O (GPIO) pins
- NV storage
- Extended internal NVM lifetime

- Host Interface
  - SPI interface
  - Five localities
  - Host interface voltage level options: 1.8 Volts, 3.3 Volts
  - I2C Slave Bus Interface

1.3 Organization of the Security Target

The sections of the ST are:

- TOE Description (Chapter 2)
- Conformance Claims (Chapter 3)
- TOE Security Problem Definition (Chapter 4)
- Security Objectives (Chapter 5)
- Security Requirements (Chapter 6)
- TOE Summary Specification (Chapter 7)
- Rationale (Chapter 8)
- TPM commands (Appendix 1)
- References, Acronyms and Glossary (Appendix 2)

The TOE Description (Section 2) includes general information about the Trusted Platform Module and the TOE, assists in understanding the TOE security requirements, and provides context for the ST evaluation.

Section 3 provides Conformance Claims 3 regarding the Common Criteria and the Protection Profile used for this Security Target.

The TOE Security Problem Definition (Section 4) describes security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes:

- Assumptions regarding the TOE intended usage and environment of use
- Threats relevant to secure TOE operation
- Organisational security policies with which the TOE must comply

Section 5 contains the security objectives that reflect the stated intent of the ST. The objectives define how the TOE will counter identified threats and how it will cover identified organisational security policies and assumptions. Each security objective is categorised as being for either the TOE or the TOE environment.

Section 6 contains the applicable security requirements taken from the Common Criteria, with appropriate refinements. The IT security requirements are subdivided as follows:

- TOE Security Functional Requirements
- TOE Security Assurance Requirements

The TOE Summary Specification (Section 7) summarises the security features of this specific TOE, the TPM2.0.

The Rationale (Section 8) demonstrates that the ST is a complete and cohesive set of requirements and that the TOE provides an effective set of IT security countermeasures within the security environment. The Rationale has three main parts. First, a Security Objectives Rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are capable of covering them. Then, a Security Requirements Rationale demonstrates that the security requirements (TOE and environment) are traceable to the security objectives and are capable of dealing with them. Finally, the TOE summary specification rationale consists of a TOE security functions rationale and an assurance measures rationale.

Section 9 identifies the TPM commands provided by the TOE.

Section 10 includes a glossary of terms and acronyms used in the ST and also provides references.

1.4 Common Criteria Conformance

This ST was built according to Common Criteria (**CC**) Version 3.1 Revision 5 (ISO/IEC 15408 Evaluation Criteria for Information Technology Security; Part 1: Introduction and general model, Part 2: Security functional requirements, and Part 3: Security assurance requirements).

The Security Target is conformant with [PP]. This means that the Security Target is conformant with Common Criteria Version 3.1 Revision 5, part 2 “extended” and part 3 [CC].

The assurance level for the TOE is **EAL 4 augmented** with ALC_FLR.1, AVA_VAN.4 and ALC_DVS.2.

2 TOE Description

The TOE description helps to understand the specific security environment and the security policy. In this context, the assets, threats, security objectives and security functional requirements can be employed. After some general remarks about the Trusted Platform Module in Sections 2.1, Section 2.2 presents a more detailed description of the TOE than in the [PP] since it refers to this particular TOE implementation.

2.1 TPM - General Remarks

The Trusted Platform Module is an integrated circuit and software platform that provides computer manufacturers with the core components of a subsystem used to assure authenticity, integrity and confidentiality in e-commerce and Internet communications within a Trusted Computing Platform, as defined in [PTP]. The TPM is a complete solution, implementing the Trusted Computing Group specification [TCG-x], which is an industry group originally founded in 1999 by COMPAQ, HP, IBM, Intel, Microsoft as “TCPA”, and later changed to the current TCG organization.

A Trusted Platform is a platform that can be trusted by local users and by remote entities. The basis for trusting a platform is a declaration by a known authority that a platform with a given identity can be trusted to measure and report the way it is operating. This operating information can be associated with data stored on the platform, to prevent the release of that data if the platform is not operating as expected. Other authorities provide declarations that describe the operating information the platform ought to produce when it is operating properly. The local user and remote entities trust the judgment of the authorities; so, when they receive proof of the identity of the platform, information about the current platform environment, and proof about the expected platform environment, they can decide whether to trust the platform to behave in a sufficiently trustworthy and predictable manner. The local user and/or remote entities must take this decision themselves because the level of trust in a platform can vary with the intended use of that platform, and only the local user and/or remote entities know that intended purpose.

The trusted mechanism of the platform uses cryptographic processes, including secrets. The trusted mechanisms are required to be isolated from the platform to protect secrets from disclosure and protect methods from subversion.

The subsystem protects itself against physical and software attacks to provide protection against attacks to the platform.

Some, but not all, subsystem capabilities must be trustworthy for the subsystem to be trustworthy. These are called the “Trusted Set” (**TS**). Other capabilities must work properly if the subsystem is to work properly, but they do not affect the level of trust in a subsystem. These are called the “Trusted platform Support Set” (**TSS**).

The Trusted Set of capabilities can be partitioned into measurement capabilities, reporting capabilities, and storage capabilities. The trusted measurement capabilities are called the "Root of Trust for **Measurement**" (**RTM**). The trusted reporting capabilities are called the "Root of Trust for **Reporting**" (**RTR**). The trusted storage capabilities are called the "Root of Trust for **Storage**" (**RTS**).

- The RTM makes reliable measurements about the platform and puts the measurement results into the RTR.
- The RTR prevents unauthorized changes to the measurement results, and reliably reports those measurement results.
- The RTS provides methods to minimize the amount of trusted storage that is required.

The RTM and the RTR cooperate to permit an entity to receive the measurements that describe the current computing environment in the platform. An entity can assess those measurement results and compare them with values that are to be expected if the platform is operating as expected. If there is a sufficient match between the measurement results and the expected values, the entity can trust computations within the platform (not just within the TS) to execute as expected.

The RTR has a cryptographic identity in order to prove to a remote entity that RTR messages come from genuine trusted capabilities and not from bogus trusted capabilities.

The TCG subsystem is a trusted subsystem that is an integral part of a computing platform. The evaluated components that make up the TCG subsystem are called the Trusted Building Blocks (**TBB**). The TBB provide useful trust and security capabilities, while minimizing the number of functions that must be trusted. The TBB consist of logical components, including the Trusted Platform Module (**TPM**), the Connection module (**PCCON**) and the Trusted Platform Support Services (**TSS**). In general, the TPM contains all trusted capabilities except for the RTM, so a TPM is common to all types of trusted platforms. The TPM uses cryptographic techniques to reliably report its identity and the measurement results. Since this raises privacy issues, the Subsystem includes features that provide privacy controls to the Owner. The PCCON provides the connection to the computing platform and the RTM. The TSS is a set of functions and data that are common to all types of platforms, which are not required to be trustworthy.

The TPM is a collection of hardware and software that support a variety of security feature that include, but are not limited to, the following:

- Algorithms: ECC, RSA, SHA-1, SHA-256, SHA-384, HMAC, AES
- Random number generation
- Key generation
- Self tests
- Physical protection

The TPM may be used to provide secure storage for an unlimited number of private keys or other data by using RSA key technology to encrypt data and keys. The resulting encrypted

file, which contains header information in addition to the data or key, is called a "**blob**". A blob is output by the TPM and can be loaded in the TPM when needed. The functionality of the TPM can also be used so that 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 functionality used to provide secure storage is specified in [TCG-1] Clause 22.

Various key types are defined within the TPM. Key types include:

- Storage Root Key (**SRK**) – the root key of a hierarchy of keys associated with a TPM. It is generated by the TPM from the Storage Primary Seed (**SPS**) at the request of the Owner. Each seed value has a different life cycle, but the way it seeds the associated hierarchies is approximately the same. This allows multiple storage hierarchies with differing security properties, as needed by various applications, without requiring that all of the SRKs occupy persistent TPM memory. An SRK may be made persistent in TPM Non-Volatile (**NV**) memory if required by the application.
- Signing key – must be a leaf of the Storage Root Key hierarchy. The private key of the key pair is used for signing operations only.
- Storage key – whose seed value is used to generate symmetric keys for protection (integrity and confidentiality) of other objects (its child keys) in the Protected Storage hierarchy.
- The Endorsement Key (**EK**) pair – an asymmetric key pair inserted in a TPM. It is used to prove that a TPM is a genuine TPM. Nuvoton TPM firmware has a pre-installed preparation for Endorsement Keys (EKs) and their certificates. A detailed description of the terms can be found in the TCG specification, Part 1.

  The TOE contains a pre-installed seed for creation of endorsement keys and certificates for one RSA (either 2048-bit or 3072-bit) EK and one 384-bit ECC EK derived from this seed. The certificates are stored in dedicated NV storage, and when creation of a specific EK is executed, NV indices for the EK and its certificate are created.

TPM algorithms, protocols, identification and authentication, and access control functions are described in the subsections below.

2.1.1 Algorithms

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 HMAC, and key generation. TOE hash functions SHA-1, SHA-256 and SHA-384 provide cryptographic services to external entities for measurements and are used internally for user authentication, signing and key derivation. The TOE is required to implement asymmetric algorithms, where the current specification supports RSA with up to 4096 bits for digital signature, secret sharing and encryption and ECC algorithms with P-256 and P-384 curves for digital signatures, secret

sharing and key exchange. The TOE provides symmetric encryption and decryption of AES-128 and AES-256 in CFB, CTR and OFB modes of operation.

2.1.2 Random Number Generator (RNG)

The RNG capability is only accessible to valid TPM commands. Intermediate results from the RNG are not available to any user. When the data is for internal use by the TPM (e.g., asymmetric key generation), the data is held in a shielded location and is not accessible to any user.

2.1.3 Key Generation

The TPM generates asymmetric key pairs. The generate function is a protected capability and the private key is held in a shielded location.

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.

2.1.4 Self Tests

The TPM provides start-up self tests and a mechanism to allow the self tests to be run on demand. The response from the self tests is either pass or fail. Self tests include checks of the following:

- RNG functionality, as defined by [FIPS140-2] and [SP800-90A].
- Integrity of the protected capabilities of the TPM. This consists of checks that ensure that the TPM firmware has not changed.
- Cryptographic services – the SHA-1, SHA-256, SHA-384, HMAC, AES, RSA and ECC modules are checked by performing the corresponding action on a known value and comparing the result to the known/expected result.

On failure of any of the above-specified tests the TPM enters Failure Mode.

2.1.5 Identification and Authentication

The TPM identification and authentication capability is used to authenticate an entity owner and to authorize use of an entity. The basic premise is to prove knowledge of a shared secret. This shared secret is the identification and authentication data. The TCG Specification uses the term “authorization” for the identification and authentication process, and the data related to identification and authentication is called authorization data.

The identification and authentication data for the TPM Owner and the owner of the Storage Root Keys are held within the TPM itself. The identification and authentication data for other owners of entities are held and protected with the entity.

2.1.6 Access Control

Access control is enforced in the TPM on all data and operations performed on that data. The TPM provides access control by denying access to some data and operations and allowing

access to other data and operations based on the authorization and policy-related attributes of the data.

Access control is detailed in [TCG-1] Clause 37 NV Memory.

2.2 TOE Overview

The Target of Evaluation (**TOE**), NPCT7xx TPM2.0 rev 1.59 configuration ver 1.4.4.4, is a Trusted Platform Module, which provides TCG-compliant security functionality.

The TPM2.0 is a single electronic device, comprising a Trusted Platform Module (TPM) for PC security, based on the TCG standard.

The TPM2.0 device includes an embedded RISC core for hidden execution of security code, flash memory-based secured information storage, a non-deterministic Random Number Generator, and performance accelerators that support cryptographic algorithms SHA-1, SHA-256, SHA-384, RSA, ECC and AES. In addition, the TPM2.0 integrates a variety of system functions, enabling efficient implementation of a highly secure, trustworthy system.

The TPM2.0 device complies with TCG specification ([TCG-x], [TIS] and [PTP]) and is developed by Nuvoton Technology Corporation.

The TPM2.0 device provides target platforms with:

- System integrity checks: Enables checking of the TOE integrity.
- Authentication: Provides assurance that the source of the data is valid and as expected.
- Data integrity checks: Provides assurance that received data is exactly as sent.
- Secure storage: Supplies shielded location and protected storage mechanism to protect sensitive and confidential data.

The TOE TPM module includes the TPM hardware and the TPM firmware. The host software that is needed to build a TCG system is not a part of the TOE. The hardware part of the TOE (see Figure 1), representing the physical scope of the TOE, is comprised of the following modules:

- Processing Unit Module
- Public Key (RSA, ECC) Accelerator Module
- SHA-1, SHA-256 and SHA-384 Accelerator
- AES Accelerator
- RNG (Random Number Generator) Module
- Clock
- GPIO Ports Module (General-Purpose Input/Output).
- Host Interface
    - SPI interface with up to 64-byte burst and maximum frequency of 54 MHz
    - Five localities

- Host interface voltage level options: 1.8 Volts, 3.3 Volts
- I2C Slave Bus Interface

The TOE TPM module works with the PCCON, which may include the PC system BIOS and other software. The PCCON is not part of this evaluation.

