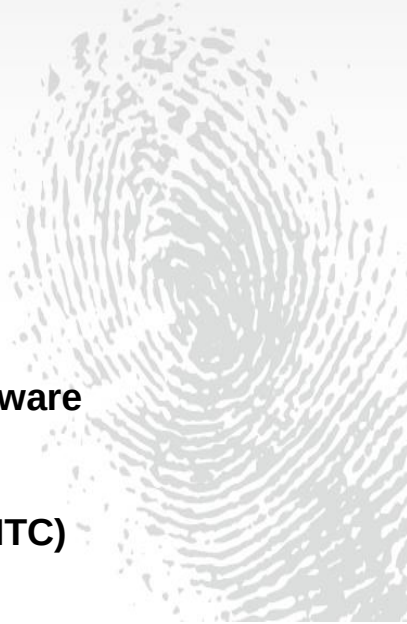
NUVOTON TECHNOLOGY CORPORATION

Security Target



Version: **1.9**

Date: **October 26, 2016**

Author: **Yossi Talmi**

Product: **TPM1.2 (Hardware FB5C85D/E, Firmware 5.81.2.1)**

Manufacturer: **Nuvoton Technology Corporation (NTC)**

Revision History

Version	Date	Description
1.0	February 2015	Updated Lite version of the ST
1.1	April 2015	Updated after ANSSI's feedback
1.2	April 2015	Added HW version FB5C85E
1.3	June 2015	Added a test plant
1.4	October 2015	Updated FW version to 5.81.1.0 and Part Numbers accordingly.
1.5	October 2015	Updated Part Numbers.
1.6	January 10, 2016	Updated Datasheet and AGD references; based on version 1.3 (**not including** the changes introduced in versions 1.4 and 1.5).
1.7	January 11, 2016	Updated Datasheet and AGD references; based on version 1.5 (**not including** the changes introduced in version 1.6).
1.8	February 08, 2016	Updated FW version to 5.81.2.1 and Part Numbers accordingly.
1.9	October 26, 2016	Added TowerJazz wafer fab.

Table of Contents

1 INTRODUCTION 5
1.1 Security Target (ST) and Target of Evaluation (TOE) Identification 5
1.2 TOE Global Overview 7
1.3 Organization of the Security Target 8
1.4 Common Criteria (CC) Conformance 9
2 TOE DESCRIPTION 10
2.1 TPM - General Remarks 10
2.1.1 Algorithms 13
2.1.2 Random Number Generator (RNG) 13
2.1.3 Key Generation 13
2.1.4 Self Tests 14
2.1.5 Identification and Authentication 14
2.1.6 Access Control 14
2.2 Security Attributes and Data 15
2.3 TOE Overview 16
2.3.1 TPM Modules 16
2.3.2 Hardware Interface 18
2.3.3 Software Interface 19
2.4 Guidance Documentation 19
2.5 TOE Life Cycle Description 19
3 CONFORMANCE CLAIMS 22
3.1 CC Conformance Claim 22
3.2 PP Claim 22
3.3 Package Claim 22
3.4 Conformance Claim Rationale 22
4 TOE SECURITY PROBLEM DEFINITION 23
4.1 Threats to Security 23
4.2 Organizational Security Policies 23
4.3 Secure Usage Assumptions 23
4.4 Security Problem Definition Rational 23
5 SECURITY OBJECTIVES 24
5.1 Security Objectives for the TOE 24
5.2 Security Objectives for the Operational Environment 24
5.3 Security Objectives Rationales 24
6 SECURITY REQUIREMENTS 26
6.1 Security Functional Requirements for the TOE 26
6.1.1 General SFR 26
6.1.2 Cryptographic Support 28
6.1.3 TPM Operational Modes 30
6.1.4 Identification, Authentication and Binding 33
6.1.5 Data Protection and Privacy 39
Delegation 39

Key Management 41
Measurement and Reporting 48
6.1.6 Data Import and Export 57
6.1.7 DAA 63
6.1.8 TSF Protection 65
6.2 Security Assurance Requirements for the TOE 66
7 TOE SUMMARY SPECIFICATION 67
7.1 TOE Security Features 67
7.1.1 SF1 – Cryptographic Operations 67
7.1.2 SF2 – Self Test 67
7.1.3 SF3 – Access Control 67
7.1.4 SF4 – Hacking and Physical Tampering Protection/Detection 68
7.1.5 SF5 – Key Management 68
7.1.6 SF6 – Random Number Generation 68
7.1.7 SF7 – Identification and Authentication 68
7.1.8 SF8 – Firmware Field Upgrade 69
7.1.9 Assignment of SFs to Security Functional Requirements 70
8 APPENDIX 72
8.1 TCG Specification Commands Implemented in the TOE 72
8.2 References 75
8.3 Acronyms 76
8.4 Glossary 77

1 Introduction

This section contains document management and overview information. The Security Target (ST) identification provides the labelling and descriptive information necessary to identify, catalogue, register, and cross reference an ST. The ST overview in Section 1.2 summarizes the ST in narrative form 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 title of this document is “Security Target of TPM1.2, revision 1.9”.

The Target of Evaluation (TOE), called “TPM1.2 with HW FB5C85D/E, FW 5.81.2.1”, is a Trusted Platform Module (**TPM**), that is, a TCG 1.2-compliant security processor with embedded firmware. In this document, the TOE is called **TPM1.2**.

Device Markings:

The device markings for the TPM1.2 with HW FB5C85D, FW 5.81.2.1 are defined as:

NPCT620AA2WX, NPCT620BA2WX, NPCT620CA2WX, NPCT620DA2WX,
NPCT620HA2WX, NPCT620IA2WX, NPCT620LA2WX, NPCT620MA2WX,
NPCT620NA2WX, NPCT620RA2WX, NPCT620SA2WX, NPCT620TA2WX,
NPCT620UA2WX, NPCT620VA2WX, NPCT622AA2WX, NPCT622BA2WX,
NPCT622CA2WX, NPCT622DA2WX,
NPCT622HA2WX, NPCT622IA2WX, NPCT622LA2WX, NPCT622MA2WX,
NPCT622NA2WX, NPCT622RA2WX, NPCT622SA2WX, NPCT622TA2WX,
NPCT622UA2WX, NPCT622VA2WX, NPCT622JA2WX,
NPCT620AA2YX, NPCT620BA2YX, NPCT620CA2YX, NPCT620DA2YX,
NPCT620HA2YX, NPCT620IA2YX, NPCT620LA2YX, NPCT620MA2YX,
NPCT620NA2YX, NPCT620RA2YX, NPCT620SA2YX, NPCT620TA2YX,
NPCT620UA2YX, NPCT620VA2YX, NPCT620JA2YX,
NPCT622AA2YX, NPCT622BA2YX, NPCT622CA2YX, NPCT622DA2YX,
NPCT622HA2YX, NPCT622IA2YX, NPCT622LA2YX, NPCT622MA2YX,
NPCT622NA2YX, NPCT622RA2YX, NPCT622SA2YX, NPCT622TA2YX,
NPCT622UA2YX, NPCT622VA2YX, NPCT622JA2YX,
NPCT650AA2WX, NPCT650BA2WX, NPCT650CA2WX, NPCT650DA2WX,
NPCT650HA2WX, NPCT650IA2WX, NPCT650LA2WX, NPCT650MA2WX,
NPCT650NA2WX, NPCT650RA2WX, NPCT650SA2WX, NPCT650TA2WX,
NPCT650UA2WX, NPCT650VA2WX, NPCT650JA2WX,

NPCT652AA2WX, NPCT652BA2WX, NPCT652CA2WX, NPCT652DA2WX,
NPCT652HA2WX, NPCT652IA2WX, NPCT652LA2WX, NPCT652MA2WX,
NPCT652NA2WX, NPCT652RA2WX, NPCT652SA2WX, NPCT652TA2WX,
NPCT652UA2WX, NPCT652VA2WX, NPCT652JA2WX, NPCT650LA0WX,
NPCT650AA2YX, NPCT650BA2YX, NPCT650CA2YX, NPCT650DA2YX,
NPCT650HA2YX, NPCT650IA2YX, NPCT650LA2YX, NPCT650MA2YX,
NPCT650NA2YX, NPCT650RA2YX, NPCT650SA2YX, NPCT650TA2YX,
NPCT650UA2YX, NPCT650VA2YX, NPCT650JA2YX,
NPCT652AA2YX, NPCT652BA2YX, NPCT652CA2YX, NPCT652DA2YX,
NPCT652HA2YX, NPCT652IA2YX, NPCT652LA2YX, NPCT652MA2YX,
NPCT652NA2YX, NPCT652RA2YX, NPCT652SA2YX, NPCT652TA2YX,
NPCT652UA2YX, NPCT652VA2YX, NPCT652JA2YX.

The device markings for the TPM1.2 with HW FB5C85E, FW 5.81.2.1 are defined as:

NPCT620AB2WX, NPCT620BB2WX, NPCT620CB2WX, NPCT620DB2WX,
NPCT620HB2WX, NPCT620IB2WX, NPCT620LB2WX, NPCT620MB2WX,
NPCT620NB2WX, NPCT620RB2WX, NPCT620SB2WX, NPCT620TB2WX,
NPCT620UB2WX, NPCT620VB2WX, NPCT620JB2WX,
NPCT622AB2WX, NPCT622BB2WX, NPCT622CB2WX, NPCT622DB2WX,
NPCT622HB2WX, NPCT622IB2WX, NPCT622LB2WX, NPCT622MB2WX,
NPCT622NB2WX, NPCT622RB2WX, NPCT622SB2WX, NPCT622TB2WX,
NPCT622UB2WX, NPCT622VB2WX, NPCT622JB2WX,
NPCT620AB2YX, NPCT620BB2YX, NPCT620CB2YX, NPCT620DB2YX,
NPCT620HB2YX, NPCT620IB2YX, NPCT620LB2YX, NPCT620MB2YX,
NPCT620NB2YX, NPCT620RB2YX, NPCT620SB2YX, NPCT620TB2YX,
NPCT620UB2YX, NPCT620VB2YX, NPCT620JB2YX,
NPCT622AB2YX, NPCT622BB2YX, NPCT622CB2YX, NPCT622DB2YX,
NPCT622HB2YX, NPCT622IB2YX, NPCT622LB2YX, NPCT622MB2YX,
NPCT622NB2YX, NPCT622RB2YX, NPCT622SB2YX, NPCT622TB2YX,
NPCT622UB2YX, NPCT622VB2YX, NPCT622JB2YX,
NPCT650AB2WX, NPCT650BB2WX, NPCT650CB2WX, NPCT650DB2WX,
NPCT650HB2WX, NPCT650IB2WX, NPCT650LB2WX, NPCT650MB2WX,
NPCT650NB2WX, NPCT650RB2WX, NPCT650SB2WX, NPCT650TB2WX,
NPCT650UB2WX, NPCT650VB2WX, NPCT650JB2WX,
NPCT652AB2WX, NPCT652BB2WX, NPCT652CB2WX, NPCT652DB2WX,
NPCT652HB2WX, NPCT652IB2WX, NPCT652LB2WX, NPCT652MB2WX,

NPCT652NB2WX, NPCT652RB2WX, NPCT652SB2WX, NPCT652TB2WX,
NPCT652UB2WX, NPCT652VB2WX, NPCT652JB2WX,
NPCT650AB2YX, NPCT650BB2YX, NPCT650CB2YX, NPCT650DB2YX,
NPCT650HB2YX, NPCT650IB2YX, NPCT650LB2YX, NPCT650MB2YX,
NPCT650NB2YX, NPCT650RB2YX, NPCT650SB2YX, NPCT650TB2YX,
NPCT650UB2YX, NPCT650VB2YX, NPCT650JB2YX,
NPCT652AB2YX, NPCT652BB2YX, NPCT652CB2YX, NPCT652DB2YX,
NPCT652HB2YX, NPCT652IB2YX, NPCT652LB2YX, NPCT652MB2YX,
NPCT652NB2YX, NPCT652RB2YX, NPCT652SB2YX, NPCT652TB2YX,
NPCT652UB2YX, NPCT652VB2YX, NPCT652JB2YX.

The Security Target is based on the following Trusted Computing Group (**TCG**) Protection Profile (**PP**): “TCG Protection Profile PC Client-Specific TPM – TPM family 1.2; level 2 revision 116 (certificate “BSI-CC-PP-0030-2008-MA-01, October 6, 2011”).

The PP and the ST are built with Common Criteria V3.1 Release 4.

1.2 TOE Global Overview

This security target describes the TOE (“TPM1.2”) and gives a short summary specification.