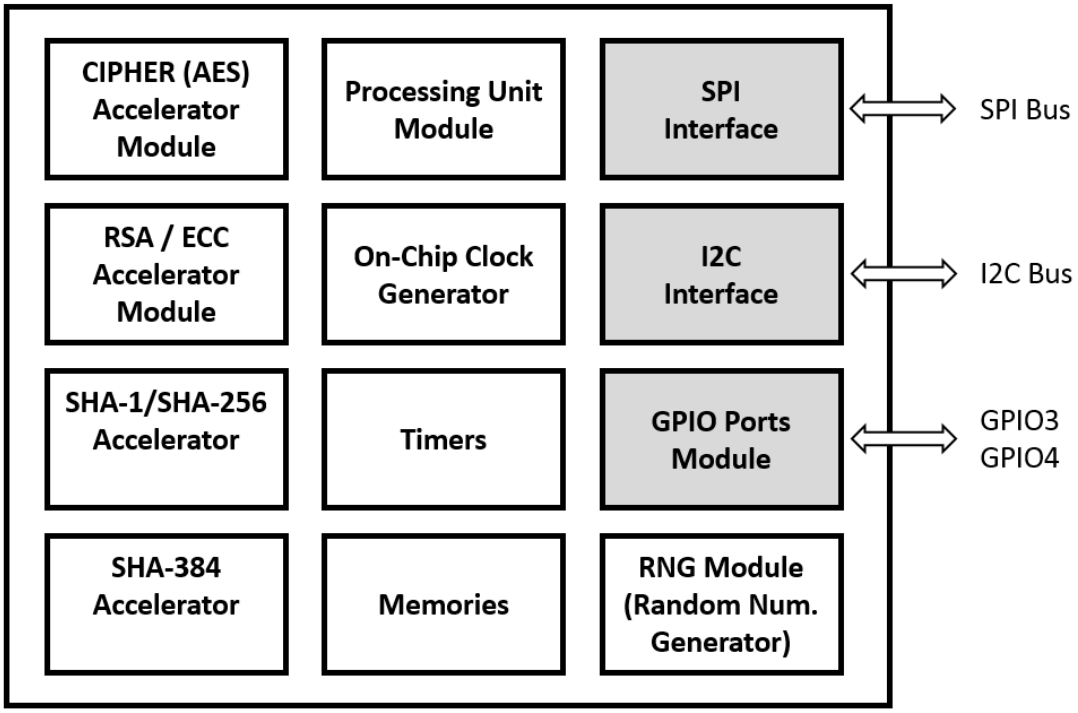

Figure 1 – TPM2.0 Block Diagram

The **TPM firmware** provides an API set that matches the TCG specification [TCG-x]. The API represents the logical scope of the TOE. TCG capabilities that must be trustworthy can be accessed only through the authentication mechanism or by supplying physical presence proof.

In addition to the TCG mandatory functions, the TPM firmware implements NTC proprietary commands and additional non-TPM related functionality.

The TPM2.0 can be used in a wide field of applications, e.g., in a remote access network to authenticate platforms to a server and vice versa. Concerning e-commerce transactions, contracts can be signed with digital signatures using the TPM2.0 asymmetric encryption functionality. Regarding a network scenario, the client PCs equipped with a TPM2.0 are able to report their platform status to the server so that the network administration is aware of their trustworthiness. In conclusion, the TPM2.0, acting as a service provider to a system, helps to make transactions more secure and trustworthy.

Hardware interface: The physical interface and the electrical interface of the TOE are the pins of the device. The electrical interface of the TOE to the external environment is the active pins of the device. Some of the pins are configurable; the life cycle of the TOE details the phases when configuration is possible. The device has 28/32 pins, which include power and ground, SMBus (I2C), SPI interface, a Physical Presence pin and general purpose I/Os. TPM commands and response may be transferred between the TPM and the host via SPI or I2C bus.

Software interface: The interface to the TPM firmware goes through the communication buffer. The host sends an input message block (command for execution) to the TOE. The TOE processes the message block, executes the command and sends a reply (status and return values).

In the communication process, there are two sides involved: the device side (the TPM) and the host side. The host side typically refers to any process in the host computer that communicates with the TPM (e.g., the BIOS or the OS resident drivers).

Guidance documentation: The guidance documentation consists of:

- The device datasheet [Datasheet], which details the specific vendor software commands and the drivers protocols.
- The TOE's programmer's guide [PRG] and security guidance [AGD] documents used during this evaluation, which detail all aspects of the TOE that are relevant for the user and administrator.
- The User Product Information [ERT] which details the TOE identification and known issues.
- The TCG main specification [TCG-x], which details all the standard TCG commands and the protocols for device initialization, starting from endorsement key-pair generation.

The guidance documents [Datasheet], [PRG], [AGD] and [ERT] are delivered to the customer by NTC in PDF format via email, whereas the TCG main specification [TCG-x] is available publicly for download from the TCG website.

TOE life cycle description: The life cycle of the TPM2.0 TOE includes several processes and conforms to the four phases specified in [PP]:

- Development of the TPM (Phase 1)
- Manufacturing and Delivery of the TPM (Phase 2)
- Platform Integration (Phase 3)
- Operational Usage (Phase 4)

Sites of Development Environment, Manufacturing and Delivery:

Design Center	
Design Center 1: Nuvoton Technology Israel Ltd.	Israel
Design Center 2: Nuvoton Technology Israel Ltd	Israel
Mask Fab	
TSMC Fab 14A	Taiwan, R.O.C.
Wafer Fab	
TSMC Fab 14A (mask and wafer manufacturing)	Taiwan, R.O.C.
TSMC Fab 8 (data center)	Taiwan, R.O.C.
TSMC Fab 3 (eFlash IP merge)	Taiwan, R.O.C.
TSMC Fab 2 and 5 (mask data preparation)	Taiwan, R.O.C.
TSMC Fab 18 (finished goods warehouse)	Taiwan, R.O.C.
Assembly Plants	
ASE Group Chung-Li	Taiwan, R.O.C
UTL (UTAC Thailand 1 / QFN)	Thailand
UTL (UTAC Thailand 2 / TSSOP)	Thailand
Wafer Test and Final Test Plants	
Nuvoton Technology Corporation	Taiwan, R.O.C.
ASE Group Chung-Li	Taiwan, R.O.C

Information on TOE Delivery to Customer:

TOE Part	Sent from Nuvoton Technology Corp.?	Deliverable Format	Delivery Method
TPM chip	Yes	Packaged IC (final product) – tested and locked chip	Courier
Guidance documents: [Datasheet], [PRG], [AGD] and [ERT]	Yes	PDF document	email
TCG main specification [TCG-x]	No. Publicly available for download from the TCG website	PDF or DOC document	Download from web
Field Upgrade package	Yes	Zip file with the encrypted field upgrade payload	email

3 Conformance Claims

3.1 CC Conformance Claim

This Security Target is conformant with Common Criteria version 3.1 Release 5, Part 2 extended.

This Security Target is conformant with Common Criteria version 3.1 Release 5, Part 3.

3.2 PP Claim

This Security Target is in strict conformance with [PP].

The Protection Profile is certified by the ANSSI under the reference ANSSI-CC-PP-2021/02 on November 30, 2021.

3.3 Package Claim

This Security Target is conformant with the assurance package defined in [PP]: EAL4 augmented with ALC_FLR.1, AVA_VAN.4 and ALC_DVS.2.

3.4 Conformance Claim Rationale

This Security Target claims strict conformance to only one PP ([PP]).

The TOE is a complete solution implementing the TCG Trusted Platform Module specification version 2.0, as defined in the PP ([TCG-x]), so the TOE is consistent with the TOE type defined in the claimed PP.

The security **problem definition** is consistent with the statement of the security problem definition of the PP.

The security **objectives** are consistent with the statement of the security objectives of the PP. Three security objectives, related to [JIL_SCRL], were added to the Security Target. These security objectives do not interfere with PP conformance.

The security **requirements** are consistent with the statement of the security requirements of the PP. All assignments and selections of the PP SFRs are reproduced in this Security Target.

4 TOE Security Problem Definition

The content of the PP ([PP], chapter 4) applies to this chapter. It is reproduced here to assist the reader's understanding. This document contains three additional threats, in comparison with the [PP], for compliance with the [JIL_SCRL].

4.1 Assets

This section of the security problem definition describes the assets of the TOE to be protected from threats.

Note that the assets are those of the PP only (see [PP] section 5, reference of tables 8 and 9).

4.2 Threats to Security

Threats to the TOE are defined in Table 4.1.

Table 4.1 – Threats

#	Threat	Description
1	T.Compromise	An undetected compromise of the data in shielded locations may occur as a result of an attacker (insider or outsider) attempting to perform actions that the individual or capability is not authorized to perform.
2	T.Bypass	An unauthorized individual or user may tamper with TSF, security attributes or other data to bypass TOE security functions and gain unauthorized access to TOE assets.
3	T.Export	A user or an attacker may export data from shielded locations without security attributes or with insecure security attributes, causing the data exported to be erroneous and unusable, to allow erroneous data to be added or substituted for the original data, and/or to reveal secrets.
4	T.Hack_Crypto	Cryptographic key generation or operation may be implemented incorrectly, allowing an unauthorized individual or user to compromise keys generated within the TPM or encrypted data, or to modify data undetected.
5	T.Hack_Physical	An unauthorized individual or user of the TOE may cause unauthorized disclosure or modification of TOE assets by physically interacting with the TOE. The attacker may be a hostile user of the TOE.
6	T.Imperson	An unauthorized individual may impersonate an authorized user of the TOE (e.g., by dictionary attacks to guess the authorization data) and thereby gain access to TOE data in shielded locations and protected capabilities.
7	T.Import	A user or attacker may import data without security attributes or with erroneous security attributes, causing key ownership and authorization to be uncertain or erroneous thus causing the system to malfunction or operate in an insecure manner.

#	Threat	Description
8	T.Insecure_State	The TOE may start up in an insecure state or enter an insecure state, allowing an attacker to obtain sensitive data or compromise the system.
9	T.Intercept	An attacker may intercept the communication between a user and the TPM subjects to gain knowledge of the commands and data sent to the subject or manipulate the communication.
10	T.Malfunction	TOE assets may be modified or disclosed to an unauthorized individual or user of the TOE, through malfunction of the TOE.
11	T.Modify	An attacker may modify data in shielded locations or their security attributes to gain access to the TOE and its assets.
12	T.Object_Attr_Change	A user or attacker may create an object with no security attributes or make unauthorized changes to security attribute values for an object, to enable attacks.
13	T.Replay	An unauthorized individual may gain access to the system and sensitive data through a “replay” or “man-in-the-middle” attack that allows the individual to capture identification and authentication data.
14	T.Repudiate_Transact	An originator of data may deny originating the data to avoid accountability.
15	T.Residual_Info	A user may obtain information that the user is not authorized to have when the data in shielded locations is no longer actively managed by the TOE (“data scavenging”).
16	T.Leak	An attacker may exploit information that is leaked from the TOE during usage of the TSF to disclose confidential assets.
17	T.Unauthorized_Load	An attacker tries to load an additional code that is not intended to be assembled with the initial TOE, i.e., the evidence of authenticity or integrity is not correct.
18	T.Bad_Activation	An attacker tries to perturbate the additional code activation so that the final TOE is different than the expected one (initial TOE or perturbated TOE).
19	T.TOE_Identification_ Forgery	An attacker tries to perturbate the TOE identification and in particular the additional code identification.

4.3 Organizational Security Policies

OSPs are defined in Table 4.2.

Table 4.2 – Organizational Security Policies

#	OSP	Description
1	OSP.Context_Management	A resource manager will be able to secure caching of resources without knowledge or assistance from the application that loaded the resource.
2	OSP.Policy_Authorisation	The TPM supports multiple trusted processes obeying the principle of least privilege by means of role-based administration and separation of duty by configuring policy authorization to allow individual entities (trusted processes, specific privileges, operations).
3	OSP.Locality	The TCG platform supports multiple transitive trust chains by means of a mechanism known as “locality”. The Host Platform's trusted processes assert their locality to the TPM. The TPM guards access to resources, PCRs and NV Storage Space, to keys and data to be imported, and to defined commands, depending on the execution environment’s privilege level.
4	OSP.RT_Measurement	The Root of Trust for Measurement calculates and stores the measurement digests as hash values of a representation of embedded data or program code (measured values) for reporting.
5	OSP.RT_Reporting	The Root of Trust for Reporting reports on the contents of the RTS. An RTR report is typically a digitally signed digest of the contents of selected values within a TPM (measurement, key properties or audit digest). The authenticity of the assets reported is based on the verification of the signature and the certificate of the signing key.
6	OSP.RT_Storage	The Root of Trust for Storage protects the assets entrusted to the TPM in confidentiality and integrity.
7	OSP.FieldUpgrade	The Platform software is allowed to perform Field Upgrade within the certified TPM or installing a new certified TPM before and after delivery to the end user. The end user shall be aware of the certification and the version of the TPM.

4.4 Secure Usage Assumptions

TOE secure usage assumptions are defined in Table 4.3.

Table 4.3 – Assumptions about the IT Environment

#	Assumption	Description
1	A.Configuration	The TOE will be properly installed and configured based on the AGD instructions.

5 Security Objectives

The content of the PP ([PP], chapter 5) applies to this chapter completely. It is reproduced here to assist the reader's understanding. The O.Secure_Load_ACode, O.Secure_AC_Activation and O.TOE_Identification objectives are expanded in this Security Target to support the ANSSI [JIL_SCRL] requirements.

5.1 Security Objectives for the TOE

TOE security objectives are defined in Table 5.1.

Table 5.1 – Security Objectives for the TOE

#	Objective	Description
1	O.Context_Management	The TOE must ensure a secure wrapping of a resource (except seeds) in a manner that securely protects the confidentiality and the integrity of the data of this resource and allows the restoring of the resource on the same TPM and during the same operational cycle only. (A TPM operational cycle is a Startup Clear to a Shutdown Clear, and contexts cannot be reloaded across a different Startup Clear to Shutdown Clear cycle from the one in which they are created.
2	O.Crypto_Key_Man	The TOE must manage cryptographic keys, including generation and derivation of cryptographic keys using the TOE Random Number Generator as source of randomness, in a manner to protect their confidentiality and integrity.
3	O.DAC	The TOE must control and restrict user access to the TOE-protected capabilities and shielded locations in accordance with a specified access control policy, where the object owner manages the access rights for their data objects using the principle of least privilege.
4	O.Export	When data is exported outside the TPM, the TOE must securely protect the confidentiality and the integrity of the data, as defined for the protected capability. The TOE shall ensure that the data security attributes being exported are unambiguously associated with the data.
5	O.Fail_Secure	The TOE must enter a secure failure mode in the event of a failure.
6	O.General_Integ_Checks	The TOE must provide checks on system integrity and user data integrity.
7	O.I&A	The TOE must identify all users and will authenticate the claimed identity except the role, "World", before granting a user access to the TOE facilities.

#	Objective	Description
8	O.Import	When data is being imported into the TOE, the TOE must ensure that the data security attributes are imported with the data and that the data is from an authorized source. In addition, the TOE will verify those security attributes according to the TSF access control rules. The TOE supports the protection of confidentiality and the verification of the integrity of imported data.
9	O.Limit_Actions_Auth	The TOE must restrict the actions a user may perform before the TOE verifies the identity of the user.
10	O.Locality	The TOE must control access to objects based on the locality of the process communicating with the TPM.
11	O.Record_Measurement	The TOE must support calculating hash values and recording the result of a measurement.
12	O.MessageNR	The TOE must provide user data integrity, source authentication, and the basis for source non-repudiation when exchanging data with a remote system.
13	O.No_Residual_Info	The TOE must ensure there is no “object reuse”, i.e., there is no residual information in information containers or system resources upon their reallocation to different users.
14	O.Reporting	The TOE must report measurement digests and must attest to the authenticity of measurement digests.
15	O.Security_Attr_Mgt	The TOE must allow only authorized users to initialize and to change security attributes of objects and subjects. The management of security attributes will support the principle of least privilege by means of role-based administration and separation of duty.
16	O.Security_Roles	The TOE must maintain security-relevant roles and association of users with those roles.
17	O.Self_Test	The TOE must provide the ability to test itself, verify that the integrity of the shielded data objects and the protected capabilities operate as designed, and enter a secure state in the case of detected errors.
18	O.Single_Auth	The TOE must provide a single-user authentication mechanism and require re-authentication to prevent “replay” and “man-in-the-middle” attacks.
19	O.Sessions	The TOE must provide the confidentiality of the parameters of the commands within an authorized session and the integrity of the audit log of the commands.
20	O.Tamper_Resistance	The TOE must resist physical tampering of the TSF by hostile users. The TOE must protect assets against leakage.
21	O.FieldUpgradeControl	The TOE restricts the Field Upgrade to authorized role and accepts only authentic update data provided by the TOE vendor.

#	Objective	Description
22	O.Secure_Load_ACode	The loader of the initial TOE will check an evidence of authenticity and integrity of the loader Additional Code. The loader enforces that only the allowed version of the Additional Code can be loaded on the Initial TOE. The loader will forbid the loading of an Additional Code not intended to be assembled with the Initial TOE. During the Load Phase of an Additional Code, the TOE will remain secure.
23	O.Secure_AC_Activation	Activation of the Additional Code and update of the identification data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the final TOE will be completed before activation. If the Atomic Activation is successful, then the resulting product is the final TOE; otherwise (in case of interruption or an incident that prevents the forming of the final TOE), the initial TOE will remain in its initial state or fail secure.
24	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 this data. After atomic activation of the additional code, the identification data of the final TOE allows identification of the initial TOE and additional code. The user must be able to uniquely identify initial TOE and additional code, which are embedded in the final TOE.

5.2 Security Objectives for the Operational Environment

Table 5.2 lists security objectives for the operational environment.

Table 5.2 – Security Objectives for the Environment

#	Objective Name	Objective Description
1	OE.Configuration	The TOE must be installed and configured properly for starting up the TOE in a secure state. The security attributes of subjects and objects will be managed securely by the authorized user.
2	OE.Locality	The developer of the host platform must ensure that trusted processes indicate their correct locality to the TPM and that untrusted processes are only able to assert locality 0 to the TPM.
3	OE.Credential	The IT environment must create EK and AK credentials by trustworthy procedures for the Root of Trust for Reporting.
4	OE.Measurement	The platform part of the Root of Trust for Measurement provides a representation of embedded data or program code (measured values) to the TPM for measurement.
5	OE.FieldUpgradeInfo	The developer, via AGD documentation, will instruct the admin how to do the upgrade and also instruct the admin that it should inform the end user regarding the Field Upgrade process, its result, whether the installed firmware is certified or not, and the version of the certified TPM.

6 Security Requirements

This section defines the TOE security functional requirements and assurance requirements. All Security Functional Requirements (except FCS_RNG.1) are from the CC Part 2. “FCS_RNG.1” is the only extended component; it is fully described in [PP] §6 (and not reproduced here).

Selections, assignments, iterations and refinements performed in the [PP] are indicated by *italics.* Operations not performed in the [PP] (selections, assignments) and additional refinements and iterations that are performed within this ST are indicated by ***bold*** *italics*.

All iterations from the PP are kept in the following text. The many application notes from the PP are not reproduced here.

The Subjects, Roles, Objects, Operations, and Security Attributes used in the Security Functional Requirements are all defined in [PP] §7.1.1 and §7.1.4.1 (and not repeated here).

All Assurance Requirements are from the CC Part 3.

6.1 Security Functional Requirements for the TOE

This section states the TOE security functional requirements. The full text of the security functional requirements is contained below (the Application Notes from the PP have not been reproduced).

6.1.1 General SFR

Security Management

FMT_SMR.1 Security roles

Hierarchical to:	No other components
Dependencies:	FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles:

1) *Platform firmware,*
2) *Platform owner,*
3) *Privacy Administrator,*
4) *Lockout Administrator,*
5) *USER,*
6) *ADMIN,*
7) *DUP,*
8) *World.*

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components
Dependencies: No dependencies

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

(1) Management of hierarchies,
(2) Management of authorization values,
(3) Management of security attributes of keys,
(4) Management of security attributes of PCR,
(5) Management of security attributes of NV storage areas,
(6) Management of security attributes of monotonic counters,
(7) Reset the management of TPM dictionary attack mitigation mechanism

FMT_MSA.2 Secure security attributes

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

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for ***security attributes of keys, PCR, NV storage areas and monotonic counters and NTC_FieldUpgrade command security attributes related.***

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components
Dependencies: No dependencies

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps *as number of milliseconds the TOE has been powered since initialization of the Clock value.*

6.1.2 Data Protection and Privacy

FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components
Dependencies: No dependencies

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from* the following objects:

- *SPS,*
- *Primary Keys,*
- *User keys,*
- *Context,*
- *PCR data,*
- *NV storage data where (TPMA_NV_PLATFORMCREATE == CLEAR),*
- *Credentials.*

6.1.3 Cryptographic Support

FCS_RNG.1 Random number generation

Hierarchical to: No other components
Dependencies: No dependencies

FCS_RNG.1.1 The TSF shall provide a ***hybrid*** Random Number Generator that implements: ***an entropy source based on a hardware RNG (designed and tested as defined by [SP800-90B]). The hardware RNG output bits are used as input to a DRBG algorithm based on CTR_DRBG with AES-256 (designed as defined by [SP800-90A] and [SP800-90C]).***

FCS_RNG.1.2 The TSF shall provide random numbers that meet: ***Statistical tests as defined by [SP800-90B]. Entropy Estimation Suite cannot practically distinguish the random numbers from output sequences of an ideal RNG.***

FCS_CKM.1/PK 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/PK The TSF shall generate cryptographic *primary* ***RSA and ECC*** keys in accordance with a specified cryptographic key generation algorithm, ***based on FIPS-approved DRBG algorithm ([SP800-90A]),*** and specified cryptographic key sizes ***RSA 2048, 3072 and 4096 bits and ECC 256 and 384 bits*** that meet the following: *TPM library specification [TCG-X]*, ***FIPS 186-3.***

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 ***FIPS-approved DRBG algorithm ([SP800-90A])*** and specified cryptographic key sizes ***2048 bits, 3072 bits and 4096 bits,*** that meet the following: *TPM library specification [TCG-X],* ***FIPS 186-3.***

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 specified cryptographic key generation algorithm ***FIPS-approved DRBG algorithm ([SP800-90A])*** and specified cryptographic key size ***of 256 bits and 384 bits*** that meet the following: *TPM library specification [TCG-X]*, ***FIPS 186-3.***

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 ***FIPS-approved DRBG algorithm ([SP800-90A])*** and specified cryptographic key sizes ***128 and 256 bits*** that meet the following: *TPM library specification [TCG-X*], ***FIPS 186-3***.

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 ***zeroisation*** that meets the following: ***FIPS 140-2, Section 4.7.6***

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 modes CFB*, ***CTR and OFB*** and cryptographic key sizes *128 and 256 bits* that meet the following: *[SP800-38A] or [ISO10116:2006] or [ISO 18033-3]*

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, SHA-256* and *SHA-384* and cryptographic key sizes "*none"* that meet the following: [*FIPS 180-4]*

FCS_COP.1/HMAC Cryptographic operation (HMAC calculation)

Hierarchical to: No other components

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

FCS_COP.1.1/HMAC The TSF shall perform *HMAC value generation and verification* in accordance with a specified cryptographic algorithm *HMAC with SHA-1, SHA-256* and *SHA-384* and cryptographic key sizes ***of 20 bytes for SHA-1, 32 bytes for SHA-256 and 48 bytes for SHA-384*** that meet the following: [*FIPS 198-1] or [ISO9797-2].*

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

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

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

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 *2048 bits, 3072 bits* ***and 4096 bits*** that meet the following: [PKCS#1v2.1]

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 curves TPM_ECC_NIST_P256* and *TPM_ECC_NIST_P384,* and cryptographic key sizes *256 bits* and *384 bits* that meet the following: [*FIPS186-4] or[ISO 14888-3]*

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 curves *TPM_ECC_NIST_P256 and TPM_ECC_NIST_P384,* and cryptographic key size *256 bits* and *384 bits* that meet the following: *[TCG-X], [SP800-56A] or [ISO15946-1].*

6.1.4 Identification and Authentication SFR

FIA_SOS.2 TSF Generation of secrets

Hierarchical to: No other components
Dependencies: No dependencies

FIA_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet *uniform distribution of random variable generating the value.*

FIA_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for
(1) *nonce values for authorization sessions*.

FMT_MSA.4/AUTH Security attribute value inheritance

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

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

(1) The bits userWithAuth and adminWithPolicy in the TPMA_OBJECT of an object are defined when the object is created and can never be changed.
(2) User authorized by policy session is allowed to change the authPolicy by means of command TPM2_PolicyAuthorize or TPM2_PolicyAuthorizeNV.

FMT_MTD.1/AUTH Management of TSF data (user authorization)

Hierarchical to: No other components
Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/AUTH The TSF shall restrict the ability to

(1) set the platformAuth and platformPolicy to the role Platform firmware;
(2) set the endorsementAuth and endorsementPolicy to the role Platform Owner;
(3) set the ownerAuth and ownerPolicy to the role Privacy Administrator,
(4) set by TPM2_Duplicate the AuthValue or policyAuth of the object under the new parent to the same AuthValue or policyAuth of the duplicated object under the old parent to the role DUP.
(5) change the lockout parameters (TPM2_DictionaryAttackParameters) to the Lockout administrator.

FIA_AFL.1/Recover Authentication failure handling (recovery)

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/Recover The TSF shall detect when *maxTries* of unsuccessful authentication attempts occur related to *unsuccessful password or HMAC authentication attempts for*

(1) objects where DA is active (i.e., noDA attribute is CLEAR)
(2) NV Index where DA is active (i.e., the TPMA_NV_NO_DA attribute is CLEAR).

FIA_AFL.1.2/Recover When the defined number of unsuccessful authentication attempts has been *met,* the TSF shall *block the authorizations for RecoveryTime seconds.*
The counter failedTries is incremented when the authentication attempt failed.
The counter failedTries is decremented by one after recoveryTime seconds if:

(1) the TPM does not record an authorization failure of a DA-protected entity,
(2) there is no power interruption, and
(3) failedTries is not zero.

The counter failedTries is reset to 0 by

(1) command TPM2_Clear()
(2) TPM2_DictionaryAttackLockReset() with lockoutAuth

FIA_AFL.1/Lockout Authentication failure handling (lockout)

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/Lockout The TSF shall detect when *1* unsuccessful authentication attempts occur related to *failed authentication attempts with lockoutAuth using command TPM2_DictionaryAttackLockReset().*

FIA_AFL.1.2/Lockout When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall *block the TPM2_DictionaryAttackLockReset command for lockoutRecovery seconds.*

FIA_AFL.1/PINPASS Authentication failure handling

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/PINPASS The TSF shall detect when *pinCount*[1] successful authentication events exceed pinLimit for an NV Index with the attribute TPM_NT_PIN_PASS.

FIA_AFL.1.2/ PINPASS When the defined number of successful authentication events has been *met*, the TSF shall *block further authorization attempts*.

FIA_AFL.1/PINFAIL Authentication failure handling

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1/PINFAIL The TSF shall detect when *pinCount*[2] unsuccessful authentication attempts exceed pinLimit for an NV Index with the attribute TPM_NT_PIN_FAIL.

FIA_AFL.1.2/ PINFAIL When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall *block further authorization attempts*.

[1] *An administrator configurable **32-bit** positive integer.*

[2] *An administrator configurable **32-bit** positive integer.*

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),
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.