The TPM1.2 is a single electronic device Trusted Platform Module (TPM). The TPM1.2 implements the TPM Main Specification Family 1.2 Revision 116 and the TCG PC Client-Specific TPM Interface Specification (TIS) Version 1.3. In this document, the TCG specification is referred to as **TCG-x**. The TPM1.2 is designed to reduce system boot time and Trusted OS loading time. It provides a solution for PC security for a wide range of PC applications.

The TPM1.2 may interface with the host platform via Low Pin Count (LPC) interface, SPI interface or I2C interface. The TPM1.2 implements the LPC and SPI interfaces as defined in the TIS specification. The I2C interface is supported by TIS emulation over the I2C physical bus interface. The TPM1.2 is Microsoft® Windows® compliant and is supported by Linux kernel v2.6.18 and higher.

The following is a summary of the TPM1.2 main features:

- Single-chip TPM solution; no external parts required
- Two package options: TSSOP28, QFN32
- TCG compliance:
    - TIS version 1.3
    - TPM Main Specification Family 1.2 Revision 116

- EK certification support
- Field Upgrade - allows secure upgrading of the TPM1.2 firmware to meet future TCG specifications and take advantage of future security enhancements
- Up to five secure General-Purpose I/O (GPIO) pins
- 128-bit AES Counter (CTR) mode for transport sessions and authentication data protection
- NV storage size of 16K bytes
- Extended internal NVM life time
- Supports 32,000,000 writes For NV index of 32 bytes
- More than 20-year data retention
- Random Number Generator (RNG)
- Cryptographic hardware accelerators for AES, SHA-1 and RSA
- Host Interface
    - Supports both LPC and SPI with 64-byte data FIFO
    - 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 with up to 400 KHz clock operation

The TPM1.2 device works with a second module called the TCG PC Connection (**PCCON**), which may include the PC system BIOS and other software. The PCCON is not part of this evaluation.

1.3 Organization of the Security Target

The sections of the ST are:

- TOE Description (Section 2)
- Conformance Claims (Section 3)
- TOE Security Problem Definition (Section 4)
- Security Objectives (Section 5)
- Security Requirements (Section 6)
- TOE Summary Specification (Section 7)

Section 2, the TOE Description, provides general information about a Trusted Platform Module and the TOE itself, serves as an aid to understanding the TOE security requirements, and provides context for the ST evaluation.

Conformance Claims are given in **Section 3** in the form of claims versus the Common Criteria and the Protection Profile used for this Security Target.

The TOE Security Problem Definition in **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
- Organizational 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 counters identified threats and how it will cover identified organizational security policies and assumptions. Each security objective is categorized as being for either the TOE or the 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 in **Chapter 7** summarizes the security features of this specific TOE, the TPM1.2.

Section 8 (Appendix) provides a table of TOE commands, a list of acronyms, a glossary of terms and a list of references.

1.4 Common Criteria (CC) Conformance

This ST was built with Common Criteria (**CC**) Version 3.1 Revision 4 (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 ST is conformant with the protection profile TCG TPMPP version 1.2 [PP]. This means that the Security Target is conformant with Common Criteria Version 3.1 Revision 4, part 2 “extended” and part 3 [CC].

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

2 TOE Description

The TOE description helps to understand the specific security environment and the security policy. Within this context, the assets, threats, security objectives and security functional requirements can be used. After some general remarks about the Trusted Platform Module in Chapters 2.1 and 2.2, Chapter 2.3 presents a more detailed description of the TOE than 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 subsystems that assure authenticity, integrity and confidentiality in e-commerce and Internet communications within a Trusted Computing Platform, as defined in the TCG-x. The TPM is a complete solution implementing the TCG-x. The Trusted Computing Group is an industry group originally founded in 1999 as “TCPA” by COMPAQ, HP, IBM, Intel, and Microsoft.

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. That 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 in order to protect secrets from disclosure and protect methods from subversion.

Each 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 in order for a 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 the subsystem. These are called the “Trusted Platform Support Set” (**TSS**).

The Trusted Set of capabilities can be partitioned into three functions: 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 reports those measurement results reliably.

The RTS provides methods to minimize the amount of trusted storage that is required.

The RTM and RTR cooperate to permit an entity to believe measurements that describe the current computing environment of the platform. An entity can assess those measurement results and compare them with the 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 subsystem is a trusted subsystem that is an integral part of a computing platform. The evaluated components that make up the 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 consists 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; therefore, the TPM is common to all types of trusted platforms. The TPM uses cryptographic techniques to report its identity and the measurement results reliably. Since this raises privacy issues, a subsystem includes features that provide privacy controls to the Owner.

The PCCON provides the connection to between the computing platform and the RTM. The

TSS is a set of functions and data that are common to all types of platforms that are not required to be trustworthy.

The TPM is a collection of hardware, firmware and/or software that among others support the following security features:

- Algorithms: RSA, SHA-1, HMAC, AES, MGF1
- Random number generation
- Key generation
- Self tests

The TPM can 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 the keys outside the TPM.

The functionality used to provide secure storage is:

- **Seal** and **Unseal** - These perform RSA encrypt and decrypt, respectively, on data that is externally generated. The sealing operation encrypts not only the data but also the platform configuration values that are stored in the Platform Configuration Registers (**PCR**s), in the TPM, and in tpmProof, which is a unique identifier for that TPM. To unseal the data, three conditions must exist: 1) the appropriate key must be available for unseal, 2) the TPM PCRs must contain the same values that existed at the time of the seal operation, and 3) the value of tpmProof must be the same as that encrypted during the seal operation. By requiring the PCR values to be duplicated at unseal and the tpmProof value to be checked, the seal operation allows software to state explicitly the future "trusted" configuration that the platform must be have for the decrypted key to be used and for decrypt to only occur on the specified TPM.
- **Unbind** - RSA decrypts a blob created outside the TPM. This blob was encrypted using a public key where the associated private key is stored in the TPM.

A number of key types are defined within the TPM. Keys may be migratable or non-migratable. A migratable key is a key that may be transported outside a specific TPM. A non-migratable key is a key that cannot be transported outside a specific TPM. Key types include:

- The Storage Root key (SRK) - The root key of a hierarchy of keys associated with a TPM. It is generated within a TPM and is a non-migratable key. Each TPM contains a SRK, generated by the TPM at the request of the Owner. Under that SRK are two trees: one dealing with migratable data and the other dealing with non-migratable data
- Signing Keys - Each must be a leaf of the Storage Root Key hierarchy. The private key of the key pair is used for signing operations only.
- Storage keys - Are used for RSA encrypt and RSA decrypt of other keys in the Protected Storage hierarchy only.
- Identity Keys - Are used for operations that require a TPM identity only.
- Binding Keys - Are used for TPM_Unbind operations only. A bind operation (performed outside the TPM) associates identification and authentication data with a particular data set, and the entire data blob is encrypted outside the TPM using a binding key, which is an RSA key. The TPM_Unbind operation uses a private key stored in the TPM to decrypt the blob so that the data (often a key pair) stored in the blob may be used.
- The Endorsement key pair - An asymmetric key pair inserted in a TPM that is used as proof that a TPM is a genuine TPM. This key is non-revocable and cannot be substituted by a new one[1].

[1] The TOE does not implement the optional function "revoke of trust" documented in the claimed PP (see [PP], §8).

Each TPM is identified and validated by its endorsement key. A TPM has only one endorsement key pair. The endorsement key is bound transitively to the platform via the TPM as follows:

- An endorsement key is bound to one and only one TPM (i.e., there is a one-to-one correspondence between an endorsement key and a TPM.)
- A TPM is bound to one and only one platform, (i.e., there is a one-to-one correspondence between a TPM and a platform.)
- Therefore, an endorsement key is bound to a platform, (i.e., there is a one-to-one correspondence between an endorsement key and a platform.

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

2.1.1 Algorithms

The TPM supports the RSA algorithm and must use the RSA algorithm for encryption and digital signatures. The TPM supports RSA key sizes of 512, 1024, and 2048 bits. The RSA public exponent must be e, where e = 2^{16}+1. All TPM storage keys are of strength equivalent to a 2048-bit RSA key. The TPM does not load a storage key whose strength is less than that of a 2048-bit RSA key. All TPM identity keys are of strength equivalent to a 2048-bit RSA key or greater.

The TPM supports the Secure Hash Algorithm (SHA-1) hash algorithm as defined by United States Federal Information Processing Standard 180-1. The output of SHA-1 is 160 bits; all areas that expect a hash value are required to support the full 160 bits. A SHA-1 digest is used in the early stages of a boot process before more sophisticated computing resources are available. Secure Hash is also used in the process of preparing data for signature or signature verification.

The TPM also supports symmetric 128-bit AES algorithm and MGF1 algorithms.

2.1.2 Random Number Generator (RNG)

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 TPM generates the HMAC key by taking the next *n* bits from the TPM RNG.

The creation of all nonce values uses the next *n* bits from the TPM RNG.

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 pass or fail. Self-tests include checks of the following:

- RNG functionality, as defined by United Stated Federal Information Processing Standard 140-2.
- Reading and extending the integrity registers. The self-test for the integrity registers will leave the integrity registers in a known state.
- Endorsement key pair integrity, if the key pair exists. This test verifies that the Endorsement key pair can sign and verify a known value. It also tests the RSA sign and verify engine. If the Endorsement key has not yet been generated, the TPM action is manufacturer specific.
- Integrity of the protected capabilities of the TPM. This consists of checks that ensure that the TPM FW has not changed.
- Cryptographic engines test. The SHA1 and AES modules are checked by performing the corresponding action on a known value and comparing the known result.

On failure of at least one of the above specified tests, the TPM enters Limited Operation Mode, in which the security-affecting commands are disabled.

On a fatal error such as FW integrity failure, the TPM enters Permanent Halt Mode, in which it shuts down all of its interfaces and does not respond.

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-x calls the identification and authentication process and this data authorization.

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

The identification and authentication protocols use a random nonce.

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 but allowing access to other data and operations based on the value of the TCG_AUTH_DATA_USAGE flag TCG_KEY_FLAGS and the TCG_KEY_USAGE flag.

The TCG_AUTH_DATA_USAGE flag defines access as either “owner” or “world”. **Owner** must be authenticated with a shared secret, as described in Section 2.1.5, above. **World** means that the key can be used by anyone, without authentication.

The TCG_KEY_FLAGS define whether a key is migratable or non-migratable and whether the key is stored in volatile storage and must be unloaded at TPM start-up.

The TCG_KEY_USAGE flag identifies the key type, as defined in Section 2.1 above. Depending on the key type, certain operations may or may not be allowed using the particular key, as described above.

On appropriate identification and authentication associated with the keys, users can use the key for the purposes permitted by the TCG_KEY_USAGE flag.

2.2 Security Attributes and Data

All data, including user key pairs, user data, and TSF data, have associated security attributes stored as flags in the TPM or associated with the data in an encrypted blob. The following security attributes are defined:

- Migration attribute - Determines if the data (or key pair) can migrate from one TPM to another. This security attribute is stored in TCG_KEY_FLAGS.
- TCG_AUTHDATA_USAGE flag - Is used to define whether the data can be access only by the owner or by the world.
- Attribute key type, stored in TCG_ KEY_USAGE - Indicates if the data is a key or key pair and the type of key (e.g., storage, binding, etc., as defined in Section 2.1, above).
- Volatility attribute - Defines whether the data must be stored in volatile or non-volatile storage and whether it is cleared at TPM start-up. This security attribute is stored in TCG_KEY_FLAGS.

Within the TPM, for the purposes of Common Criteria evaluation, TSF data is defined as containing all of the following:

- The Endorsement Key Pair
- The Storage Root Key (SRK)
- TPMProof, i.e., the random number (nonce) that each TPM maintains to validate that the data originated at this TPM
- PCR values
- TPM owner identification and authentication data
- Entity owner identification and authentication data
- Migration authorization data, which is used in creating migratable key blobs
- Security attributes as defined above

User data is defined as all user keys and other data that may be passed to the TPM for signature, decryption, etc.

2.3 TOE Overview

The Target of Evaluation (TOE)—the TPM1.2 with HW FB5C85D/E and FW 5.81.1.0—is a Trusted Platform Module that provides TCG-compliant security functionality.

The TPM1.2 is a single-chip device comprising a Trusted Platform Module (TPM) for PC security based on the TCG standard.

The TPM1.2 device, developed by Nuvoton Technology Corporation, complies with TPM Main Specification Family 1.2 Revision 116 and TCG PC Client-Specific TPM Interface Specification (TIS) Version 1.3.