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components
Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.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*)
on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components
Dependencies: No dependencies

FIA_UAU.5.1 The TSF shall provide
(1) Password based authentication mechanism,
(2) HMAC based authentication mechanism,
(3) Policy based authentication mechanism
to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the *following rules:*

(1) If userWithAuth in the TPMA_OBJECT bits is set, for operations that require USER role authorization may be given if the caller provides proof of knowledge of the authValue of the object with an HMAC authorization session or a password. If this attribute is CLEAR, then HMAC or password authorizations may not be used for USER role authorizations.

(2) If the adminWithPolicy in the TPMA_OBJECT bits is set then HMAC or password authorizations may not be used for ADMIN role authorizations. If this attribute is CLEAR, then authorization for operations that require ADMIN role may be given if the caller provides proof of knowledge of the authValue of the object with an HMAC authorization session or a password.

(3) A password based authentication mechanism is required if the authHandle parameter of the command shall contain TPM_RS_PW.

(4) A HMAC or policy based authentication is required if the authHandle parameter of the command contain a valid handle of an authorization session.

- *(a) A HMAC based authentication is required if the authorization session shall be created with a sessionType of TPM_SE_HMAC,*
- *(b) A policy based authentication is required if the authorization session shall be created with a sessionType of TPM_SE_POLICY.*

(5) A policy based authentication mechanism verifies that a policy session provides a sequence of policy assertions combined in logical AND and OR relations, which policyDigest matches the authPolicy associated with the object and the other conditions of a policy session context are fulfilled. The assertions may express conditions for

- *(a) successful authentication with authValue defined for the authorized entity and the object to be accessed,*
- *(b) the command code of the authorized command to be executed,*
- *(c) the cpHash of the authorized command to be executed,*
- *(d) special condition for command TPM2_Duplicate(),*
- *(e) the locality of the authorized command to be executed,*
- *(f) the referenced object handle,*
- *(g) the current system time,*
- *(h) the content of the NV memory,*
- *(i) the value of selected PCR,*
- *(j) the assertion of physical presence if supported by the TOE,*
- *(k) the value of a shared secret,*
- *(l) the presence of a valid signature of the given parameters,*
- *(m) the value of the TPMA_NV_WRITTEN attribute of the specified NV index,*
- *(n) the value of the TPM_NT_PIN_PASS attribute of the specified NV index,*
- *(o) the value of the TPM_NT_PIN_FAIL attribute of the specified NV index,*
- *(p) the key template of the commands TPM2_CreatePrimary, TPM2_Create, and TPM2_CreateLoaded,*
- *(q) the validity of a Ticket.*

The TSF shall update the representation of the state of the TPM and its environment (policyDigest) on execution of the enhanced authorization commands defined in [TCG-2] section 23. The result of the updated policyDigest shall depend on the called command and its dedicated parameters.

(6) The command TPM2_PolicyRestart shall reset a policy authorization session to its initial state.

FIA_UAU.6 Re-authenticating

Hierarchical to: No other components

Dependencies: No dependencies

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions *that multiple commands need to be executed in one authorization session*.

FIA_USB.1 User-subject binding

Hierarchical to: No other components

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

(1) the shared secret for the TPM objects to access (sessionKey),
(2) the handle of opened authentication session,
(3) the physical presence if supported by the TOE and asserted,
(4) the state of the TPM and its environment (policyDigest).

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

(1) The TSF shall initialize the policyDigest value representing the state of the TPM and its environment with a zero digest (0…0). This shall take place at execution of the command TPM2_StartAuthSession.

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

(1) The TSF shall create the shared secret (sessionKey) and the session handle in the case of a session based authorization using the command TPM2_StartAuthSession.
(2) The TSF shall invalidate the shared secret (sessionKey) and the session handle in each of the following situations:
  (a) The command TPM2_FlushContext is executed for the corresponding session handle.
  (b) The flag continueSession of the session attributes is cleared.
  (c) The command TPM2_Startup is executed with the argument TPM_SU_CLEAR or TPM_SU_STATE.

6.1.5 TSF Protection

FPT_TST.1 TSF testing

Hierarchical to: No other components
Dependencies: No dependencies

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

*(1) at the request of the authorized user "**World**"*
(a) the TPM2_SelfTest command and of selected algorithms using the TPM2_IncrementalSelfTest command,

(2) at the conditions
(a) Initialization state after reset and before the reception of the first command,
(b) Prior to execution of the command using a not self-tested function,

(3) none

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

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of ***the objects kept in NV storage:***
- ***SPS,***
- ***Primary Keys,***
- ***User keys,***
- ***Context,***
- ***PCR data,***
- ***NV storage data where (TPMA_NV_PLATFORMCREATE == CLEAR)***
- ***Credentials.***

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

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

Hierarchical to: No other components
Dependencies: No dependencies

FPT_FLS.1.1/FS The TSF shall preserve a secure state *by entering the Fail state* when the following types of failures occur:
(1) If during TPM Restart or TPM Resume, the TPM fails to restore the state saved at the last Shutdown(STATE), the TPM shall enter Failure Mode and return TPM_RC_FAILURE.
(2) failure detected by TPM2_ContextLoad when the decrypted value of sequence is compared to the stored value created by TPM2_ContextSave(),
(3) failure detected by self test according to FPT_TST.1,
(4) Total reset counter overflow occurred
(5) RAM space allocation failure
(6) Illegal argument values encountered at stages which could only be due to attack
(7) Error encountered during random number generation

(8) Error encountered during NV Commit operation
(9) Error encountered during EK manufacture procedure
(10) Error encountered during NTC_FieldUpgrade command

FPT_FLS.1/SD Failure with preservation of secure state (shutdown)

Hierarchical to: No other components
Dependencies: No dependencies

FPT_FLS.1.1/SD The TSF shall preserve a secure state *by shutdown* when the following types of failures occur:
(1) detection of a physical attack,
(2) detection of environmental condition out of spec values.

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_ITT.1 Basic internal transfer protection

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

FDP_ITT.1.1 The TSF shall enforce the **TPM state control, TPM Object Hierarchy, Data import and export, Measurement and reporting, Access Control, NVM and Credential SFPs** to prevent the *disclosure* of user data when it is transmitted between physically-separated parts of the TOE

Refinement: *Even for single chip implementations, the different memories, the CPU and other functional units of the TOE (e.g., a cryptographic coprocessor) are seen as physically-separated parts of the TOE.*

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components
Dependencies: No dependencies

FPT_ITT.1.1 The TSF shall protect TSF data from *disclosure* when it is transmitted between separate parts of the TOE.

Refinement: *Even for single chip implementations, the different memories, the CPU and other functional units of the TOE (e.g., a cryptographic coprocessor) are seen as physically-*separated *parts of the TOE.*

6.1.6 TPM Operational States

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 initialization

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

Subjects as defined in Table 7:

(1) Platform firmware with the security attributes platformAuth, platformPolicy and physical presence if supported by the TOE,
(2) all other subjects; their security attributes are irrelevant for this SFP.

Objects as defined in Table 8 and Table 9:

(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) **Admin role is authorized to change** the TPM state to FUM if the authenticity of the first digest or the signature could be successfully verified.*
(2) While in FUM state the Platform firmware is authorized to import or activate firmware data only after successful verification of its integrity and authenticity (see FDP_UIT.1/States).
*(3) The FUM state shall only be left when **the verified firmware data has been signed using HMAC2, and reset signal has been detected.***
(4) In the Init state the subject "World" is authorized to execute the command TPM2_Startup and the sequence _TPM_Hash_Start, _TPM_Hash_Data, and _TPM_Hash_End.
(5) In the Init state every subject is authorized to process the Resume operation on the Shutdown BLOB with state transition to Operational.
(6) In the Init state every subject is authorized to process the Restart operation on the Shutdown BLOB with state transition to Operational.

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

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

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

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

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

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

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

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

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 initialization (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*** errors.

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

FDP_SDI.1 Stored data integrity monitoring

Hierarchical to: No other components
Dependencies: No dependencies

FDP_SDI.1.1 The TSF shall monitor user data stored in containers controlled by the TSF for *data modifications and modification of hierarchy* on all objects, based on the following attributes*: HMAC over the sensitive area of an object of the TPM hierarchy, object creation ticket*.

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

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

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

Subjects:

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

Objects:

(1) PPS,
(2) EPS,
(3) SPS,
(4) PPO,
(5) EK,
(6) SRK,
(7) Null Seed,
(8) object in a TPM hierarchy;

Operations:

(1) TPM2_CreatePrimary,
(2) TPM2_CreateLoaded,
(3) TPM2_HierarchyControl,
(4) TPM2_Clear,
(5) TPM2_ClearControl,
(6) TPM2_HierarchyChangeAuth,
(7) TPM2_SetPrimaryPolicy,
(8) TPM2_Load,
(9) TPM2_LoadExternal,
(10) TPM2_ReadPublic,
(11) Use.

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

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

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

Subjects:

(1) Platform Software with security attribute authorization state gained by authentication with platformAuth or platformPolicy,
(2) Platform Owner with security attribute authorization state gained by authentication with ownerAuth or ownerPolicy,
(3) Privacy administrator with security attribute authorization state gained by authentication with endorsementAuth or endorsementPolicy,
(4) Lockout administrator with security attribute authorization state,
(5) USER with authentication state gained with authValue or authPolicy,
(6) World with no security attributes;

Objects:

(1) EPS,
(2) PPS,
(3) SPS,
(4) EK,
(5) PPO,
(6) SRK,
(7) Null Seed,
(8) object in a TPM hierarchy with security attributes: state of the hierarchy, fixedParent, fixedTPM.

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

(1) The subject World is authorized to create an EPS whenever the TPM is powered on and no EPS is present.

(2) The subject World is authorized to create a PPS whenever the TPM is powered on and no PPS is present.

(3) The subject World is authorized to create an SPS whenever the TPM is powered on and no SPS is present.

(4) The subject World is authorized to create a Null Seed whenever the TPM is reset.

(5) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth is authorized to change the SPS to a new value from the RNG (TPM2_Clear). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_Clear command.

(6) The Platform firmware is authorized to create a Platform Primary Object under PPS. The physical presence is not required if it is not if supported by the TOE or disabled for TPM2_CreatePrimary or TPM2_CreateLoaded command.

(7) The Platform owner is authorized to create a primary object (SRK) under SPS.

(8) The privacy administrator is authorized to create a primary object (EK) under EPS.

(9) The subject World is authorized to create temporary objects for no hierarchy (using the Null Seed).

(10) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth are authorized to remove all TPM context associated with a specific owner (TPM2_Clear). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_ClearControl command.

(11) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the lockout administrator with lockoutAuth are authorized to disable and enable the execution of TPM2_Clear by the command TPM2_ClearControl. The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_ClearControl command.

(12) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE, the Platform owner, the privacy administrator and the lockout administrator are authorized to change the authorization secret for a hierarchy or lockout (TPM2_HierarchyChangeAuth). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyChangeAuth command.

(13) The Platform firmware with platformAuth, platformPolicy or physical presence, if supported by the TOE the Platform owner and the privacy administrator are authorized to set the authorization policy for the platform hierarchy (platformPolicy), the storage hierarchy (ownerPolicy) and the endorsement hierarchy (endorsementPolicy) using the command TPM2_SetPrimaryPolicy. The physical presence is not required if it is not

supported by the TOE or disabled for the TPM2_SetPrimaryPolicy command.

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

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

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

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

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

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

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

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

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

FMT_MSA.3.2/Hier The TSF shall allow *the creator of an object in a TPM hierarchy* to specify alternative initial values to override the default values when an object or information is created.

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

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

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

(1) The Platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE is authorized to enable and to disable the use of the platform hierarchy and its associated NV storage

(TPM2_HierarchyControl changing phEnable or phEnableNV). The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_HierarchyControl command.

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

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

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

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

6.1.7 Data Import and Export

FDP_ACC.1/ExIm Subset access control (export and import)

Hierarchical to: No other components

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/ExIm The TSF shall enforce *the Data Export and Import SFP* on

Subjects:
(1) USER,
(2) DUP,
(3) World;

Objects:
(1) Platform Primary Key,
(2) Endorsement Primary Key,
(3) Storage Primary Key
(4) User Key,
(5) Context;

Operations:
(1) duplicate by means of TPM2_Duplicate,
(2) export by means of TPM2_Create,
(3) load by means of TPM2_Load,
(4) export and load by means of TPM2_CreateLoaded,
(5) load by means of TPM2_LoadExternal,
(6) import by means of TPM2_Import,
(7) unseal by means of TPM2_Unseal,
(8) save by means of TPM2_ContextSave,
(9) load by means of TPM2_ContextLoad,
(10) remove a context by means of TPM2_FlushContext.

FDP_ACF.1/ExIm Security attribute based access control (export and import)

Hierarchical to: No other components

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

FDP_ACF.1.1/ExIm The TSF shall enforce the *Data Export and Import SFP* to objects based on the following:

Subjects:
(1) USER with authentication state gained with authValue or authPolicy,
(2) DUP with authentication state gained with authPolicy,
(3) World without any successful authentication;

Objects:
(1) Platform Primary Object with the security attributes platformAuth,
(2) Endorsement Primary Key with the security attributes authorization data,
(3) User Key with the security attributes authorization data,
(4) Context with the security attributes sequence number, hierarchy selector, HMAC.

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

(1) The subject DUP is authorized to duplicate a loaded object under the following conditions:
- *(a) the authorization of the subject shall be provided in an authorization session for duplication,*
- *(b) the object attribute "fixedParent" must not be set, and*
- *(c) the object attribute "nameAlg" must not be TPM_ALG_NULL.*

(2) The subject USER is authorized to export an object using the TPM2_Create command.

(3) The subject USER authorized for the parent object is allowed to load objects into the TPM hierarchy using the command TPM2_Load.