The TPM1.2 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 such as credit card numbers, passwords and keys.

2.3.1 TPM Modules

- Processing Unit Module
- Memories, including ROM RAM and NVM
- RSA Accelerator Module
- SHA-1 Accelerator Module
- CIPHER (AES) Accelerator Module
- RNG Module (Random Number Generator)
- Timers
- On Chip Clock Generator
- GPIO Ports Module (General-Purpose Input/Output)
- Host interfaces:
    - Supports LPC, SPI and I2C Slave Bus Interface
    - Five localities
    - Host interface voltage level options: 1.8 Volts or 3.3 Volts

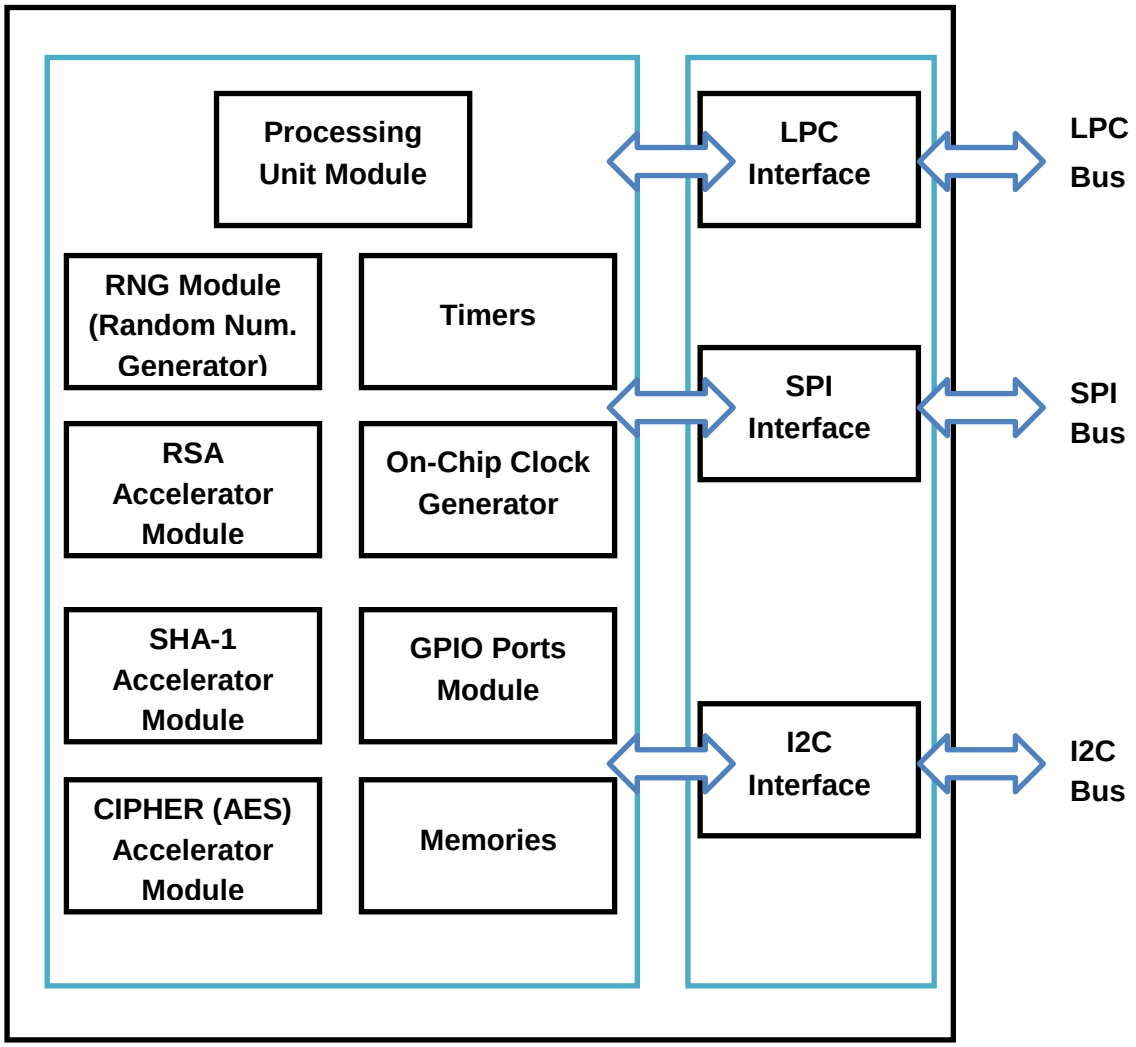


Figure 1 – TPM1.2 Block Diagram

The **firmware part of the TOE** provides an API set that matches the TCG-x, in which 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 firmware implements NTC proprietary commands and additional non-TPM related functionality.

The TOE also supplies an area for secure storage. The shielded locations are accessible only for protected functions. The following data is stored in shielded locations:

- The private keys of the Endorsement-Key and Store-Root-Key remain in shielded locations and never leave the TPM
- Platform secrets
- TPM Owner secret
- Imported protected data (e.g., keys) is stored only in a shielded location
- Platform Configuration Registers (**PCR**s) and Data Integrity Registers (**DIR**s)

The TOE TPM module supplies the following cryptographic services for the user:

- Hardware True-Random generator, to generate random numbers also for entities outside of the TPM. Keys generation: generation of 2048, 1024 and 512-bit asymmetric key pairs
- Performing RSA encryption and decryption on externally generated data, using keys stored in the shielded location
- Digital signature, using the RSA algorithm
- Integrity metrics
- SHA1 hashing
- AES encryption/decryption

The TPM1.2 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 TPM1.2 asymmetric encryption functionality. Regarding a network scenario, the client PCs equipped with a TPM1.2 are able to report their platform status to the server so that the network administration is aware of their trustworthiness. In conclusion, the TPM1.2 acts as a service provider to a system helps to make transactions more secure and trustworthy.

2.3.2 Hardware Interface

The physical interface as well as the electrical interface of the TOE is comprised of the pins of the device. The electrical interface of the TOE to the external environment is comprised of the active pins of the device. The device has 28 pins, which include power and ground, LPC bus pins, SMBus (I2C), SPI interface, a Physical Presence pin and general-purpose I/Os. TPM commands and responses to them may be transferred between the TPM and the host via LPC, SPI or I2C bus.

2.3.3 Software Interface

The interface to the firmware is via 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: device side (the TPM) and host side. The host side refers, globally, to any process in the host computer that communicates with the TPM (e.g., the BIOS or the OS resident drivers).

2.4 Guidance Documentation

The guidance documentation consists of:

- The device datasheet (Datasheet) - Details the specific vendor software commands and driver protocols
- The AGD document used during this evaluation (AGD) - Details all the aspects relevant for the user and administrator of the TOE
- The TCG main specification (**TCG-x**) - Details all the standard TCG commands and protocols for device initialization, starting from endorsement key-pair generation.

The guidance documents (Datasheet and AGD) are delivered to the customer by Nuvoton: the TCG main specification (TCG-x) is available publicly.

2.5 TOE Life Cycle Description

The life cycle of the TPM1.2 TOE includes several processes.

The following table details their purpose and where they are performed. These processes are mapped onto the Protection Profile life cycle phases (“TOE Life-Cycle” column).

In the table, the processes in white are Nuvoton’s responsibility, i.e., part of the evaluation under assurance class ALC; the processes in grey fall outside ALC Class evaluation criteria.

TOE Life-Cycle (ref: [PP])		**Processes**		**Who / Where**	**Inputs**	**Outputs**
TOE Development Environment	1	1a	Chip development	Design center	IC and dedicated software specification	Verilog/VHDL Verilog/VHDL source code; netlist
					Netlist	GDSII files
		1b	Firmware development	Design center	(TPM specification)	TPM firmware
	2	2a	Mask manufacturing	Mask Fab	GDSII files	Masks
		2b	Chip manufacturing	Wafer Fab	Masks	Wafers
		2c	Wafer level testing (EWS)	Test Fab	Wafers	Wafers
		2d	Device packaging (wafer sawing and plastic molding)	Assembly plant	Wafers	Packaged chips (blank)
		2f	Final test	Test plant	Packaged chips (blank)	Packaged chips (programmed)
TOE Delivery		3	TOE delivery	Test plant	Packaged chips (programmed)	Packaged chips (programmed)
TOE Preparation	3		Platform delivery	PC manufacturer	TPM firmware (upgrade)	TPM chip with upgraded firmware
	4		Platform deployment			
	5		Platform identity registration			
TOE Operational	6		Platform operation	PC end user	TPM firmware (upgrade)	TPM chip with upgraded firmware
	7		Platform recycling and retirement			

The PC manufacturer or the end user may use the Field Upgrade functionality to install a new TPM firmware on hardware of a previous TPM after delivery of the TOE. The TPM accepts only authentic update data provided by the TOE vendor.

Field Upgrade preserves user data and TSF data. Thus Field Upgrade does not change the Storage key hierarchy for protection of owner-related and other user and TSF data. After Field Upgrade, the new TPM is ready for operational use in the environment of the end user.

The installation of the new firmware may be performed in phases 3 to 6. The previous TOE requires authorization for Firmware Upgrade and verifies the integrity and authenticity of the TPM firmware upgrade data, as provided by the TPM firmware manufacturer.

Sites of the Development Environment:

- Design Center: Nuvoton Technology Israel, Hasadnaot 8, Hertzelia, Israel
- Design Center 2: Nuvoton Technology Israel, Hataa'sia 8, Ramat Gavriel, Migdal Haemek, Israel
- Design: Common Criteria Consultancy Services: Trusted Labs S.A.S, 5 rue du Bailliage, 78000 Versailles, France
- Mask Fabs:
    - TOPPAN Photomasks France, 224 Bld JF Kennedy, 91105 CORBEIL-ESSONNES Cedex, France
    - TOPPAN Photomasks Germany, Rähnitzer Allee 9, 01109 Dresden, Germany
- Wafer Fabs:
    - Tower Semiconductor Ltd., Israel
    - TowerJazz Panasonic Semiconductor Corporation (TPSCo), Japan
- Assembly plants:
    - AMKOR TECHNOLOGY PHILIPPINES, INC. (ATP) - P1, KM 22 East Service Road, Special Economic Zone, Cupang, Muntinlupa City PHILIPPINES 1702
    - AMKOR TECHNOLOGY PHILIPPINES, INC. (ATP) - P3/P4, 119 North Science Avenue, Special Economic Processing Zone, Laguna Technopark, Binan Laguna, PHILIPPINES 4024
    - ASE GROUP Chung-Li - 550,Chung-Hwa Road, Section 1Chung-Li, 320, Taiwan, R.O.C
- Wafer test and final test plant:
    - Nuvoton Technology Taiwan - No. 4, Creation Rd. III, Hsinchu Science Park, Taiwan, R.O.C
    - ASE GROUP Chung-Li - 550,Chung-Hwa Road, Section 1Chung-Li, 320, Taiwan, R.O.C

3 Conformance Claims

3.1 CC Conformance Claim

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

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

3.2 PP Claim

This Security Target is in strict conformance with the TCG PC Client-Specific TPM family 1.2 Level 2 revision 116 Protection Profile [PP].

An additional OSP deals with the TOE firmware field upgrade capability. This OSP implies an additional TOE objective and an additional TOE environment objective. The relevant PP SFRs are updated for the firmware upgrade capability.

The Protection Profile is registered and certified by the BSI under the reference BSI-CC-PP-0030-2008-MA-01 dated October 6, 2011.

3.3 Package Claim

This Security Target is conformant with the assurance package defined in the claimed Protection Profile: 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 1.2, as defined in the PP ([TCG-x]); thus 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 (an organizational security policy has been added for the TOE firmware field upgrade capability).

The security objectives are consistent with the statement of the security objectives of the PP (a TOE objective and a TOE environment objective have been added).

The security requirements are consistent with the statement of the security requirements of the PP (two SFRs are added for the firmware field upgrade security features). 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 completely with supplements regarding the Field Upgrade support.

4.1 Threats to Security

Threats to the TOE are defined in the PP (PP, Chapter 4).

4.2 Organizational Security Policies

OSPs are defined in the PP ([PP], Chapter 4) with the following supplement:

Table 1 – Secure Usage Assumptions

#	OSP	Description
1	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.

4.3 Secure Usage Assumptions

TOE secure usage assumptions are defined in the PP (PP, Chapter 4).

4.4 Security Problem Definition Rational

Strict conformance to the PP is claimed. It is a relevant claim as:

- The threats in the ST are exactly those of the claimed PP (no additional threat added by the presence of the Firmware Field Upgrade functionality);
- The assumptions in the ST are identical to those of the claimed PP (no additional assumption added by the presence of the Firmware Field Upgrade functionality).

The OSPs in the ST is however a superset of the OSPs in the claimed PP: “OSP.FieldUpgrade” is added. This additional OSP induces additional security objectives: “O.FieldUpgradeControl” and “OE.FieldUpgradeInfo”.

5 Security Objectives

The content of the PP ([PP], Chapter 5) applies to this chapter completely with supplements regarding the Field Upgrade support.

5.1 Security Objectives for the TOE

TOE security objectives are defined in the PP (PP, Chapter 5) with the following supplement:

Table 2. – Security Objectives for the TOE

#	Objective	Description
1	O.FieldUpgradeControl	The TOE provides a field upgrade capability with the following security features: • Control of authenticity and integrity of the loaded firmware • If the field upgrade process succeeds, then the resulting product is the Final TOE; otherwise (in case of interruption or incident which prevents the forming of the Final TOE), the Initial TOE remains in its initial state or fail secure • Control of the loaded version (no possibility of loading an uncertified firmware version) • Identification of the final TOE (allowing identification of the Initial TOE and of the loaded firmware)

5.2 Security Objectives for the Operational Environment

TOE security objectives for the operational environment are defined in the PP (PP, Chapter 5) with the following supplement:

Table 3. – Security Objectives for the Environment

#	Objective Name	Objective Description
1	OE.FieldUpgradeInfo	The end user shall be aware of the Field Upgrade process, its result and the version of the certified TPM.

5.3 Security Objectives Rationales

Strict conformance to the PP is claimed. It is a relevant claim as the statement of security requirements in the ST is a superset of the security requirements in the claimed PP.

The additional security objective **O.FieldUpgradeControl** requires that the TOE restricts access to the Field Upgrade capability and accepts only authentic update data provided by the TOE vendor. This objective is addressed by the following SFRs:

- FMT_SMR.1/**Security roles** defines a set of roles that the TSF shall maintain. Also, the association of users with these roles is required by this SFR, the TPM owner.

- FDP_ACF.1/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.
- FDP_UIT.1/Firmware requires that the TSF shall enforce a SFP to provide and use integrity protection capabilities for firmware update data on reception of that data.
- FDP_UCT.1/Firmware requires that the TSF shall enforce a SFP to use confidentiality protection capabilities for firmware update data on reception of that data.
- FPT_FLS.1 requires that the TSF shall preserve a secure state during a failure of the field upgrade process
- FMT_MSA.2 requires that the TSF shall ensure that only secure values are accepted for the TPM_FiledUpgrade command.
- FCS_COP.1 This SFR identifies the cryptographic algorithms available on the TOE. As regards the FieldUpgrade capabilities data decryption is performed by an AES 128 CTR mode, and integrity control is performed by using RSA signature scheme PKCS#1 v2.0.

Overall, only the two SFRs “FDP_UIT.1” and “FDP_UCT.1” have been added to those of the PP; these SFRs are only relevant to **O.FieldUpgradeControl**.

In spite of these two additional SFRs, the rationale given in the PP remains fully valid for the Security Target. This rationale demonstrates that each security objective for the TOE is covered by at least one security functional requirements, and each dependency of the security requirements is satisfied (or justified when the dependency is not justified).

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] §3 (but is not reproduced here).

Selections, assignments and refinements performed in the [PP] are indicated by *italics.*

All iterations from the PP are kept in the following text. Most of the application notes from the PP are not reproduced.

Subjects, Objects, Operations and User roles in the Security Functional Requirements are all defined in [PP] §1.3.4 (but are not reproduced 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) TPM owner* *2) Entity owner* *3) Delegated entity* *4) Entity user* *5) User using operatorAuth* 6) *“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 the TPM modes of operation* *2) Management of Delegation Tables and Family Tables* *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 Action Flag of TPM dictionary attack mitigation mechanism* *8) None.*

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 TPM_Field Upgrade security attributes related.*

FPT_TDC.1 Inter-TSF Basic TSF Data Consistency

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TDC.1.1	The TSF shall provide the capability to consistently interpret *digest of migrated Key, migratable Key Flag, payload field of migrated key, MSA ticket* when shared between the TSF and another trusted IT product.
FPT_TDC.1.2	The TSF shall use roles defined in [TCG-2] and [TCG-3] when interpreting the TSF data from another trusted IT product.

6.1.2 Cryptographic Support

FCS_CKM.1 Cryptographic Key Generation

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 keys in accordance with a specified cryptographic key generation algorithm RSA key generator and specified cryptographic key sizes RSA 512, 1024, 2048 that meet the following: P1363 [P1363].
FCS_CKM.1.1/*AES*	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *AES key generator* and specified cryptographic key sizes *128* bits that meet the following: *none*
Application note:	*All 128-bit AES keys are generated by the internal TPM random number generator.*

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. The hardware RNG output bits are used as input of a FIPS-approved DRNG algorithm (FIPS 186-2 Appendix 3.1) that relies on an implementation of the SHA-1 algorithm at the firmware level.*
FCS_RNG.1.2	The TSF shall provide random numbers that meet *FIPS 186-2 Appendix 3.1.*

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

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	The TSF shall perform the cryptographic operations identified in the table below in accordance with a specified cryptographic algorithm id*entified in the table* below and cryptographic keys of size *identified in the table below* that meet the following: see "*Standards" (see table).*

Operation	**Algorithm**	**Key Size (bits)**	**Standards**
Hash calculation	*SHA-1*	*Not applicable*	*FIPS180-2*
HMAC calculation and verification	*HMAC SHA-1*	*160*	*RFC2104, FIPS180-2*
Signature generation and verification	*RSA signature* *[TCG-1] 31.2.1, 31.2.2 and 31.2.3*	*512, 1024, 2048*	*PKCS#1 V2.0* *FIPS180-2*
Encryption and decryption	*RSA encryption* *[TCG-1] 31.1.1*	*512, 1024, 2048*	*PKCS#1 V2.0*
Symmetric encryption and decryption	*TPM_ALG_MGF1*	*(variable)*	*PKCS#1 V2.0*
Symmetric encryption and decryption	*AES mode CTR, ECB*	*128*	*FIPS 197*

6.1.3 TPM Operational Modes

FDP_ACC.1/Modes Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/Modes	The TSF shall enforce the TPM Mode Control SFP on all subjects, all objects *and all commands.*

FDP_ACF.1/Modes Security Attribute-Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static Attribute Initialization
FDP_ACF.1.1/Modes	The TSF shall enforce the TPM Mode Control SFP to objects based on the following: all subjects and all objects, flags disable, deactivated, owner and ownership.
FDP_ACF.1.2/Modes	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *1) The TPM shall prevent the execution of a command if the TPM is disabled and the command to be executed for the operation is not available according to Table 5, "Avail Disabled" column (see Appendix Section 8.1).* *2) The TPM shall prevent the execution of a command if the TPM is permanently or temporarily inactive and the command to be executed for the operation is not available according to Table 5, "Avail Deactivated" column (see Appendix Section 8.1).* *3) The TPM shall prevent the execution of a command if the TPM is unowned and the command to be executed for the operation is not allowed according to Table 5, "No Owner" column (see Appendix Section 8.1).* *4) The TPM will prevent use of firmware update data when authenticity of these data is not successfully verified.*
FDP_ACF.1.3/Modes	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *1) The command marked with "A" in Table 5, "Avail Disabled" column (see Appendix Section 8.1), is allowed to be executed if the TPM is disabled and the underlying NV storage does not require authorization.* *2) The command to be executed for the operation is marked with "A" in Table 5, Avail Deactivated column (see Appendix Section 8.1), is allowed to be executed if the TPM is permanently or temporarily inactive and the underlying NV storage does not require authorization.*
FDP_ACF.1.4/Modes	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *none*.

FDP_UIT.1/Firmware Data Exchange Integrity

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	The TSF shall enforce the TPM Mode Control SFP to receive user data in a manner protected from modification/errors.
FDP_UIT.1.2	The TSF shall be able to determine on receipt of user data, whether modification has occurred.

Note: This SFR is not part of the claimed PP. It has been added to cover the TOE firmware upgrade capability (source: TPM2.0 draft PP, where “firmware update data” are considered as “user data”).

FDP_UCT.1/Firmware Data Exchange Confidentiality

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_UCT.1.1	The TSF shall enforce the TPM Mode Control SFP to receive user data in a manner protected from unauthorized disclosure.

Note: This SFR is not part of the claimed PP. It has been added to cover the TOE firmware upgrade capability (source: TPM2.0 draft PP, where “firmware update data” are considered as “user data”).

FMT_MSA.1/Modes Management of Security Attributes

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/Modes	The TSF shall enforce the TPM Mode Control SFP to restrict the ability to modify the security attributes TPM operational mode *flags disable, deactivated and ownership* to *TPM owner, role using operatorAuth and user "World" under physical presence based on the rules:* *1) The TPM is disabled, inactive and unowned when created.* *2) The TPM owner is allowed to set the TPM operational modes to disabled, inactive and unowned.* *3) The TPM owner is allowed to set the TPM operational modes to enabled and disabled.* *4) A user "World" is allowed to own an enabled and unowned TPM if the flag ownership is TRUE.* *5) A user "World" under physical presence is allowed to set the TPM operational modes to disabled inactive and unowned at once.* *6) A user "World" under physical presence is allowed to set permanently an enabled TPM to active and inactive.* *7) The user "World" under physical presence is allowed to deactivate temporarily an enabled and active TPM.* *8) The user authenticated by operatorAuth is allowed to deactivate temporarily an enabled and active TPM.* *9) A user "World" under physical presence is allowed to set the TPM operational modes to enabled and to disabled.* *10) A user is not allowed to own a disabled or owned TPM.* *11) A user is not allowed to activate or deactivate a disabled TPM without setting unowned at the same time.* *12) A user "World" under physical presence is allowed to set the flag ownership to TRUE.* *13) The TPM owner is allowed to modify the flag ownership.*

FMT_MSA.1/PhysP Management of Security Attributes

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/PhysP	The TSF shall enforce the *TPM Mode Control SFP, Delegation SFP, Key Management SFP, NVS SFP* to restrict the ability to *set to the default value, assert by HW, assert by command, enable HW setting, disable HW setting, enable SW setting, disable SW setting, locking temporarily, locking permanently* the security attributes *physical presence* to *user "World" based on the additional rules:*

1) *If TPM_STCLEAR_FLAGS ->physicalPresenceLock is TRUE then assertion by command locking temporarily is not allowed.*
2) *If TPM_PERMANENT_FLAGS ->physicalPresenceHWEnable is FALSE then assertion by hardware is not allowed.*
3) *If TPM_PERMANENT_FLAGS ->physicalPresenceCMDEnable is FALSE then assertion by command and locking temporarily are not allowed.*
4) *If TPM_PERMANENT_FLAGS ->physicalPresenceLifetimeLock is TRUE then modifications to the states of flags that enable HW setting, disable HW setting, enable SW setting, disable SW setting, and locking permanently are not allowed.*

6.1.4 Identification, Authentication and Binding

FMT_MTD.1/AuthData Management of TSF Data

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

FMT_MTD.1.1/AuthData The TSF shall restrict the ability to modify and create the authentic*ation data* to *TPM Owner, user under physical presence and Entity Owner based on the rules:*

1) *The registering user creates the authentication data for the role TPM Owner by successful execution of the TPM_TakeOwnership command.*
2) *The registering user under physical presence creates the authentication data operatorAuth by successful execution of the TPM_SetOperatorAuth command.*
3) *The Entity Owner creates the authentication data for a new object by creating this object within an ADIP session.*
4) *The TPM owner modifies the authentication data for the role TPM Owner and for the object Storage Root Key by successful execution of the TPM_ChangeAuthOwner command.*
5) *The user under physical presence modifies the authentication data operatorAuth by successful execution of the TPM_SetOperatorAuth command.*
6) *The Entity Owner modifies the authentication data for the owned object by successful execution of the TPM_ChangeAuth command.*
7) *The Entity Owner modifies the authentication data for the owned object by successful execution of the TPM_ChangeAuthAsymStart and TPM_ChangeAuthAsymFinish commands.*

FMT_MTD.1/Deleg Management of TSF Data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security Roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1/Deleg The TSF shall restrict the ability to *modify and create* the authentication *data of a delegation blob* to *TPM Owner and authorized users based on the rules:*

1) *If TPM owner creates authentication data for a delegation blob by means of the TPM_Delegate_CreateOwnerDelegation command then the delegated access rights are equal to the permissions defined by publicInfo.*
2) *If the authorization of the TPM_Delegate_CreateOwner-Delegation command is a delegation of an enabled delegation family with valid verificationCount, the publicInfo identifies a delegation row of this family, and the access rights bits set in the publicInfo are a subset of the access rights bits set in this identified delegation table row then the delegated access rights are equal to the publicInfo.*

FIA_UID.1 Timing of Identification

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UID.1.1	The TSF shall allow: *1) Executing commands indicated in Table 5, "RQU" column (see Appendix Section 8.1,) as not requesting authentication.* *2) Accessing objects where Entity owner has given the user "World" access based on the value of TPM_AUTH_DATA_USAGE.* *3) None.* on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to be identified successfully 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) *Executing commands indicated in Table 5, "RQU" column (see Appendix Section 8.1), as not requesting authentication* 2) *Accessing objects where Entity owner has given the user "World" access based on the value of TPM_AUTH_DATA_USAGE.* *3) None.* 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.4 Single-Use Authentication

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.4.1	The TSF shall prevent reuse of authentication data related to 1) *OIAP authorization session* 2) *OSAP authorization session* 3) *DSAP authorization session* 4) *Transport session*

FIA_UAU.5 Multiple Authentication Mechanisms

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FIA_UAU.5.1	The TSF shall provide 1) *OIAP authorization session* 2) *OSAP authorization session* 3) *DSAP authorization session* 4) *Transport session* 5) *Commands that require authorization and are executed outside an authorization session*. to support user authentication.
FIA_UAU.5.2	The TSF shall authenticate any user's claimed identity according to the *rule "Dictionary attack": to detect and mitigate dictionary attacks*

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 condition the user sent *a command that requires authentication within a session*.