(4) The subject USER is authorized to export and load an object using the TPM2_CreateLoaded command.

(5) The subject World is authorized to load public objects into any TPM hierarchy using the command TPM2_LoadExternal.

(6) The subject USER authorized for the parent object is allowed to import an object using the TPM2_Import command under the following conditions:
- *(a) The attributes "fixedTPM" and "fixedParent" of the object shall not be set.*
- *(b) If an encryption of the object to import is performed, then an integrity evidence value shall be part of the imported object.*
- *(c) If an integrity evidence value is present, the object shall only be imported after the integrity was successfully verified.*

(7) The subject World is authorized to read the public portion of a TPM object using the command TPM2_ReadPublic.

(8) The subject USER is authorized to unseal a sealed data object using the TPM2_Unseal command.

(9) Every subject is authorized to save a context without authorization.

(10) Every subject is authorized to load a saved context without authorization if
- *(a) the sequence number is in the accepted range,*
- *(b) the integrity of the context is successfully verified,*
- *(c) the TPM was not reset after the context saving and*
- *(d) the hierarchy associated with the context was not changed or disabled.*

(11) Every subject is authorized to remove all context associated with a loaded object or session from the TPM memory (TPM2_FlushContext).

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

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

(1) No subject is authorized to move an object to another TPM's object hierarchy (using the duplicate and import operation) if the fixedTPM or the fixedParent attribute of that object is set.

(2) No subject is authorized to move an object to another position in a TPM object hierarchy (using the duplicate operation) if the fixedParent attribute of that object is set.

FMT_MSA.1/ExIm Management of security attributes (export and import)

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/ExIm TSF shall enforce the *Data Export and Import SFP* to restrict the ability to *use* the security attributes *authorization data* to *every subject*.

FMT_MSA.3/ExIm Static attribute initialization (export and import)

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

FMT_MSA.3.1/ExIm The TSF shall enforce the *Data Export and Import SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

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

FDP_ETC.2/ExIm Export of user data with security attributes (export and import)

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

FDP_ETC.2.1/ExIm The TSF shall enforce the *Data Export and Import SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.2.2/ExIm The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3/ExIm The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4/ExIm The TSF shall enforce the following rules when user data is exported from the TOE:

(1) The sensitive area of an object from the TPM hierarchy shall be integrity-protected with an HMAC before its export using the command TPM2_Create or TPM2_CreateLoaded. The used key and the IV shall be derived from the secret seed of the parent in the TPM hierarchy.

(2) The sensitive area of an object from the TPM hierarchy shall be symmetrically encrypted before its export using the command TPM2_Create or TPM2_CreateLoaded. The used key and the IV should be derived from the secret seed of the parent in the TPM hierarchy.
(3) An exported context (using the command TPM2_ContextSave) shall be symmetrically encrypted and integrity protected with a HMAC.
(4) When exporting an object using the command TPM2_Duplicate then the following actions shall be performed:
(a) If the encryptedDuplication attribute is set or the caller provides a symmetric algorithm then the sensitive part of the data shall be symmetrically encrypted and integrity protected (called: inner duplication wrapper).
(b) *If the encryptedDuplication attribute is set or the caller provides a new parent in a TPM hierarchy then the inner duplication wrapper shall be symmetrically encrypted and integrity protected (called outer duplication wrapper). The used key shall be derived from a seed that shall be asymmetrically encrypted with the public key of the intended new parent in the TPM object hierarchy*.

FDP_ITC.2/ExIm Import of user data with security attributes (export and import)

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] FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1/ExIm The TSF shall enforce the *Data Export and Import SFP* when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2/ExIm The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3/ExIm The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4/ExIm The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5/ExIm The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE:
(1) *If an inner or an outer wrapper is present then a valid integrity value shall be present*.

FDP_UCT.1/ExIm Basic data exchange confidentiality (export and import)

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

FDP_UCT.1.1/ExIm The TSF shall enforce the *Data Export and Import SFP* to *transmit* user data in a manner protected from unauthorized disclosure.

FDP_UIT.1/ExIm Data exchange integrity (export and import)

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/ExIm The TSF shall enforce the *Data Export and Import SFP* to *transmit and receive* user data in a manner protected from modification errors.

FDP_UIT.1.2/ExIm The TSF shall be able to determine on receipt of user data, whether *modification* has occurred.

6.1.8 Measurement and reporting

FDP_ACC.1/M&R Subset access control (measurement and reporting)

Hierarchical to: No other components

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/M&R The TSF shall enforce the *Measurement and Reporting SFP* on

Subjects:

(1) Platform firmware,
(2) USER,
(3) ADMIN,
(4) World;

Objects:

(1) PCR,
(2) TPM objects;

Operations:

(1) TPM2_PCR_Allocate,
(2) TPM2_PCR_Reset,
(3) TPM2_PCR_Extend,
(4) TPM2_PCR_Event,
(5) TPM2_PCR_Read,
(6) TPM2_Quote,
(7) TPM2_CertifyCreation.

FDP_ACF.1/M&R Security attribute based access control (measurement and reporting)

Hierarchical to: No other components

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

FDP_ACF.1.1/M&R The TSF shall enforce the *Measurement and Reporting SFP* to objects based on the following:

Subjects:

(1) Platform firmware with security attribute authorization state gained by authentication with platformAuth or platformPolicy or locality,
(2) USER with authentication state gained with authValue or authPolicy,
(3) ADMIN with authentication state gained with authValue or authPolicy,
(4) World with no security attributes;

Objects:

(1) PCR with the security attribute PCR-attributes TPM_PT_PCR,
(2) TPM objects with the security attributes authentication data (authValue, authPolicy).

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

(1) The Platform firmware authorized with platformAuth, platformPolicy or with physical presence if supported by the TOE is authorized to set the desired PCR allocation of the PCR and the algorithms (TPM2_PCR_Allocate). The

physical presence is not required if it is not supported by the TOE or disabled for TPM2_PCR_Allocate command.

(2) Authorized subjects of role USER are allowed to extend the PCR using the command TPM2_PCR_Extend if the command locality permits the extension of the intended PCR.

(3) Authorized subjects of role USER are allowed to update the PCR using the command TPM2_PCR_Event if the command locality permits the extension of the intended PCR.

(4) Authorized subjects of role USER are allowed to reset the PCR using the commands TPM2_PCR_Reset if the command locality permits the reset attribute of the PCR.

(5) The subject World is authorized to read values of PCR using the command TPM2_PCR_Read.

(6) Authorized subjects of role USER are allowed to quote PCR values using the command TPM2_Quote. The authorization shall be done based on the key that is used for the quotation.

(7) Authorized subjects of role USER are allowed to prove the association between an object and its creation data by creation of a ticket using the command TPM2_CertifyCreation. The authorization shall be done based on the key that is used to sign the attestation block.

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

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

FMT_MSA.1/M&R Management of security attributes (measurement and reporting)

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/M&R TSF shall enforce the *Measurement and Reporting SFP* to restrict the ability to *modify* the security attributes *PCR extension algorithm, used hash algorithm to Platform firmware*.

FMT_MSA.3/M&R Static attribute initialization (measurement and reporting)

Hierarchical to: No other components

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1/M&R The TSF shall enforce the *Measurement and Reporting SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

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

FCO_NRO.1/M&R Selective proof of origin (measurement and reporting)

Hierarchical to: No other components

Dependencies: FIA_UID.1 Timing of identification

FCO_NRO.1.1/M&R The TSF shall be able to generate evidence of origin for transmitted *attestation structure (TPM2B_ATTEST) and object creation tickets* at the request of the *originator*.

FCO_NRO.1.2/M&R The TSF shall be able to relate the

(1) magic number for identification whether the TPM produced the signed digest or any external entity,
(2) type of the attestation structure indicating the contents of the attested parameter,
(3) qualified name of the key used to sign the attestation data (qualifiedSigner),
(4) external information supplied by the caller,
(5) *values of clock, resetCount*, restartCount and Safe,
(6) the firmware version

of the originator of the information, and the *command depending value of either*

(1) PCR data (using the command TPM2_Quote), or
(2) audit digests (using the command TPM2_GetSessionAuditDigest), or
(3) a ticket that was produces by the TPM (using the command TPM2_CertifyCreation)

of the information to which the evidence applies.

FCO_NRO.1.3/M&R The TSF shall provide a capability to verify the evidence of origin of information to *recipien*t given *as soon as the recipient can verify the signature and has confidence to the key that is used to sign*.

6.1.9 Access SFR

FDP_ACC.1/AC Subset access control (access control)

Hierarchical to: No other components

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/AC *The TSF shall enforce the Access Control SFP on:*

Subjects:

(1) Platform firmware,
(2) Platform owner,
(3) Privacy administrator,
(4) Lockout administrator,
(5) USER,
(6) DUP,
(7) ADMIN,
(8) World;

Objects:

(1) User key,
(2) TPM objects,
(3) Clock
(4) Data (to which cryptographic operation applies);

Operations:

(1) TPM2_EvictControl,
(2) TPM2_ClockSet,
(3) TPM2_ClockRateAdjust,
(4) TPM2_ReadClock,
(5) TPM2_GetTime,
(6) TPM2_VerifySignature,
(7) TPM2_Sign,
(8) TPM2_GetRandom,
(9) TPM2_StirRandom,
(10) TPM2_RSA_Encrypt,
(11) TPM2_RSA_Decrypt,
(12) TPM2_ECDH_KeyGen,
(13) TPM2_ECDH_ZGen,
(14) TPM2_ECC_Parameters,
(15) TPM2_HMAC_Start,
(16) TPM2_HashSequenceStart,
(17) TPM2_SequenceUpdate,
(18) TPM2_SequenceComplete,
(19) TPM2_EventSequenceComplete,
(20) TPM2_HMAC,
(21) TPM2_Hash.

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

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

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

Subjects:

(1) Platform firmware with security attribute authorization state gained by authentication with platformAuth, platformPolicy or physical presence if supported by the TOE,
(2) Platform owner with security attribute authorization state gained by authentication with ownerAuth or ownerPolicy,
(3) Privacy administrator with security attribute authorization state gained by authentication with endorsementAuth or endorsementPolicy,
(4) Lockout administrator with security attribute authorization state,
(5) USER with authentication state gained with userAuth or authPolicy,
(6) DUP with authentication state gained with authPolicy,
(7) ADMIN with authentication state gained with userAuth or authPolicy,
(8) World with no security attributes;

Objects:

(1) User key with security attributes TPM_ALG_ID, TPMA_OBJECT,
(2) TPM objects,
(3) Clock with security attributes: resetCount, restartCount, safe-flag,
(4) Data with security attribute "externally provided".

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

(1) The Platform firmware authorized with platformAuth, platformPolicy or with physical presence if supported by the TOE and the Platform owner are authorized 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 Platform owner are authorized 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 authorized to get the current value of time, clock, resetCount, restartCount and safe (TPM2_ReadClock).
(4) A subject with the role USER endorsed by the Privacy administrator and the keyHandle identifier of a loaded key that can perform digital signatures is authorized to get the current value of time and clock (TPM2_GetTime).
(5) No subject is authorized to set the clock to a value less than the current value of clock using the TPM2_ClockSet command.
(6) No subject is authorized to set the clock to a value greater than its maximum value (0xFFFF000000000000) using the TPM2_ClockSet command.

(7) A subject with the role USER is authorized to generate digital signatures using the command TPM2_Sign for externally provided data (hash). The user authorization shall be done based on the required authorization of the key that will perform signing. The key attributes shall allow the signing operation for externally provided data.
(8) Any subject is authorized to verify digital signatures using the command TPM2_VerifySignature.
(9) Any subject is authorized to request data from the random number generator using the command TPM2_GetRandom.
(10) Any subject is authorized to add additional information to the state of the random number generator using the command TPM2_StirRandom.
(11) Any subject is authorized to perform RSA encryption using the command TPM2_RSA_Encrypt for externally provided data. The key attributes shall allow the encrypt operation for externally provided data.
(12) A subject with the role USER is authorized to perform RSA decryption using the command TPM2_RSA_Decrypt for externally provided data. The user authorization shall be done based on the required authorization of the key that will be used for decryption. The key attributes shall allow the decrypt operation for externally provided data.
(13) Any subject is authorized to generate ECC ephemeral key pairs using the command TPM2_ECDH_KeyGen.
(14) A subject with the role USER is authorized to recover a value that is used in ECC based key sharing protocols using the command TPM2_ECDH_ZGen. The user authorization shall be done based on the required authorization of the involved private key.
(15) Any subject is authorized to request the parameters of an identified ECC curve using the command TPM2_ECC_Parameters.
(16) The subject USER is authorized to start a HMAC sequence using the command TPM2_HMAC_Start.
(17) The subject World is authorized to start a hash or event sequence using the command TPM2_HashSequenceStart.
(18) The subject USER is authorized to add data to a hash, event or HMAC sequence using the command TPM2_SequenceUpdate.
(19) The subject USER is authorized to add the last part of data (if any) to a hash or HMAC sequence using the command TPM2_ SequenceComplete.
(20) The subject USER is authorized to add the last part of data (if any) to an event sequence using the command TPM2_EventSequenceComplete.
(21) Any subject is authorized to perform hash operations on a data buffer using the command TPM2_Hash.
(22) A subject with the role USER is authorized to perform HMAC operations on a data buffer. The user authorization shall be done based on the required authorization of the involved symmetric key.
(23) A subject with the role USER is authorized to generate HMACs using the command TPM2_HMAC for externally provided data (hash). The user authorization shall be done based on the required authorization of the key that will perform the HMAC. The key attributes shall allow the signing operation for externally provided data.