FIA_AFL.1 Authentication Failure Handling

Hierarchical to:	No other components.
Dependencies:	FIA_UAU.1 Timing of Authentication
FIA_AFL.1.1	The TSF shall detect when *1* unsuccessful authentication attempt occurs related to authentication attempts for the same user.
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall 1) *Set the Action Flag to TRUE.* *2) Increment a number of failure counter and switch the TPM to a lockout state during a period depending of the counter value and growing exponentially. The set of allowed commands is reduced to authorization not-required commands. The whole set of commands is available again after the period has elapsed.*

FMT_MTD.1/Lock Management of TSF Data

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security Roles FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/Lock	The TSF shall restrict the ability to reset to FALSE the Action Flag of TPM *dictionary attack mitigation mechanism* to *the TPM Owner and Delegated Entity.*

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 behalf of that user: 1) *authData* 2) *locality* 3) Physical presence 4) *Authorization handle and shared secret if the subject is a OSAP or DSAP session* 5) *Authorization associated with the delegation blob if the subject is a DSAP session*
FIA_USB.1.2	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on behalf of users: 1) *The shared secret is associated with the authorization gained by the user providing the AuthData for the entity identified in the TPM_OSAP command establishing the OSAP session.* 2) *The shared secret is associated with the authorization gained by the user providing the AuthData and the delegation blob for establishing the DSAP session.* 3) *The present value of the user locality is assigned to the command executed by this user*. 4) *The physical presence of the user is assigned to the command executed by that user.*

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

1) *The TSF shall set the security attributes of the subject TPM Owner to values defined by the TPM_OwnerClear command if:*
   a) *the subject executing the TPM_OwnerClear command is bound to the TPM owner and all command parameters and the security attribute DisableOwnerClear are FALSE*
   b) *the subject with physical presence is executing the TPM_ForceClear command and the security attribute disableForceClear is FALSE*
2) *The TSF shall delete the shared secret for the authorization of the OSAP session if the user executes the TPM_Reset command.*
3) *The TSF shall delete the shared secret for the authorization of the OSAP session and DSAP session if:*
   a) *the user executes the TPM_FlushSpecific or the TPM_Terminate_Handle command.*
   b) *the user clears the TPM Owner by executing the TPM_OwnerClear or TPM_ForceClear command.*
   c) *the user is the TPM owner and executes the TPM_ChangeAuthOwner command.*
   d) *any of the following commands are executed:*
      i. *TPM_Delegate_Manage*
      ii. *TPM_Delegate_CreateOwnerDelegation with Increment==TRUE*
      iii. *TPM_Delegate_LoadOwnerDelegation*
4) *The TSF shall delete enforced by the user the shared secret for the authorization of all OSAP sessions associated with the counter by executing the TPM_ReleaseCounter or TPM_ReleaseCounterOwner command.*
5) *The TSF shall delete the shared secret for the authorization of the session if the user sets the continueUse flag to FALSE in the command within an OSAP or DSAP session.*
6) *The TSF shall delete automatically the shared secret for the authorization of the OSAP session and DSAP session acting on behalf of users after the session:*
   a) *executes a command that returns an error*
   b) *uses a resource evicted from the TOE or otherwise invalidated*
   c) *executes any command for which the shared secret is used to encrypt an input parameter (TPM_ENCAUTH)*

6.1.5 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: *any object.*

Delegation

FDP_ACC.1/Deleg Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/Deleg	The TSF shall enforce the *Delegation SFP* on *Delegated Entities, user data and commands*.

FDP_ACF.1/Deleg Security Attribute-Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static Attribute Initialization
FDP_ACF.1.1/Deleg	The TSF shall enforce the Delegation SFP to objects based on the following: Delegated Entities and commands with the delegated permission defined in the Delegation table row, locality, pcrInfo and key handle of the key in the Delegation owner blob.
FDP_ACF.1.2/Deleg	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: 1) *The TSF shall disallow the execution of a command in a DSAP session if the permission of this command is not set in the Delegation table row in the Delegation owner blob used for the DSAP session.* 2) *The TSF shall disallow the execution of a command in a DSAP session if the PCR_SELECTION of the DSAP session is not NULL and the pcrInfo of the DSAP session does not match the current PCR value of the PCR_SELECTION and locality*.
FDP_ACF.1.3/Deleg	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *1) The TSF shall allow the delegation of the TPM Owner authorized commands listed in [TCG-2], table of Section 20.2.1.* *2) The TSF shall allow the delegation of the TPM Key authorized commands listed in [TCG-2], table of Section 20.2.3.*

FDP_ACF.1.4/Deleg	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *1) The TSF shall deny the delegation of the TPM Owner authorized commands listed in [TCG-2], table of Section 20.2.2.* *2) The TSF shall deny the delegation of the TPM Key authorized commands listed in [TCG-2], table of Section 20.2.4.*

FMT_MSA.1/DFT Management of Security Attributes

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/DFT	The TSF shall enforce the Delegation SFP to restrict the ability to modify, delete, enable, disable and create the security attributes *Family table* to: 1) *TPM owner* 2) *User under physical presence if:* a) *the opCode is TPM_FAMILY_CREATE* b) *DisableForceClearis FALSE* *c) TPM_Delegate_Admin_Lock is false*

FMT_MSA.1/DT Management of Security Attributes

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/DT	The TSF shall enforce the Delegation SFP to restrict the ability to query, modify *and create* the security attributes *Delegation table* to: 1) *TPM owner* 2) *User "World" if the TPM owner is not installed and max NV writes without an owner is not exceeded and TPM_Delegate_Admin_Lock is false.*

FMT_MSA.3/Deleg Static Attribute Initialization

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of Security Attributes FMT_SMR.1 Security Roles
FMT_MSA.3.1/Deleg	The TSF shall enforce the Delegation SFP to provide permissive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/Deleg	The TSF shall allow the TPM owner to specify alternative initial values to override the default values when an object or information is created.

Key Management

FDP_ACC.1/KeyMan Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/KeyMan	The TSF shall enforce the *Key Management SFP* on 1) *Subjects: commands executing on behalf of users.* 2) *Objects: keys.* 3) *Operations: create, activate AIK, delete, export, import, signature generation, encryption, decryption.*

FDP_ACF.1/KeyMan Security Attribute-Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static Attribute Initialization
FDP_ACF.1.1/KeyMan	The TSF shall enforce the *Key Management SFP* to objects based on the following: 1) *Subjects: commands with security attributes ownerAuth, srkAuth, AuthData, locality, physical presence.* 2) *Objects:* a) *EK with the SFR-related, security attribute ownership of the TOE.* b) *SRK with the SFR-related, security attribute disableOwnerClear and disableForceClear of the TOE.* c) *User keys with security attributes authDataUsage, keyUsage, keyFlags, and OwnerEvict.* d) *Wrapped Key Blob with security attributes keyUsage, keyFlags, algorithmParms and pcrInfo.*

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

1) *The "World" user is allowed to create an EK if the EK does not exist already.*
2) *The "World" user is allowed to read the public part of an EK if the TOE is unonwed.*
3) *The TPM owner is allowed to read the public part of an EK.*
4) *The "World" user is allowed to create an SRK if the ownership flag is TRUE.*
5) *The TPM owner is allowed to delete an SRK if the disableOwnerClear flag is FALSE.*
6) *The "World" user under physical presence is allowed to delete an SRK if the disableForceClear flag is FALSE.*
7) *The user authenticated as TPM owner and the owner of the SRK is allowed to generate an AIK.*
8) *The TPM owner is allowed to activate the AIK if the imported blob is a TPM_EK_BLOB structure and the actual state meets the identified PCR values and the locality.*
9) *The TPM owner is allowed to use the AIK for signing audit data, quoted data, or a tick stamped blob.*
10) *The Entity owner of a key with the security attribute keyUsage, TPM_KEY_STORAGE = TRUE, is allowed to generate a User Key and export this User key wrapped with the key he owns except if this Entity owner is not the TPM owner and the key generated is an AIK.*
11) *The Entity owner of the key to be used for import of Wrapped Key Blob is allowed to import a User key in a Wrapped Key Blob if the security attribute keyUsage, TPM_KEY_STORAGE = TRUE, of the import key is set.*
12) *The Entity owner is not allowed to use a User key if at least one of the following conditions is met:*
    a) *the security attribute authDataUsage of the User Key object for access does not match the authentication status of the subject,*
    b) *the security attribute usageAuth of the User Key object for access does not match the authentication data used by the user bound to the subject,*
    c) *the security attributes keyUsage or algorithmParms or keyFlags of the User Key object does not allow use of the command to be executed,*
    d) *the security attribute PCRInfo of the User Key object does not allow use of the object in the current state of the identified PCR and locality*.
13) *The TPM owner is allowed to delete a User key if the security attribute OwnerEvict, OwnerEvict = FALSE.*

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

1) *Bits setting of the security attribute keyFlags affects the capacity of the TPM commands to perform management operation on the objects User keys or Wrapped Key Blob. Commands affected by this rule: TPM_Unseal, TPM_CreateWrapKey, TPM_LoadKey, TPM_LoadKey2, TPM_GetPubKey, TPM_CMK_SetRestrictions, TPM_CMK_CreateKey, TPM_CMK_CreateBlob, TPM_CMK_ConvertMigration, TPM_CertifyKey, TMP_CerifyKey2, TPM_MakeIdentity, TPM_DSAP, TPM_ChangeAuthAsymStart, TPM_LoadKey.*

2) *The value of the authDataUsage security attribute affects the capacity of the TPM commands to use the object User keys under authentication conditions. Commands affected by this rule: TPM_TakeOwnership, TPM_CreateWrapKey, TPM_LoadKey2, TPM_GetPubKey, TPM_CMK_CreateKey, TPM_CertifyKey, TPM_CertifyKey2, TPM_MakeIdentity, TPM_EstablishedTransport, TPM_ChangeAuthAsymStart, TPM_LoadKey.*

3) *The value of the keyUsage security attribute affects the capacity of the TPM commands to use the object User keys or Wrapped Key Blob for specific type of operations. Commands affected by this rule: TPM_CMK_SetRestrictions, TPM_ChangeAuthAsymStart, TPM_Take_Ownership, TPM_Seal, TPM_Unseal, TPM_Sealx, TPM_Unbind, TPM_Sign, TPM_CertifyKey, TPM_LoadKey, TPM_LoadKey2, TPM_CreateWrapKey, TPM_MakeIdentity, TPM_GetPubKey, TPM_MigrateKey, TPM_DSAP, TPM_Quote, TPM_ActivateIdentity, TPM_ConvertMogrationBlob, TPM_CertiySelfTest, TPM_CMK_CreateKey, TPM_CMK_ConvertMigrationBlob, TPM_TickStampBlob and TMK_EstablishTransport.*

4) *The status of ownerEvict security attribute affects the capacity of the TPM commands to evict the object User keys.*

5) *The values contained by the algorithmParms security attributes affect the capacity of the TPM commands to perform configuration operations on the User keys object. Commands affected by this rule: TPM_TakeOwnership, TPM_AuthorizeMigrationKey, TPM_CMK_CreateTicket and TPM_CMK_CreateBlob, TPM_CreateEndorsementKeyPair.*

6) *The values of pcrInfo security attributes affect the capacity of TPM commands to access the object Wrapped Key Blob for certain management operations. Commands affected by this rule: TPM_Seal, TPM_Unseal, TPM_LoadKey, TPM_LoadKey2, TPM_MakeIdentity, TPM_GetPubKey, TPM_Sealx, TPM_CertifyKey, TPM_CertifyKey2, TPM_CMK_CreateKey, TPM_NV_WriteValue, TPM_NV_WriteValueAuth, TPM_NV_ReadValueAuth.*

7) *The status of disableOwnerClear security attribute can allow the possibility to the owner to clear the object SRK.*
8) *The status of the disableForceClear security attribute can allow the possibility to the TPM_ForceClear command to be executed.*
9) *The status of the owner authorization (through flag TPM_PF_READPUBEK in TPM_PERMANENT_FLAGS) security attribute can allow the reading of the public portion of the object EK. Command affected by this rule: TPM_readPubek.*

FDP_ACF.1.4/KeyMan	The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *same rules as in FDP_ACF.1.3/KeyMan with different values.*

Application note: The values of the flags and attributes outlined in this requirement are defined in the TCG specification ([TCG-2], [TCG-3]). They are not detailed here to avoid overloading the text.