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

FDP_ACF.1.4/AC The TSF shall explicitly deny access of subjects to objects based on the following additional rules: ***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/AC Management of security attributes (access control)

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/AC TSF shall enforce *the Access Control SFP* to restrict the ability to

(1) *query* the security attributes *digital signature of the audit session digest (TPM2_GetSessionAuditDigest)* to *privacy administrator*

(2) query the security attributes TPMT_PUBLIC_PARMS (TPM2_TestParms) to World.

(3) Query the security attributes TPMS_ALGORITHM_DETAILS_ECC (TPM2_ECC_Parameters) to World.

(4) increment the security attributes resetCount and restartCount to every subject,

(5) reset the security attributes resetCount, restartCount and the safe-flag of the TPM Clock by means of command TPM2_Clear to Platform firmware authorized by platformAuth, platformPolicy or physical presence (if supported by the TOE) and the lockout administrator,

(6) if supported by the TOE: change the security attribute Physical Presence requirement for all commands in the setList of TPM2_PP_Comands to "required" and all commands in the clearList_to "not required" of TPM2_PP_Commands to Platform firmware authorized by platformAuth, platformPolicy or physical presence,

(7) change the security attributes authorization secret (authValue) of TPM objects (TPM2_ObjectChangeAuth) to ADMIN.

FMT_MSA.3/AC Static attribute initialization (access control)

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

FMT_MSA.3.1/AC The TSF shall enforce *the Access Control SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/AC The TSF shall allow the *USER, ADMIN* to specify alternative initial values to override the default values when an object or information is created.

FDP_UCT.1/AC Basic data exchange confidentiality (access control)

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

FDP_UCT.1.1/AC The TSF shall enforce the *Access Control SFP to transmit* user data in a manner protected from unauthorized disclosure.

FTP_ITC.1/AC Inter-TSF trusted channel (access control)

Hierarchical to: No other components
Dependencies: No dependencies

FTP_ITC.1.1 The TSF shall provide 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.

FTP_ITC.1.2 The TSF shall permit *another trusted IT product* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for
(1) an authorization session,
(2) an encryption session, identified by the encrypt or decrypt attribute of the session
in order to transfer commands and responses between the other trusted IT product and the TOE.

FMT_MOF.1/AC Management of security functions behaviour (access control)

Hierarchical to: No other components
Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1/AC The TSF shall restrict the ability *to disable and enable* the functions *TPM2_Clear to Platform firmware and the lockout administrator*.

6.1.10 Non-Volatile Storage

FDP_ACC.1/NVM Subset access control (non-volatile memory)

Hierarchical to: No other components

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/NVM The TSF shall enforce the NVM SFP on

Subjects:

(1) Platform firmware,
(2) Platform owner,
(3) USER,
(4) ADMIN,
(5) World;

Objects:

(1) (ordinary, counter, bit field, extended) NV index,
(2) objects of the TPM hierarchy;

Operations:

(1) TPM2_NV_DefineSpace,
(2) TPM2_NV_UndefineSpace,
(3) TPM2_NV_UndefineSpaceSpecial,
(4) TPM2_NV_Read,
(5) TPM2_NV_ReadPublic,
(6) TPM2_NV_Increment,
(7) TPM2_NV_Extend,
(8) TPM2_NV_SetBits,
(9) TPM2_NV_Write,
(10) TPM2_NV_ReadLock,
(11) TPM2_NV_WriteLock,
(12) TPM2_NV_Certify,
(13) TPM2_EvictControl.

FDP_ACF.1/NVM Security attribute based access control (non-volatile memory)

Hierarchical to: No other components

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

FDP_ACF.1.1/NVM The TSF shall enforce the *NVM SFP* to objects based on the following:

Subjects as defined in Table 7:

(1) Platform firmware, Platform owner, USER, ADMIN, World with the security attributes:
  (a) authentication status,
  (b) physical presence if supported by the TOE.

Objects as defined in Table 8:

(1) NV index, NV counter index, NV bit field index, NV extend index, NV pin pass index, NV pin fail index with the security attributes:

(a) NV attributes,

(b) status whether physical presence is required for Platform firmware authorization.

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

(1) The Platform firmware authenticated with platformAuth, platformPolicy or physical presence if supported by the TOE and the Platform owner are authorized to reserve space to hold the data associated with that index (TPM2_NV_DefineSpace). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_NV_DefineSpace command.

(2) The Platform firmware authenticated with platformAuth, platformPolicy or physical presence if supported by the TOE and the Platform owner are authorized to remove a NV index (TPM2_NV_UndefineSpace). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_NV_UndefineSpace command.

(3) The Platform firmware authenticated with platformAuth, platformPolicy or physical presence if supported by the TOE is authorized to remove a platform created NV index that has the attribute TPMA_NV_POLICY_DELETE set (TPM2_NV_UndefineSpaceSpecial). The physical presence is not required if it is not supported by the TOE or disabled for TPM2_NV_UndefineSpaceSpecial command.

(4) Any subject is authorized to read the public area of a NV index by the command TPM2_NV_ReadPublic.

(5) The subject Platform firmware with the role USER is authorized to read a NV index by the command TPM2_NV_Read if the TPMA_NV_PPREAD value of the NV index attribute is set and the NV index is not temporarily blocked by its attribute TPMA_NV_READLOCKED. If the TPMA_NV_AUTHREAD attribute is set then the authentication shall use authValue of the index, if the TPMA_NV_POLICYREAD attribute is set then the authentication shall use authPolicy of the index.

(6) The subject Platform owner with the role USER is authorized to read a NV index by the command TPM2_NV_Read if the TPMA_NV_OWNERREAD value of the NV index attribute is set and the NV index is not temporarily blocked by its attribute TPMA_NV_READLOCKED. If the TPMA_NV_AUTHREAD attribute is set then the authentication shall use authValue of the index, if the TPMA_NV_POLICYREAD attribute is set then the authentication shall use authPolicy of the index.

(7) The subject Platform firmware with the role USER is authorized to write to a NV index if the TPMA_NV_PPWRITE value of the NV index attribute is set and the NV index is not temporarily blocked by its attribute TPMA_NV_WRITELOCKED or permanently blocked by its attribute TPM_NV_WRITEDEFINE. If the TPMA_NV_AUTHWRITE attribute is set then the authentication shall use authValue of the index, if the TPMA_NV_POLICYWRITE attribute is set then the authentication shall use authPolicy of the index.

(8) The subject Platform owner with the role USER is authorized to write to a NV index if the TPMA_NV_OWNERWRITE value of the NV index attribute

is set and the NV index is not temporarily blocked by its attribute TPMA_NV_WRITE_STCLEAR or permanently blocked by its attribute TPM_NV_WRITEDEFINE. If the TPMA_NV_AUTHWRITE attribute is set then the authentication shall use authValue of the index, if the TPMA_NV_POLICYWRITE attribute is set then the authentication shall use authPolicy of the index.

(9) An authorized subject to write a NV index (see number 7 and 8) is allowed to update a NV counter index only in the following way:

a) The modification shall only be possible using the command TPM2_NV_Increment. The command TPM2_NV_Increment shall increment the value of the NV counter index by one.

b) The TPM shall ensure that, when a NV counter index is read, its value is not less than a previously reported value of the counter.

(10) An authorized subject to write a NV index (see number 7 and 8) is allowed to update a NV index of type "Extend" only by the command TPM2_NV_Extend.

(11) An authorized subject to write a NV index (see number 7 and 8) is allowed to update a NV index of type "Bit Field" only by the command TPM2_NV_SetBits.

(12) An authorized subject to write a NV index (see number 7 and 8) is allowed to update a NV index that is not of type "Bit Field", "Counter" or "Extend" by the command TPM2_NV_Write.

(13) The subject platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the Platform owner are authorized to import transient TPM objects if they are part of any TPM hierarchy, if the object attributes allow the import and if the objects contain both public and private portions. This shall be done by the command TPM2_EvictControl. The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_EvictControl command.

(14) *The subject platform firmware with platformAuth, platformPolicy or physical presence if supported by the TOE and the Platform owner are authorized to delete persistent TPM objects if the object attributes allow the deletion. This shall be done by the command TPM2_EvictControl. The physical presence is not required if it is not supported by the TOE or disabled for the TPM2_EvictControl command.*

(15) *An authorized subject is allowed to certify the contents of an NV index or a portion of an NV index using the command TPM2_NV_Certify*

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

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

(1) If phEnableNV is CLEAR

a) NV indices that have TPMA_PLATFORM_CREATE SET may not be read by TPM2_NV_Read, TPM2_NV_ReadPublic, TPM2_NV_Certify, TPM2_PolicyNV or written, by TPM2_NV_Write, TPM2_NV_Increment, TPM2_NV_Extend, TPM2_NV_SetBits (TPM_RC_HANDLE).

b) The platform cannot define (TPM_RC_HIERARCHY) or undefined (TPM_RC_HANDLE) indices.

FMT_MSA.1/NVM Management of security attributes (non-volatile memory)

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/NVM TSF shall enforce the *NVM SFP* to restrict the ability to *modify* the security attributes *NV index attributes to the authorized role of the subject that executes the NV related command*.

FMT_MSA.3/NVM Static attribute initialization (non-volatile memory)

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

FMT_MSA.3.1/NVM The TSF shall enforce the *NVM SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

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

FMT_MSA.4/NVM Security attribute value inheritance (NVM)

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

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

(1) If TPMA_NV_READ_STCLEAR of the NV Index is SET and the authPolicy of the NV Index is provided and
a) TPMA_NV_PPREAD is set and platformAuth is provided or
b) TPMA_NV_OWNERREAD is set and ownerAuth is provided or
c) TPMA_NV_AUTHREAD is set and authValue is provided
c) the command TPM2_NV_ReadLock shall SET TPMA_NV_READLOCKED for the NV Index. TPMA_NV_READLOCKED will be CLEAR by the next TPM2_Startup(TPM_SU_CLEAR).

(2) If TPMA_NV_WRITEDEFINE or TPMA_NV_WRITE_STCLEAR attributes of an NV location are SET and the authPolicy of the NV Index is provided or

a) TPMA_NV_PPWRITE is set and platformAuth is provided or
b) TPMA_NV_OWNERWRITE is set and ownerAuth is provided or
c) TPMA_NV_AUTHWRITE is set and authValue is provided
d) the command TPM2_NV_WriteLock shall SET TPMA_NV_WRITELOCKED for the NV Index. TPMA_NV_WRITELOCKED will be clear on the next TPM2_Startup(TPM_SU_CLEAR) unless TPMA_NV_WRITEDEFINE is SET.

FMT_MTD.1/NVM Management of TSF data (non-volatile memory)

Hierarchical to: No other components
Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/NVM The TSF shall restrict the ability to modify *the authorization secret (authValue) for a NV index* to *ADMIN using the command TPM2_NV_ChangeAuth.*

FDP_ITC.1/NVM Import of user data without security attributes (non-volatile memory)

Hierarchical to: No other components
Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialization

FDP_ITC.1.1/NVM The TSF shall enforce the *NVM SFP* when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2/NVM The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3/NVM The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *none*

FDP_ETC.1/NVM Export of user data without security attributes (non-volatile memory)

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

FDP_ETC.1.1/NVM The TSF shall enforce the *NVM SFP* when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2/NVM The TSF shall export the user data without the user data's associated security attributes.

6.1.11 Credentials

FDP_ACC.1/Cre Subset access control (credentials)

Hierarchical to: No other components

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Cre The TSF shall enforce the *Credential SFP* on

Subjects:

(1) USER,
(2) ADMIN,
(3) World;

Objects:

(1) Credential;

Operations:

(1) TPM2_ActivateCredential,
(2) TPM2_MakeCredential.

FDP_ACF.1/Cre Security attribute based access control (credentials)

Hierarchical to: No other components

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

FDP_ACF.1.1/Cre The TSF shall enforce the *Credential SFP* to objects based on the following:

Subjects:

(1) USER with authentication state gained with userAuth or authPolicy,
(2) ADMIN with authentication state gained with authValue or authPolicy,
(3) World with no security attributes;

Objects:

(1) *Credential with security attribute HMAC over the credential BLOB*.

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

(1) The subject World is authorized to create a credential using the command TPM2_MakeCredential.
(2) *The subject of role ADMIN regarding the object for which the credential was created and the role USER regarding the key for the decryption of the credential BLOB is authorized to activate the credential using the command TPM2_ActivateCredential*.

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

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

FMT_MSA.3/Cre Static attribute initialization (credentials)

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

FMT_MSA.3.1/Cre The TSF shall enforce the *Credential SFP* to provide *restrictive* default values for security attributes that are used to enforce the SFP.

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

FMT_MSA.1/Cre Management of security attributes (credentials)

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/Cre TSF shall enforce *the Credential SFP* to restrict the ability to *use* the security attributes *HMAC in the credential BLOB* to *USER*.

FCO_NRO.1/Cre Selective proof of origin (credentials)

Hierarchical to: No other components
Dependencies: FIA_UID.1 Timing of identification

FCO_NRO.1.1/Cre The TSF shall be able to generate evidence of origin for transmitted *TPM objects* at the request of the *originator*.

FCO_NRO.1.2/Cre The TSF shall be able to relate *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.

FCO_NRO.1.3/Cre The TSF shall provide 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*.

6.2 Security Assurance Requirements for the TOE

The Security Assurance Requirements (SAR) for the TOE are the assurance components of Evaluation Assurance Level 4 (EAL4), as defined in [CC] and augmented with ALC_FLR.1 ,AVA_VAN.4 and ALC_DVS.2.

7 TOE Summary Specification

The TOE summary specification in the following section specifies the security functionality as well as the assurance measures of the TOE.

7.1 TOE Security Features

The TOE consists of eight security features (SF) to meet the Security Functional Requirements.