FMT_MSA.1/KeyMan Management of Security Attributes

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/KeyMan	The TSF shall enforce the Key Management SFP to restrict the ability *to assign the initial value* the security attributes: 1) *srkParamsof the SRK to user "World",* 2) *authDataUsage, usageAuth, keyUsage, algorithmParms, keyFlags and PCRInfo associated with the generated User key to the Entity owner.*

FMT_MSA.1/KEvi Management of Security Attributes

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/KEvi	The TSF shall enforce the Key Management SFP to restrict the ability to modify the security attributes TPM_KEY_CONTROL_OWNER_EVICT of a loaded key to the Entity *owner.*

FMT_MSA.3/KeyMan Static Attribute Initialization

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of Security Attributes FMT_SMR.1 Security Roles
FMT_MSA.3.1/KeyMan	The TSF shall enforce the Key Management SFP to provide restrictive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/KeyMan	The TSF shall allow *Entity owner* to specify alternative initial values to override the default values when an object or information is created.

Key Migration

FDP_ACC.1/MigK Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/Mig	The TSF shall enforce the *Key Migration SFP* on *1) Subjects: TPM owner, Entity owner;* *2) Objects: User key, Wrapped Key Blob, Migration Key Blob, Certified Migration Key Blob;* *3) Operations: TPM_CreateMigrationBlob, TPM_CMK_CreateKey, TPM_CMK_CreateBlob, TPM_CMK_ConvertMigration, TPM_ConvertMigrationBlob, TPM_MigrateKey commands.*

FDP_ACF.1/MigK Security Attribute-Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static Attribute Initialization
FDP_ACF.1.1/MigK	The TSF shall enforce the *Key Migration SFP* to objects based on the following: *1) Subjects: TPM owner, Entity owner of the key with security attributes restrictDelegate and migrationScheme,* *2) Objects:* *a) User key with security attribute migratable,* *b) Wrapped Key Blob with the security attribute payload type,* *c) Migration Key Blob with the security attribute payload type,* *d) Certified Migration Key Blob with the security attributes payload type and migrationAuth.*

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

1) *The Entity owner of a certifiable migratable User key is allowed to create a Wrapped Key Blob for this migratable key by means of the TPM_CMK_CreateKey command, if it is authorized by use of the CMK Migration Approval Ticket and in case of delegated commands the restrictions for the migration of keys are fulfilled.*
2) *The Entity owner of a migratable User key authorized for use of the Migration key authorization ticket is allowed to create a Migration Key Blob for this migratable key by means of the TPM_CreateMigrationBlob command.*
3) *The Entity owner of a certifiable migratable User key authorized for use of the Migration key authorization ticket and the Restriction Ticket is allowed to create a Certified Migration Key Blob for this migratable key by means of the TPM_CMK_CreateBlob command.*
4) *The Entity owner of private part of the migration User key is allowed to migrate a Migration Key Blob and a Certified Migration Key Blob to a conversion key by means of the TPM_MigrateKey command.*
5) *The Entity owner of the private part of migration User key is allowed to convert a Migration Key Blob by means of the TPM_ConvertMigrationBlob command and a Certified Migration Key Blob by means of the PM_CMK_ConvertMigration command if in case of delegated commands the restrictions for the migration of keys are fulfilled*.

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

1) *The value of the migrationScheme security attribute affects the capacity of the subject TPM owner or Entity owner to create an object Migration Key Blob. Commands affected by this rule: TPM_CreateMigrationBlob and TPM_CreateBlob.*
2) *The value of the migratable security attribute affects the capacity of the subject Entity owner to access an object User key to perform operations related to migration. Commands affected by this rule: TPM_CMK_CreateKey, TPM_CMK_CreateBlob and TPM_CMK_ConvertMigration.*
3) *The value of the payload type security attribute affects the capacity of the subject Entity owner or TPM owner to access an object Wrapped Key Blob, Migration Key Blob or Certified Migration Key Blob to perform migration related operations. Commands affected by this rule: TPM_CreateMigrationBlob, TPM_ConvertMigrationBlob, TPM_CMK_CreateKey, TPM_CMK_CreateBlob and TPM_CMK_ConvertMigration.*

4) *The value of the migrationAuth security attribute affects the capacity of the subject Entity owner or TPM owner to create the Migration Key Blob. Commands affected by this rule: TPM_CreateMigrationBlob and TPM_CMK_CreateBlob.*

FDP_ACF.1.4/MigK	The TSF shall explicitly deny access of subjects to objects based on the following additional rules*: same rules as in FDP_ACF.1.3/MigK with different values.*

Application note: the values of the flags and attributes outlined in this requirement are defined in the TCG specification ([TCG-2], [TCG-3]). They are not detailed here to avoid overloading the text.

FMT_MSA.1/MigK Management of Security Attributes

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/MigK	The TSF shall enforce the Key Migration SFP to restrict the ability to assign initial value the security attributes restrictDelegate*, migrationScheme, migrationAuthorityApproval* to *TPM owner*.

FMT_MTD.1/MigK Management of TSF Data

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security Roles
	FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/MigK	The TSF shall restrict the ability to create the CMK Migration Approval *Ticket, Migration Key Authorization Ticket, Restrict Ticket* to *TPM owner*.

Measurement and Reporting

FDP_ACC.1/M&R Subset Access Control

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

1) *Subjects: SHA-1 session, user "World" and Entity owner,*
2) *Objects: PCR, User key,*
3) *Operations: TPM_SHA1Start, TPM_SHA1Update, TPM_SHA1Complete, TPM_SHA1CompleteExtend TPM_PCR_Reset, TPM_Extend, TPM_PCRRead, TPM_Quote TPM_Quote2, TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END commands.*

FDP_ACF.1/M&R Security Attribute-Based Access Control

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:

1) *Subjects:*
    a) *SHA-1 session*
    b) *user with the security attributes locality*
    c) *Entity owner of the signature key with the security attribute usageAuth*
2) *Objects:*
    d) *PCR with security attributes pcrReset and pcrResetLocal*
    e) *pcrExtendLocalUser key with security attribute keyUsage*

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 SHA-1 session is allowed to reset the digest of the SHA-1 session by the TPM_SHA1Start command.*
2) *The SHA-1 session is allowed to calculate the new digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data by the TPM_SHA1Update command.*
3) *The SHA-1 session is allowed (i) to finish the calculation of the digest of the SHA-1 session as the SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to output the hash value by the TPM_SHA1Complete command.*

4) *The SHA-1 session is allowed (i) to finish the calculation of the digest of the SHA-1 session as SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to extend the value of the indicated PCR by the PM_SHA1CompleteExtend command.*
5) *If the pcrReset is TRUE, the TPM_Startup command is allowed to set a PCR to 0xFF…FF.*
6) *If the pcrReset is FALSE the TPM_Startup command is allowed to set a PCR to 0x00…00.*
7) *If the user presents the locality matching the security attribute pcrResetLocal of the selected PCR and the pcrReset of this PCR is TRUE, then the TPM_PCR_Reset command is allowed to reset this PCR to 0x00…00 or 0xFF…FF, where the concrete value is defined in the platform specific specification of the TOE.*
8) *If the user presents the locality matching the security attribute pcrExtendLocal of the selected PCR, the TPM_SHA1CompleteExtend command is allowed (i) to finish the calculation of the digest of the SHA-1 session as the SHA-1 hash value of the digest of the SHA-1 session and the presented data and (ii) to extend the value of the selected PCR with the final digest of the SHA-1 session.*
9) *If the user presents the locality matching the security attribute pcrExtendLocal of the selected PCR, the TPM_Extend command is allowed to extend the value of the selected PCR with the presented data.*
10) *The user "World" is allowed to read the PCR object with the TPM_PCRRead command.*
11) *The Entity owner is allowed to quote the PCR indicated by the parameter targetPCR with the User key, which security attribute keyUsage equals to TPM_KEY_SIGNING, TPM_KEY_IDENTITY, or TPM_KEY_LEGACY, by means of the TPM_Quote or TPM_Quote2 command.*
12) *The user "World" under locality 4 is allowed to execute LPC commands TPM_HASH_START, TPM_HASH_DATA and TPM_HASH_END.*
13) *None.*

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

1) *The value of the pcrReset and pcrResetLocal security attributes affects the capacity of a user with specific locality security attribute to access a PCR object to interfere in Measurement and Reporting operations mechanism. Command affected by this rule: TPM_PCR_Reset.*
2) *The value of the pcrExtendLocal security attribute affects the capacity of the Entity owner of a key to access the User key object to perform Measurement and Reporting operations. Commands affected by this rule: TPM_SHA1CompleteExtend and TPM_Extend.*

3) *The value of the keyUsage security attribute affects the capacity of the user with specific locality security attribute to access the object PCR to perform Measurement and Reporting operations. Commands affected by this rule: TPM_Quote and TPM_Quote2.*

FDP_ACF.1.4/M&R The TSF shall explicitly deny access of subjects to objects based on the following additional rules*: same rules as in FDP_ACF.1.3/M&R with different values.*

Application note: the values of the flags and attributes outlined in this requirement are defined in the TCG specification ([TCG-2], [TCG-3]). They are not detailed here to avoid overloading the text.

FMT_MSA.3/M&R Static Attribute Initialization

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 the *Entity owner* 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

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 TPM_QUOTE_INFO or TPM_QUOTE_INFO2 structure at the request of the originator.

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

1) *PCR values of the requested PCR indices in case of TPM_QUOTE_INFO,*
2) *PCR values of the requested PCR indices, and locality at release in case of TPM_QUOTE_INFO2*

of the originator of the information, and

1) *external data in the TPM_QUOTE_INFO,*
2) *external data in the TPM_QUOTE_INFO2*

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 *recipient* given *the attributes of the Attestation Identity Key Credential if an Attestation Identity Key is used.*

Non-Volatile Storage

FDP_ACC.1/NVS Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/NVS	The TSF shall enforce the *NVS SFP* on 1) *Subjects: user "World", Entity owner and TPM owner,* 2) *Objects: NV storage areas,* 3) *Operations: create, write, read.*

FDP_ACF.1/NVS Security Attribute-Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static Attribute Initialization
FDP_ACF.1.1/NVS	The TSF shall enforce the *NVS SFP* to objects based on the following: 1) *Subjects: user "World", Entity owner and TPM owner with the security attributes physical presence, locality and current PCR values,* 2) *Objects: NV storage with the security attributes nvLocked, noOwnerNVWrite, pcrInfoRead, pcrInfoWrite, localityAtRelease, and permissions TPM_NV_PER_READ_STCLEAR, TPM_NV_PER_WRITE_STCLEAR TPM_NV_PER_AUTHWRITE, TPM_NV_PER_OWNERWRITE TPM_NV_PER_PPWRITE, TPM_NV_PER_AUTHREAD, TPM_NV_PER_PPREAD, TPM_NV_PER_OWNERREAD, TPM_MAX_NV_WRITE_NOOWNER.*
FDP_ACF.1.2/NVS	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: 1) *The user "World" under physical presence is allowed to create NV storage by means of the TPM_NV_DefineSpace command if nvLocked is 0 and noOwnerNVWrite does not exceed TPM_MAX_NV_WRITE_NOOWNER.* 2) *The TPM owner is allowed to create a NV storage area by means of the TPM_NV_DefineSpace command.* 3) *The user "World" is allowed to write the NV storage area if nvLocked of the TPM_PERMANENT_FLAGS is FALSE and max NV writes without an owner is not exceeded.*

4) The TPM owner is allowed to write an NV storage area by means of the TPM_NV_WriteValue command if:
   a) *TPM_NV_PER_OWNERWRITE is TRUE;*
   b) *the user satisfies the requirement for physical presence defined in TPM_NV_PER_PPWRITE;*
   c) *the locality of the user matches the localityAtRelease defined for the TPM_NV_DATA_AREA and;*
   d) *pcrInfWrite defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoWrite.*

5) The Entity owner is allowed to write an NV storage area by means of the TPM_NV_WriteValueAuth command if:
   a) *TPM_NV_PER_AUTHWRITE is TRUE;*
   b) *the user match the requirement for physical presence defined in TPM_NV_PER_PPWRITE;*
   c) *the locality of the user matches the localityAtRelease defined for the TPM_NV_DATA_AREA and;*
   d) *pcrInfWrite defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoWrite.*

6) The TPM owner is allowed to read an NV storage area by means of the TPM_NV_ReadValue command if:
   a) TPM_NV_PER_OWNERREAD is TRUE,
   b) the user matches the requirement for physical presence defined in TPM_NV_PER_PPREAD,
   c) the locality of the user matches the localityAtRelease defined in the pcrInfoRead and
   d) pcrInfoRead defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoRead.

7) The Entity owner is allowed to read an NV storage area by means of the TPM_NV_ReadValueAuth command if:
   a) TPM_NV_PER_AUTHREAD is TRUE;
   b) the user matches the requirement for physical presence defined in TPM_NV_PER_PPREAD;
   c) the locality of the user matches the localityAtRelease defined in the pcrInfoRead and;
   d) pcrInfoRead defines a PCR selection the actual values of the selected PCR shall match the digestAtRelease in pcrInfoRead.

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

1) Subjects are allowed to access NV storage object through the TPM_NV_DefineSpace command, depending on the values of security attributes nvLocked, pcrInfoRead, pcrInfoWrite, TPM_MAX_NV_WRITE_NOOWNER, TPM_NV_PER_OWNERWRITE, TPM_NV_PER_AUTHWRITE, TPM_NV_PER_AUTHREAD, TPM_NV_PER_ WRITEDEFINE and TPM_NV PER_PPWRITE.

2) *Subjects are allowed to access NV storage object through the TPM_NV_WriteValue command, depending on the values of security attributes nvLocked, pcrInfoWrite, localityAtRelease TPM_MAX_NV_WRITE_NOOWNER, TPM_NV_PER_OWNERWRITE, TPM_NV_PER_AUTHWRITE, TPM_NV_PER_PPWRITE and TPM_NV_PER_WRITE_STCLEAR. Subjects are allowed to access NV storage object through the TPM_NV_WriteValueAuth command, depending on the values of security attributes pcrInfoWrite, localityAtRelease TPM_NV_PER_AUTHWRITE, TPM_NV_PER_PPWRITE, TPM_NV_PER_WRITEDEFINE and TPM_NV_PER_WRITE_STCLEAR.*

3) *Subjects are allowed to access NV storage object through the TPM_NV_ReadValue command, depending on the values of security attributes nvLocked, pcrInfoRead, PM_NV_PER_AUTHREAD, TPM_NV_PER_OWNERREAD, TPM_NV_PER_PPREAD and TPM_NV_PER_READ_STCLEAR.*

4) *Subjects are allowed to access NV storage object through the TPM_NV_ReadValueAuth command, depending on the values of security attributes pcrInfoRead, localityAtRelease, TPM_NV_PER_AUTHREAD, TPM_NV_PER_PPREAD and TPM_NV_PER_READ_STCLEAR.*

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

1) *If TPM_NV_PER_READ_STCLEAR is TRUE the NV storage area cannot be read after read with a data size of 0 until successful write or TPM_Startup(ST_Clear).*
2) *If TPM_NV_PER_WRITE_STCLEAR is TRUE the NV storage area cannot be written after write to the specified index with a data size of 0 until TPM_Startup(ST_Clear).*
3) *If TPM_NV_PER_WRITEDEFINE is TRUE the NV storage area cannot be written after performing the TPM_NV_DefineSpace command and one successful write to the index with datasize of 0.*
4) *If TPM_NV_PER_GLOBALLOCK is TRUE the NV storage area cannot be written after successful write to index 0 until TPM_Startup(ST_Clear)*
5) *The access to commands:*
   a) *TPM_NV_WriteValue is denied if security attributes TPM_NV_PER_OWNERWRITE and TPM_NV_PER_AUTHWRITE are both set to TRUE.*
   b) *TPM_NV_ReadValue is denied if security attributes TPM_NV_PER_OWNERREAD and TPM_NV_PER_AUTHREAD are both set to TRUE.*
6) *Same rules as in FDP_ACF.1.3/NVS with different values.*

Application note: the values of the flags and attributes outlined in this requirement are defined in the TCG specification ([TCG-2], [TCG-3]). They are not detailed here to avoid overloading the text.

FMT_MSA.3/NVS Static Attribute Initialization

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of Security Attributes FMT_SMR.1 Security Roles
FMT_MSA.3.1/NVS	The TSF shall enforce the NVS SFP to provide restrictive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/NVS	The TSF shall allow the TPM owner and user "World" under physical *presence* to specify alternative initial values to override the default values when an object or information is created.

Counter

FDP_ACC.1/MC Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/MC	The TSF shall enforce the *Monotonic Counter SFP* on *1) Subjects: TPM owner, Delegated entity, Entity owner of the monotonic counter object, user "World",* *2) Objects: Monotonic counter,* 3) *Operations: create, increment, read, release*.

FDP_ACF.1/MC Security Attribute-Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static attribute initialization
FDP_ACF.1.1/MC	The TSF shall enforce the *Monotonic Counter SFP* to objects based on the following: *1) Subjects: TPM owner, Delegated entity, Entity owner of the monotonic counter object, user "World",* 2) *Objects: Monotonic counter with security attribute countID*.

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

1) *The TPM owner and Delegated entity are allowed to create a Monotonic counter, OSAP and DSAP sessions are required for creation of the Monotonic counter.*
2) *The Entity owner of the monotonic counter object is allowed to increment the Monotonic counter if the countID is set in TPM_STCLEAR_DATA for the current boot cycle.*
3) *The user "World" is allowed to read the Monotonic counter value if he addresses the Monotonic counter object correctly with valid countID.*
4) *The Entity owner of the monotonic counter object is allowed to release the Monotonic counter.*
5) *The TPM owner is allowed to release the Monotonic counter.*

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

1) *The value of the countID security attribute affects the capacity of a subject to access the object Monotonic counter to perform operations on the monotonic counter. Commands affected by this rule: TPM_IncrementCounter, TPM_ReadCounter, TPMReleaseCounter and TPM_ReleaseCounterOwner depends on the values of the security attribute countID.*

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

1) *The TSF shall disallow the operation read or increment the monotonic counter if the countID is invalid.*
2) *Same rules as in FDP_ACF.1.3/MC with different values.*

Application note: the values of the flags and attributes outlined in this requirement are defined in the TCG specification ([TCG-2], [TCG-3]). They are not detailed here to avoid overloading the text.

FMT_MSA.1/MC Management of Security Attributes

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/MC	The TSF shall enforce the *Monotonic Counter SFP* to restrict the ability to: *1) modify the security attributes countIDto the Entity owner executing TPM_IncrementCounter.* *2) set to NULL the security attributes countIDto TPM_Startup(ST_CLEAR),* *3) set to invalid value the security attributes countIDto* a) *Entity owner of the monotonic counter executing the TPM_ReleaseCounter command.* b) *TPM owner executing the TPM_ReleaseCounterOwner command.*

FMT_MSA.3/MC Static Attribute Initialization

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1/MC	The TSF shall enforce the *Monotonic Counter SFP* to provide restrictive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/MC	The TSF shall allow *no role* to specify alternative initial values to override the default values when an object or information is created.

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 *Tick Count Value of ticks since start of the tick session to an accuracy of tickRatemicroseconds.*

FCO_NRO.1/STS Selective Proof of Origin

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of Identification
FCO_NRO.1.1/STS	The TSF shall be able to generate evidence of origin for transmitted TPM_SIGN_INFO structure at the request of the originator.
FCO_NRO.1.2/STS	The TSF shall be able to relate the current tick count of the originator of the information, and external data in the TPM_SIGN_INFO structure of the information to which the evidence applies.
FCO_NRO.1.3/STS	The TSF shall provide a capability to verify the evidence of origin of information to *recipient* given *the attributes of the Attestation Identity Key Credential if an Attestation Identity Key is used.*

6.1.6 Data Import and Export

FDP_ACC.1/EID Subset Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/IED	The TSF shall enforce the *Export and Import of Data SFP* on 1) Subjects: *TPM owner, Entity owner*; *2)* Objects: *Sealed Data, Context, Bound Blob;* 3) Operations: *export, import, save, load, unbind*.

FDP_ACF.1/EID Security Attribute-Based Access Control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static Attribute Initialization
FDP_ACF.1.1/EID	The TSF shall enforce the *Export and Import of Data SFP* to objects based on the following: *1) Subjects: TPM owner with security attribute locality, Entity owner with security attribute locality, user "World";* *2) Objects:* *a) Sealed data with security attribute pcrInfo and tpmProof,* *b) Context with the security attribute resourceType and tpmProof,* *c) Bound Blob with the security attributes payload type.*

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

1) *The Entity owner of the key to be used for export of sealed data is allowed to export Sealed Data if this export key has the security attribute TPM_KEY_STORAGE and is not migratable.*
2) *The Entity owner of the key to be used for import of sealed data is allowed to import Sealed Data if:*
   a) *this import key has the security attribute TPM_KEY_STORAGE and is not migratable,*
   b) *the security attributes pcrInfo of sealed data blob match the values in the PCR indicated by pcrInfo,*
   c) *the security attributes tmpProof of sealed data blob match the values tpmProof in the TPM_PERMANENT_DATA of the TOE.*
3) *The user "World" is allowed to save Context if the resourceType is TPM_RT_KEY, TPM_RT_AUTH, TPM_RT_TRANS or TPM_RT_DAA_TPM.*
4) *The user "World" is allowed to load Context if*
   a) *the resourceType is TPM_RT_KEY, TPM_RT_AUTH, TPM_RT_TRANS or TPM_RT_DAA_TPM and*
   b) *the tpmProof used as secret for the HMAC of the context matches the tpmProof in TPM_PERMANENT_DATA.*
5) *The Entity owner of the private part of the bind key is allowed to unbind a Bound blob if the payload type is TPM_PT_BIND.*

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

a) The execution of the TPM_Unseal command depends on the value of security attributes TPMproof and payload type.

FDP_ACF.1.4/EID The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *same rules as in FDP_ACF.1.3/EID with different values.*

Application note: the values of the flags and attributes outlined in this requirement are defined in the TCG specification ([TCG-2], [TCG-3]). They are not detailed here to avoid overloading the text.

FMT_MSA.3/EID Static Attribute Initialization

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of Security Attributes FMT_SMR.1 Security Roles
FMT_MSA.3.1/EID	The TSF shall enforce the Export and Import of Data SFP to provide restrictive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/EID	The TSF shall allow the *Entity owner* to specify alternative initial values to override the default values when an object or information is created.

FDP_ETC.2 Export of User Data with Security Attributes

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access Control, or FDP_IFC.1 Subset Information Flow Control]
FDP_ETC.2.1	The TSF shall enforce the Key Management SFP, Key Migration SFP, Export and Import of Data SFP when exporting user data, controlled under the SFP(s), outside of the TOE.
FDP_ETC.2.2	The TSF shall export the user data with the user data's associated security attributes.
FDP_ETC.2.3	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	The TSF shall enforce the following rules when user data is exported from the TOE: 1) *User keys exported by means of the TPM_CreateWrapKey command shall be exported with the security attrib*utes a) *keyUsage,* b) *keyFlags,* c) *algorithmParms and* d) *PCRInfo with structure identified in KeyInfo if the key is bound to PCRs.* 2) *AIK keys shall be exported with the security attributes* a) *keyUsage,* b) *keyFlags,* c) *algorithmParms and* d) *PCRInfo with structure identified in idKeyParms.* 3) *Migration key blobs shall be exported with the security attributes* a) *keyUsage,* b) *keyFlags,* c) *algorithmParms and* d) *PCRInfo with structure identified in KeyInfo if the key is bound to PCRs.*

4) *Certified migration key blobs shall be exported with the security attributes*
   a) *keyUsage,*
   b) *keyFlags,*
   c) *algorithmParms and*
   d) *PCRInfo with structure TPM_PCR_INFO_LONG.*
5) *Sealed Data shall be exported with the security attributes pcrInfo and tpmProof.*
6) *Context shall be exported with the security attributes resource type and use the tpmProof as secret for the HMAC of the context*.