SF1: Cryptographic Operations
SF2: Self Test
SF3: Access Control
SF4: Hacking and physical tampering protection/detection
SF5: Key Management
SF6: Random Number Generation
SF7: Identification and Authentication
SF8: Firmware Field Upgrade

7.1.1 SF1 – Cryptographic Operations

There are three functions within the TPM related to cryptographic operations:

1. Asymmetric (public key) cryptography in the form of RSA digital signature generation and verification, RSA encryption and decryption, ECC digital signature generation and verification, ECC key agreement, and key derivation
2. Symmetric key cryptography in the form of AES encryption and decryption and HMAC signatures
3. Hash generation

7.1.2 SF2 – Self Test

The TOE supports a suite of self tests to check and demonstrate the correct operation of the TOE security.

7.1.3 SF3 – Access Control

The TOE provides a set of access control security function policies (called hereafter globally *Protected Operations Access Controls (*POAC), comprising access control policies documented in the FDP_ACC.1 iterations) to protect the sensitive NV objects of the TPM.

The TOE enforces the POAC policy on NV. The TOE provides access control by denying access to some objects based on attributes such as TPMA_NV_READ_STCLEAR and TPMA_NV_WRITE_STCLEAR. For a TPM compatible with this specification, use of PCR for

access control requires a policy. The policy should be created at the time of object creation so that the policy requires the selected PCR to have a specific value.

7.1.4 SF4 – Hacking and Physical Tampering Protection/Detection

The TOE supports the following functionality for protection against and detection of hacking and physical tampering:

- Tamper evidence: The TOE is provided in a single package. Any intent to gain physical access to the TPM protected areas will result in obvious damage to the TOE enclosure.
- Snooping protection/detection: The TOE is equipped with a mechanism for protection against snooping the user data or the design during operation

7.1.5 SF5 – Key Management

The TOE supports generation of asymmetric cryptographic key pairs in accordance with the specified cryptographic key generation algorithm RSA and specified cryptographic key sizes RSA 2048, 3072 and 4096 bits, as defined by [PKCS#1 V2.1]. The source of randomness is the TOE Random Number Generator (**RNG**). The generate function is a protected capability and the private key is held in a shielded location. The TOE supports generation of ECC keys in accordance with [FIPS 186-3], section B.4.1 "Key Pair Generation Using Extra Random Bits".

Key generation produces three different types of keys. The first, an ordinary key, is produced using the RNG to seed the computation. The result of the computation is a secret key value kept in a Shielded Location.

The second type, a Primary Key, is derived from a seed value, not the RNG directly. The RNG usually generates the seed that is usually persistently stored on the TPM. Generation of a Primary Key from a seed is based on use of an approved key derivation function (**KDF**). The KDF from [SP800-108] is widely used in this specification.

The third type is derived keys, which are generated from the sensitive value of the parent key.

The storage of keys in shielded locations is specified in [TCG-1] Clause 22 Protected Storage]. Specifically, the destruction of keys is done according to FIPS 140-2 section 4.7.6.

7.1.6 SF6 – Random Number Generation

The TPM supports generation of random numbers using HW RNG module. The HW Random Number Generator is based on physical probabilistic controlled effects. It is implemented with conformance to [SP800-90A] and [FIPS 140-2].

7.1.7 SF7 – Identification and Authentication

The TOE identification and authentication capability is used to authorise the use of a Protected Object and Protected Capability. [TCG-1] Clause 19 Authorizations and

Acknowledgements refers to the identification and authentication process, and their related data, as authorization. Authentication is achieved either by knowledge of a shared secret (password or HMAC secret) named 'authValue' assigned to the entity, or by verification of a specific state of the TOE encoded in an 'authPolicy' which is assigned to the entity. The authorization may be for a command only or session based – with session type of either HMAC or Policy. Session based authorisation uses *handles* and random *nonces*. The handle is assigned when the session is created and identifies the session until the session is closed, while a nonce is used only for one message and its reply. The nonces are used by both sides of the transaction to compute command-dependent authentication values using secrets or shared secrets and nonce-data.

Protected entities and their authentication data may be stored persistently in the TPM or outside the TPM.

SF7 supplies the verification of evidence of origin for transmitted data signed using identity keys, by using either RSA algorithm or ECC and KDFe for secret decryption.

7.1.8 SF8 – Firmware Field Upgrade

The TOE provides a secure method to upgrade the Upgradable Software part of the firmware.

The Field Upgrade process does not expose the firmware as plain text and uses authentication to verify the integrity and source of the firmware. This is achieved by using ECC signature scheme ECDSA with curve NIST P-384 and SHA-256 algorithms and Nuvoton's ECC 384 bits Key.

If the Field Upgrade process succeeds, then the resulting product is the Final TOE.

The TOE has dedicated TPM commands that reports the version of the Upgradable Software and the BootLoader (see Section 1.1).

7.1.9 Assignment of SFs to Security Functional Requirements

The justification of the mapping between security functional requirements and security functionalities is given in Table 7.2.

Table 7.2 – Assignment of Security Functional Requirements to Security Functions

	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8
FMT_SMR.1							X	X
FMT_SMF.1			X		X		X	
FMT_MSA.2			X				X	X
FCS_RNG.1					X	X		
FIA_SOS.1							X	
FIA_SOS.2							X	
FMT_MTD.1/AUTH							X	

	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8
FIA_AFL.1/Lockout							X	
FIA_AFL.1/Recover							X	
FIA_AFL.1/PINFAIL							X	
FIA_AFL.1/PINPASS							X	
FIA_UID.1							X	
FIA_UAU.1							X	
FIA_UAU.**5**							X	X
FIA_UAU.6							X	
FIA**_USB.1**			X				X	
FMT_MSA.4/AUTH			X				X	
FDP_ACC.2/States			X					X
FDP_ACF.1/**States**			X					X
FMT_MSA.1/States			X					X
FMT_MSA.3/States			X					X
FDP_UIT.1/States								X
FPT_TST.1		X						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				X	X
FTP_ITC.1/AC			X				X	
FMT_MOF.1/**AC**			X					
FCS_CKM.1/**PK**					X			
FCS_CKM.1/ECC					X			
FCS_CKM.1/RSA					X			
FCS_CKM.1/**SYMM**					X			
FCS_CKM.4					X			
FCS_COP.1/AES	X							
FCS_COP.1/**SHA**	X							
FCS_COP.1/HMAC	X							
FCS_COP.1/RSAED	X							
FCS_COP.1/RSASign	X							
FCS**_COP.1/ECDSA**	X							
FCS_COP.1/ECDEC	X							
FDP_ACC.1/NVM			X					
FDP_ACF.1/**NVM**			X					
FMT**_MSA.1/NVM**			X					
FMT_MSA.3/**NVM**			X					

	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8
FMT**_MSA.4/NVM**			X					
FMT_MTD.1/NVM			X					
FDP_ITC.1/**NVM**			X					
FDP_ETC.1/**NVM**			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/Cre			X				X	
FDP_ACF.1/**Cre**			X				X	
FMT_MSA.1/**Cre**			X				X	
FMT_MSA.3/Cre			X				X	
FCO_NRO.1/**Cre**							X	
FDP_ACC.1/M&R			X				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_RIP.1							X	
FPT_FLS.1/**FS**		X						X
FPT_FLS.1/SD				X				
FPT_PHP.3				X				
FDP_ITT.1				X				
FPT_ITT.1				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					

8 Rationale

This section provides the evidence that supports the claims that the ST is a complete and cohesive set of objectives and requirements, and that the TOE summary specification addresses the requirements.

8.1 Rationale for the Security Objectives

The security problem definition of this security target is consistent with the statement of the security problem definition in the PP.

The rationale given in the PP ([PP], §5.3) remains fully valid for the Security Target. The PP has been augmented with three Security Objectives that were added to support the ANSSI [JIL_SCRL] requirements. The three Security Objectives do not interfere with PP conformance and they were added directly to counter the relevant threats as follows:

- Security Objective O.Secure_Load_ACode directly counters Threat T.unauthorized_Load
- Security Objective O.Secure_AC_Activation directly counters Threat T.Bad_Activation
- Security Objective O.TOE_Identification directly counters Threat T.TOE_Identification_Forgery

In addition, the three Security Objectives enforce the Organisational Security Policy OSP.FieldUpgrade.

Table 8.1 provides an overview of the mapping between the security objectives for the TOE and the threats countered by the objectives, the Organisational Security Policies they enforce and the Assumptions they address.

Table 8.1 – Security Objectives Rationale

	O.Context_Management	O.secure_Load_ACode	O.Secure_AC_Activation	O.TOE_Identification	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O._FieldUpgradeControl	OE.Configuration	OE.Locality	OE.Credential	OE.Measurement	OE.FieldUpgradeInfo
T.Compromise						X				X						X		X										
T.Bypass																		X	X										
T.Export							X											X						X					
T.Hack_Crypto					X																								
T.Hack_Physical						X																	X						
T.Imperson										X	X	X	X						X						X				
T.Import											X																		
T.Insecure_State								X	X									X						X					
T.Intercept							X				X											X							
T.Malfunction								X												X									
T.Modify						X				X		X							X										
T.Object_Attr_Change																		X											
T.Replay																				X									
T.Repudiate_Transact															X														
T.Residual_Info																X													
T.Leak																							X						
T.unauthorized_Load		X																											
T.Bad_Activation			X																										
T.TOE_Identification_Forgery				X																									
OSP.Context_Management	X																												
OSP.Policy_Authorisation						X												X											
OSP.Locality													X												X				
OSP.RT_Measurement														X													X		
OSP.RT_Reporting																	X									X			
OSP.RT_Storage					X	X	X			X	X																		
OSP.FieldUpgrade		X	X	X																			X						X
A.Configuration																								X					

8.2 Rationale for Security Requirements

The security requirements rationale for sufficiency, dependency and assurance is described in the Protection Profile [PP], §8.3.

The augmentation ALC_DVS.2 satisfies the PP constraints since it is higher than ALC_DVS.1.

8.2.1 Sufficiency of SFR

The sufficiency of the SFR is described in the PP section 8.3.1. The mapping demonstrates that each security objective for the TOE is covered by at least one SFR and that each SFR addresses at least one security objective of the TOE.

The additional security objective, O.Secure_Load_Acode, requires that the loader of the initial TOE will check for evidence of authenticity and integrity of the loader Additional Code and that during the Load Phase of an Additional Code, the TOE will remain secure. This objective is addressed by the following SFRs:

- FMT_MSA.2 requires that the TSF shall ensure that only secure values are accepted for the TPM_FieldUpgrade
- FDP_UIT.1/States requires that the TSF shall enforce an SFP to provide and use integrity protection capabilities for firmware update data on reception of that data.
- FDP_UCT.1/AC requires that the TSF shall enforce an SFP to use confidentiality protection capabilities for firmware update data on reception of that data to avoid additional code disclosure.

The additional security objective, O.Secure_AC_Activation, requires that Activation of the Additional Code and update of the identification data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the final TOE will be completed before activation. If the Atomic Activation is successful, then the resulting product is the final TOE; otherwise (in case of interruption or an incident that prevents the forming of the final TOE), the initial TOE will remain in its initial state or fail secure. This objective is addressed by the following SFRs:

- FPT_FLS.1/FS requires that the TSF shall preserve a secure state during a failure of the field upgrade process
- FDP_ACF.1/States Modes Security attribute-based access control defines rules to enforce a policy regarding the TOE states, including the state transition regarding the Field Upgrade mode state. It enforces atomicity by switching the state only if the complete upgrade has been processed.

The additional security objective, O.TOE_Identification, requires that 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 this data. After atomic activation of the additional code, the identification data of the final TOE allows identification of the initial TOE and additional code. The user must be able to uniquely identify the initial TOE and additional code, which are embedded in the final TOE. This objective is addressed by the following SFR:

- FDP_UIT.1/States requires that the TSF shall enforce an SFP to provide and use integrity protection capabilities for firmware update data on reception of that data.

Table 8.2 – Security Requirements Rationale Including [JIL_SCRL]

	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl	O.Secure_Load_ACode	O.Secure_AC_Activation	O.TOE_Identification
FMT_SMR.1																x					x			
FMT_SMF.1															x									
FMT_MSA.2															x							x		
FCS_RNG.1		x																						
FPT_STM.1												x												
FIA_SOS.2							x																	
FMT_MTD.1/AUTH							x																	
FIA_AFL.1/Lockout							x											x						
FIA_AFL.1/Recover							x											x						
FIA_AFL.1/PINFAIL							x											x						
FIA_AFL.1/PINPASS							x											x						
FIA_UID.1							x		x															
FIA_UAU.1							x		x															
FIA_UAU.5							x			x	x								x		x			
FIA_UAU.6							x											x						
FIA_USB.1			x				x			x						x								
FMT_MSA.4/AUTH			x				x								x									
FDP_ACC.2/States			x																		x			
FDP_ACF.1/States			x																		x		x	
FMT_MSA.1/States			x												x						x			
FMT_MSA.3/States			x												x						x			
FDP_UIT.1/States								x													x	x		x
FPT_TST.1					x	x											x							
FDP_ACC.1/AC			x							x														
FDP_ACF.1/AC			x							x														
FMT_MSA.1/AC			x							x					x									
FMT_MSA.3/AC			x							x					x									

	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl	O.Secure_Load_ACode	O.Secure_AC_Activation	O.TOE_Identification
FDP_UCT.1/AC																			x			x		
FTP_ITC.1/AC																			x					
FMT_MOF.1/AC			x																					
FCS_CKM.1/PK		x																						
FCS_CKM.1/ECC		x		x			x																	
FCS_CKM.1/RSA		x																						
FCS_CKM.1/SYMM		x																						
FCS_CKM.4		x																						
FCS_COP.1/AES	x	x		x				x											x		x			
FCS_COP.1/SHA											x	x							x		x			
FCS_COP.1/HMAC	x	x		x			x	x											x		x			
FCS_COP.1/RSAED				x				x											x		x			
FCS_COP.1/RSASign												x		x							x			
FCS_COP.1/ECDSA												x		x										
FCS_COP.1/ECDEC				x			x																	
FDP_ACC.1/NVM			x																					
FDP_ACF.1/NVM			x																					
FMT_MSA.1/NVM			x												x									
FMT_MSA.3/NVM			x												x									
FMT_MSA.4/NVM			x												x									
FMT_MTD.1/NVM			x												x									
FDP_ITC.1/NVM								x																
FDP_ETC.1/NVM				x																				
FDP_ACC.1/ExIm			x	x				x																
FDP_ACF.1/ExIm			x	x				x																
FMT_MSA.1/ExIm			x	x				x							x									
FMT_MSA.3/ExIm			x	x				x							x									
FDP_ETC.2/ExIm	x			x																				
FDP_ITC.2/ExIm	x							x																
FDP_UCT.1/ExIm	x			x				x																
FDP_UIT.1/ExIm	x			x				x				x												

	O.Context_Management	O.Crypto_Key_Man	O.DAC	O.Export	O.Fail_Secure	O.General_Integ_Checks	O.I&A	O.Import	O.Limit_Actions_Auth	O.Locality	O.Record_Measurement	O.MessageNR	O.No_Residual_Info	O.Reporting	O.Security_Attr_Mgt	O.Security_Roles	O.Self_Test	O.Single_Auth	O.Sessions	O.Tamper_Resistance	O.FieldUpgradeControl	O.Secure_Load_ACode	O.Secure_AC_Activation	O.TOE_Identification
FDP_ACC.1/Cre												x												
FDP_ACF.1/Cre												x												
FMT_MSA.1/Cre												x			x									
FMT_MSA.3/Cre												x			x									
FCO_NRO.1/Cre						x						x		x										
FDP_ACC.1/M&R			x								x													
FDP_ACF.1/M&R			x								x													
FMT_MSA.1/M&R			x								x			x	x									
FMT_MSA.3/M&R			x								x			x	x									
FCO_NRO.1/M&R												x		x										
FDP_RIP.1													x											
FPT_FLS.1/FS					x												x						x	
FPT_FLS.1/SD					x												x							
FPT_PHP.3																				x				
FDP_ITT.1																				x				
FPT_ITT.1																				x				
FDP_SDI.1				x		x		x																
FDP_ACC.1/Hier			x																					
FDP_ACF.1/Hier			x																					
FMT_MSA.1/Hier			x												x									
FMT_MSA.3/Hier			x												x									
FMT_MSA.4/Hier			x												x									

8.2.2 SFR Dependency Rationale

The dependency rationale described in the Protection Profile section 8.3.2 demonstrates that the dependencies of the SFR are fulfilled or provides an explanation in case those dependencies are not fulfilled. No SFR has been added; therefore, the dependency rationale is still valid.

9 Appendix 1

The TOE implements all TPM2.0 commands marked as “Mandatory” in [PTP] specification. In addition, it implements the following optional commands:

- TPM2_ChangePPS
- TPM2_ChangeEPS
- TPM2_ECC_Ephemeral
- TPM2_GetCommandAuditDigest
- TPM2_GetTime
- TPM2_HMAC
- TPM2_NV_Certify
- TPM2_NV_GlobalWriteLock
- TPM2_PolicyTicket
- TPM2_PolicyPhysicalPresence
- TPM2_PP_Commands
- TPM2_SetCommandCodeAuditStatus
- TPM2_Rewrap
- TPM2_ZGen_2Phase
- TPM2_ACT_SetTimeout

10 Appendix 2

10.1 References

Nuvoton TPM

[PRG] *NPCT7xx TPM2.0 Programmer's Guide, Revision 1.15, July 4, 2025 ("Programmer's Guide")*

[Datasheet] *NPCT7xx Trusted Platform Module Family 2.0, Revision 1.34, July 4, 2025 ("Datasheet")*

[ERT] *NPCT7xx User Product Information, Revision 2.19, July 4, 2025 ("Errata")*

[AGD] *NPCT7xx Guidance Document, Common Criteria AGD Component, Revision 2.3, July 4, 2025*

Common Criteria

[CC] Common Criteria for Information Technology Security Evaluation, version 3.1, revision 5, April 2017
Part 1: Introduction and general model, CCMB-2017-04-001,
Part 2: Security functional requirements, CCMB-2017-04-002,
Part 3: Security Assurance Requirements, CCMB-2017-04-003

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, version 3.1, revision 5, April 2017, CCMB-2017-04_004

[AIS31] A proposal for: Functionality classes and evaluation methodology for true (physical) random number generators, Version 3.1, 25.09.2001

[JIL_SCRL] Security requirements for post-delivery code loading Draft Version 1.0, February 2016, ANSSI

Protection Profile

[PP] Trusted Computing Group Protection Profile PC Client Specific Trusted Platform Module, TPM Library specification Family 2.0; Level 0 Revision 1.59, 29 September 2021, Version 1.3.

TCG

[PTP] TCG PC Client Specific Platform TPM Profile (PTP) Specification, Family 2.0, Version 01.04 Revision 37 (February 3, 2020)

[TIS] TCG PC Client Specific TPM Interface Specification (TIS), Version 1.3 (21 March 2013)

[TCG-1] TPM Main Part 1 Architecture, Specification version 2.0, revision 1.59 (November 8, 2019)

[TCG-2] TPM Main Part 2 TPM Structures, Specification version 2.0, revision 1.59 (November 8, 2019)

[TCG-3] TPM Main Part 3 Commands, Specification version 2.0, revision 1.59 (November 8, 2019)

[TCG-4] TPM Main Part 4 Supporting Routines, Specification version 2.0, revision 1.59 (November 8, 2019)

https://www.trustedcomputinggroup.org/home

Literature

[P1363] IEEE P1363-2000, Standard Specifications for Public Key Cryptography, Institute of Electrical and Electronics Engineers, Inc. (note reaffirmation PAR is actual running)

[HMAC] RFC 2104: HMAC: Keyed-Hashing for Message Authentication, http://www.ietf.org/rfc/rfc2104.txt

[FIPS140-2] Federal Information Processing Standards Publication 140-2

[SP800-90A] NIST Special Publication 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators; Revision 1, June 2015

[SP800-90B] NIST Special Publication 800-90B: Recommendation for the Entropy Sources Used for Random Bit Generation; January 2018

[SP800-90C] NIST Special Publication 800-90C: Recommendation for Random Bit Generator (RBG) Constructions; second draft, April 2016

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

[FIPS186-4] FIPS PUB 186-4 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION Digital Signature Standard (DSS)

[SP800-38A] NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation. December 2001

[SP800-56A] NIST Special Publication 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptology. March 2007

[SP800-108] NIST Special Publication 800-108: Recommendation for Key Derivation Using Pseudorandom Functions. October 2009

[FIPS198-1] FIPS 198-1 Federal Information Processing Standards Publication, The Keyed-Hash Message Authentication Code (HMAC), July 2008

[ISO10116:2006] ISO/IEC 10116:2006, *Information technology — Security techniques — Modes of operation for an n*-bit *block cipher*

[ISO14888-3] ISO/IEC 14888-3, Information technology -- Security techniques -- Digital signature with appendix -- Part 3: Discrete logarithm based mechanisms

[PKCS#1v2.1] IETF RFC 3447, PKCS #1 v2.1: RSA Cryptography Standard, RSA Laboratories, June 14, 2002

[ISO9797-2] ISO/IEC 9797-2, Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 2: Mechanisms using a dedicated hash-function

[ISO18033-3] ISO/IEC 18033-3, Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers

[ISO15946-1] ISO/IEC 15946-1, Information technology — Security techniques — Cryptographic techniques based on elliptic curves — Part 1: General

[FIPS 197] Federal Information Processing Standards Publication 197: Specification for the ADVANCED ENCRYPTION STANDARD (AES), November 26, 2001

10.2 Acronyms and Glossary

Acronyms

CC	Common Criteria
EAL	Evaluation Assurance Level
IT	Information Technology
NTC	Nuvoton Technology Corporation
PP	Protection Profile
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy

Glossary

AES:	Symmetric key encryption defined by NIST as FIPS 197.
Blob:	Opaque data of fixed or variable size. The meaning and interpretation of the data is outside the scope and context of the Subsystem.
Challenger:	An entity that requests and has the ability to interpret integrity metrics from a Subsystem.
Conformance Credential:	A credential that states the conformance to the TCG specification of: the TPM; the method of incorporation of the TPM into the platform; the RTM; and the method of incorporation of the RTM into the platform.
Denial-of-service attack:	An attack on a system (or subsystem) that has no effect on information except to prevent its use.
Endorsement Credential:	A credential containing a public key (the endorsement public key) that was generated by a genuine TPM.
Endorsement Key:	A term used ambiguously, depending on context, to mean a pair of keys, or the public key of that pair, or the private key of that pair; an asymmetric key pair generated by or inserted in a TPM that is used as proof that a TPM is a genuine TPM; the public endorsement key (PUBEK); the private endorsement key (PRIVEK).
Identity Credential:	A credential issued by a Privacy CA that provides an identity for the TPM.
Integrity metric(s):	Values that are the results of measurements on the integrity of the platform.
Man-in-the-middle attack:	An attack by an entity intercepting communications between two others without their knowledge and by intercepting that communication is able to obtain or modify the information between them.
Migratable:	A key that may be transported outside the specific TPM.
Nonce:	A nonce is a random value that provides protection from replay and other attacks. Many of the commands and protocols in the specification require a nonce.
Non-Migratable:	A key that cannot be transported outside a specific TPM; a key that is (statistically) unique to a particular TPM.
Owner:	The entity that owns the platform in which a TPM is installed. Since there is, by definition, a one-to-one relationship between the TPM and the platform, the Owner is also the Owner of the TPM. The Owner of the platform is not necessarily the “user” of the platform (e.g., in a corporation, the Owner of the platform might be the IT department while the user is an employee.) The Owner has administration rights over the TPM.
PKI Identity Protocol:	The protocol used to insert anonymous identities into the TPM.
Platform Credential:	A credential that states that a specific platform contains a genuine TCG Subsystem.
Privacy CA:	An entity that issues an Identity Credential for a TPM based on trust in the entities that vouch for the TPM via the Endorsement Credential, the Conformance Credential, and the Platform Credential.

Private Endorsement Key (PRIVEK): The private key of the key pair that proves that a TPM is a genuine TPM. The PRIVEK is (statistically) unique to only one TPM.

Public Endorsement Key (PUBEK): A public key that proves that a TPM is a genuine TPM. The PUBEK is (statistically) unique to only one TPM.

Random Number Generator (RNG): A pseudo-random number generator that must be initialized with unpredictable data and provides, “random” numbers on demand.

Root of Trust for Measurement (RTM): The point from which all trust in the measurement process is predicated.

Root of Trust for Reporting (RTR): The point from which all trust in reporting of measured information is predicated.

Root of Trust for Storing (RTS): The point from which all trust in Protected Storage is predicated.

RSA: An (asymmetric) encryption method using two keys: a private key and a public key. Reference: http://www.rsa.com.

SHA-1: A NIST defined hashing algorithm producing a 160-bit result from an arbitrary sized source as specified in FIPS 180-1.

Storage Root Key (SRK): The root key of a hierarchy of keys associated with a TPM; generated within a TPM; a non-migratable key.

Subsystem: The combination of the TSS and the TPM.

Support Services (TSS): Services to support the TPM but that do not need the protection of the TPM. The same as Trusted Platform Support Services.

TCG-protected capability: A function that is protected within the TPM, and has access to TPM secrets.

TPM Identity: One of the anonymous PKI identities belonging to a TPM; a TPM may have multiple identities.

Trusted Platform Agent (TPA): Trusted Platform Agent; the component within the platform that reports integrity metrics, logs, Validation Data, etc. to a Challenger; outside the scope of this specification.

Trusted Platform Measurement Store (TPMS): Storage locations within the Subsystem that contain unprotected logs of measurement process.

Trusted Platform Module (TPM): The set of functions and data, common to all platform types, that must be trustworthy if the Subsystem is to be trustworthy; a logical definition in terms of protected capabilities and shielded locations.

Trusted Platform Support Services (TSS): The set of functions and data, common to all platform types, that are not required to be trustworthy (and therefore do not need to be part of the TPM).

User: An entity that uses the platform in that a TPM is installed. The only rights that a User has over a TPM are the rights given to the User by the Owner. These rights are expressed in the form of authentication data, given by the Owner to the User, that permits access to entities protected by the TPM. The User of the platform is not necessarily the “owner” of the platform (e.g., in a corporation, the owner of the platform might be the IT department while the User is an employee). There can be multiple Users.

Validation Credential: A credential that states values of measurements that should be obtained when measuring a particular part of the platform when the part is functioning as expected.

Validation Data: Data inside a Validation Credential; the values that the integrity measurements should produce when the part of a platform described by the Validation Credential is working correctly.

Validation Entity: An entity that issues a Validation Certificate for a component; the manufacturer of that component; an agent of the manufacturer of that component.

Nuvoton provides comprehensive service and support.
For product information and technical assistance, contact the nearest Nuvoton center.

Important Notice

Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Furthermore, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.

Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.

CONTACT INFORMATION

For Nuvoton Sales Offices in your region, visit us at:

https://www.nuvoton.com/buy/worldwide-sales-offices/

For Cloud Computing Product Line information, contact:

CloudComputing@nuvoton.com

Please note that all data and specifications are subject to change without notice.
All trademarks of products and companies mentioned in this document belong to their respective owners.

www.nuvoton.com