FDP_ITC.2 Import of User Data with Security Attributes

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	The TSF shall enforce the Key Management SFP, Key Migration SFP, Export and Import of Data SFP when importing user data, controlled under the SFP, from outside of the TOE.
FDP_ITC.2.2	The TSF shall use the security attributes associated with the imported user data.
FDP_ITC.2.3	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	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	The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: 1) *User keys imported by means of the TPM_LoadKey2 command shall be imported with the security attributes contained in Wrapped key blob.*

FDP_UCT.1/Exp Basic Data Exchange Confidentiality

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/Exp	The TSF shall enforce the Key Management SFP, Key Migration *SFP, Export and Import of Data SFP* to be able to *transmit* user data *1) data together with the security attributes pcrInfo of an imported sealed data,* *2) migratable key of an imported Migration Key Blob or Certified Migration Key Blob,* *3) private portion of the key of an imported Wrapped Key Blob,* *4) data of the TPM_CONTEXT_SENSITIVE structure in the exported Context,* in a manner protected from unauthorized disclosure.

FDP_UCT.1/Imp Basic Data Exchange Confidentiality

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/Imp	The TSF shall enforce the Key Management SFP, Key Migration *SFP, Export and Import of Data SFP by providing the ability to receive* user data *1) data together with the security attributes TPM_PCR_INFO in a sealed data object,* *2) migratable key exported in a created or converted Migration Key Blob,* *3) migratable key exported in a created or converted Certified Migration Key Blob,* *4) private portion of the key exported in a Wrapped Key Blob,* *5) data of the TPM_CONTEXT_SENSITIVE structure in the loaded context,* *6) data of the wrapped command within a transport session* in a manner protected from unauthorized disclosure.

FDP_UIT.1/Data Data Exchange Integrity

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/Data	The TSF shall enforce the Key Management SFP, Key Migration SFP, Export and Import of Data SFP to be able to transmit *and receive* user data in a manner protected from *modification, deletion and insertion* errors.
FDP_UIT.1.2/Data	The TSF shall be able to determine on receipt of user data: *1) exported key,* *2) migratable key and its security attributes in a created or converted Migration Key Blob,* *3) migrated migratable key and its security attributes in a Wrapped Key,* *4) certified migratable key and its security attributes in a created or converted Certified Migration Key Blob,* *5) migrated Certified Migratable Key and its security attributes in a Wrapped Key Blob,* *6) saved Context,* whether *modification, deletion and insertion* has occurred.

FDP_UIT.1/Session Data Exchange Integrity

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/Session	The TSF shall enforce *the TPM Mode Control SFP, Delegation SFP, Measurement and Reporting SFP, NVS SFP, Monotonic Counter SFP Key Management SFP, Key Migration SFP, Export and Import of Data SFP* to be able *to transmit and receive:* *1) command input,* *2) return output data and* *3) ordinal, header information and data of the wrapped command in a transport session* in a manner protected from *modification, deletion, insertion and replay* errors.
FDP_UIT.1.2/Session	The TSF shall be able to determine on receipt of user data command *input,* whether *modification, deletion and insertion and replay* has occurred.

FAU_GEN.1 Audit Data Generation

Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable Time Stamps
FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the *not specified* 191 level of audit; and c) *Transport session*.
FAU_GEN.1.2	The TSF shall record within each audit record at least the following information**:** α) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and β) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, χ) *Signed hash value of the TPM_TRANSPORT_LOG_IN structures of the received commands and TPM_TRANSPORT_LOG_OUT structures of the command responses*.

6.1.7 DAA

FDP_ACC.1/DAA Subset Access Control - DAA

Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security Attribute-Based Access Control
FDP_ACC.1.1/DAA	The TSF shall enforce the *DAA SFP* on *1) Subjects: TPM owner* *2) Objects: DAA_tpmSpecific* 3) *Operations: TPM_DAA_Join and TPM_DAA_Sign commands*

FDP_ACF.1/DAA Security Attribute-Based Access Control - DAA

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset Access Control FMT_MSA.3 Static Attribute Initialization
FDP_ACF.1.1/DAA	The TSF shall enforce the *DAA SFP* to objects based on the following: *1) Subjects: TPM owner* 2) *Objects: DAA_tpmSpecific*

FDP_ACF.1.2/DAA	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *1) The TPM owner is allowed to execute the TPM_DAA_Join and TPM_DAA_Sign commands.*
FDP_ACF.1.3/DAA	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *none*.
FDP_ACF.1.4/DAA	The TSF shall explicitly deny access of subjects to objects based on the following additional rule: *1) The TSF shall disallow the TPM_DAA_Sign if the DAA_tpmSpecific is not generated by the same TOE.*

FMT_MSA.1/DAA Management of Security Attributes

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/DAA	The TSF shall enforce the *DAA SFP* to restrict the ability *to modify* the security attributes *DAA parameters* to the *Entity owner*.

FMT_MSA.3/DAA Static Attribute Initialization

Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of Security Attributes
FMT_SMR.1 Security Roles	
FMT_MSA.3.1/DAA	The TSF shall enforce the DAA SFP to provide restrictive default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/DAA	The TSF shall allow the *Entity owner* to specify alternative initial values to override the default values when an object or information is created.

FPR_UNL.1 Unlinkability

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPR_UNL.1.1	The TSF shall ensure that *users* are unable to determine whether *Direct Anonymous Attestation with randomized base name of the verifier is related as follows: performed by the same identity*.

6.1.8 TSF Protection

FPT_FLS.1 Failure with Preservation of Secure State

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: failure of any crypto operations including *RSA encryption, RSA decryption, SHA-1, RNG, RSA signature generation, HMAC generation*; *failure of any commands or internal operations, dictionary attack on authorization, failure of the firmware field upgrade process.*

FPT_TST.1 TSF Testing

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of self tests at the request of an authorized user, at the condition: after each power-on and reset, prior to execution of the first call to a capability that uses those functions to demonstrate the correct operation of the TSF operation of the TSF.
FPT_TST.1.2	The TSF shall provide authorized users with the capability to verify the integrity of TSF data.
FPT_TST.1.3	The TSF shall provide authorized users with the capability to verify the integrity of *TSF.*

Refinement:
After power-on and reset the TOE shall self test all internal functions that are necessary to perform the following operations:

a) *TPM_SHA1Start*
b) *TPM_SHA1Update*
c) *TPM_SHA1Complete*
d) *TPM_SHA1CompleteExtend*
e) *TPM_Extend*
f) *TPM_Startup*
g) *TPM_ContinueSelfTest*
h) *TPM_SelfTestFull*
i) *TPM_HASH_START / TPM_HASH_DATA / TPM_HASH_END*
j) *TPM_NV_ReadValue for indices with the attributes TPM_NV_PER_AUTHREAD and TPM_NV_PER_OWNERREAD set to FALSE*
k) *TSC_ORD_PhysicalPresence*
l) *TSC_ORD_ResetEstablishmentBit*
m) *TPM_GetCapability with the property TPM_CAP_PROPERTY, subcap property TPM_CAP_PROP_TIS_TIMEOUT*

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.

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.

This assurance package is the assurance package of the claimed Protection Profile ([PP]).

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 four functions within the TPM related to cryptographic operations:

- RSA digital signature generation and verification
- RSA encryption and decryption and the generation of hash
- AES encryption/decryption
- HMAC values

The AES encryption/decryption module operates with conformance to FIPS-197 with a key size of 128 bits using counter mode encryption/decryption.

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 functions.

7.1.3 SF3 – Access Control

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

The TOE enforces the POAC policy on subjects (commands), objects (keys and user data) and operations (signature generation/verification, encryption or decryption). The TOE provides access control by denying access to some subjects, objects and operations, and allowing access to other subjects, objects and operations, based on three different security attributes, which are stored as flags in the TPM or associated with the data in an encrypted blob.

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 monolithic package. Any intent to gain physical access to the TPM protected areas will result in evident 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

There are three functions within the TPM related to key management:

- Generation of asymmetric key pairs
- Key storing
- Key destruction

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 512, 1024 and 2048 bits as defined by PKCS#1 V2.0. The TOE supports storing of cryptographic keys by storing them in a randomized location inside the shielded location.

The TPM supports destruction of cryptographic keys by invalidating the keys in accordance with FIPS 140-2.

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 FIPS-186-2.

7.1.7 SF7 – Identification and Authentication

The TOE 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-x calls the identification and authentication process and this data authorization. In both cases, the protocol exchanges nonce-data so that both sides of the transaction can compute a HMAC

using secrets or shared secrets and nonce-data. Each side computes the hash value and can compare to the value transmitted. Network listeners cannot infer directly the authorization data from the hashed objects sent over the network. The identification and authentication data for the TOE owner and the owner of the Storage Root Key are held within the TOE itself. The identification and authentication data for other owners of entities are held and protected with the entity.

The TPM provides two protocols for authentication and identification, to authenticate an entity owner and to authorize use of an entity without revealing the authorization data on the network or the connection to the TOE. The first protocol is the "*Object-Independent Authorization Protocol*" (OIAP), which allows the exchange of nonces with a specific TPM. The second protocol is the "*Object-Specific Authorization Protocol*" (OSAP)", which allows establishment of an authentication session for a single entity. Both identification and authentication protocols use a random nonce. This requires that a nonce from one side be in use only for a message and its reply. For instance, the TOE would create a nonce and send that on the reply. The requestor would receive that nonce and then include it in the next request. The TOE would validate that the correct nonce was in the request and then create a new nonce for the reply. This mechanism is in place to prevent replay attacks and man-in-the-middle attacks.

The TOE prevents the reuse of authentication related to authorization data by using *nonces* for each message and response for all authorization protocols. The *nonce* values from the TOE use the internal RNG. Re-authentication of users is done by using the authorization protocol with a new *nonce* for each message and response.

Any operational role can access all protected commands and shielded locations only through the authentication mechanism. The access-right of commands, user data, keys and operations are defined by different security attributes as defined in Chapter 7.1.3. The TPM allows access to data and keys with "world" access and access to different commands on behalf of the user to be performed before the user is authenticated/identified. In contrast to this, each user must be authenticated/identified successfully before allowing any other TSF-mediated action on behalf of that user.

Furthermore, the SF7 supplies the generation and verification of evidence of origin for transmitted data-signed using identity keys, by using RSA algorithm for the signature operation at all times.

7.1.8 SF8 – Firmware Field Upgrade

Field upgrading of the TPM firmware is managed securely in the following way:

The Field Upgrade process does not expose the FW as plain text. This is achieved by using AES algorithm (CTR mode) and Nuvoton's Field Upgrade Symmetric AES 128-bit Key.

The Field Upgrade process uses authentication to verify the integrity and source of the FW. This is achieved by using RSA signature scheme TPM_SS_RSASSAPKCS1v15_SHA, HASH algorithms and Nuvoton's Field Upgrade RSA 2048-bit Key.

If the field upgrade process succeeds, then the resulting product is the Final TOE; otherwise (in case of interruption or an incident that prevents the formation of the Final TOE), the Initial TOE remains either in its initial or failed.

The TOE has a dedicated TPM command that reports the version of the TOE firmware.

7.1.9 Assignment of SFs to Security Functional Requirements

The justification of the mapping between security functional requirements and security functionalities is shown in Table 4:

Table 4. – Assignment of Security Functional Requirements to Security Functions

#	SFR	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8
1	FMT_SMR.1							X	X
2	FMT_SMF.1			X		X		X	
3	FMT_MSA.2			X				X	X
4	FPT_TDC.1			X					
5	FCS_CKM.1					X			
6	FCS_RNG.1						X		
7	FCS_CKM.4					X			
8	FCS_COP.1	X							X
9	FDP_ACC.1/Modes			X					X
10	FDP_ACF.1/Modes			X					X
	FDP_UIT.1								X
	FDP_UCT.1								X
11	FMT_MSA.1/Modes			X					
12	FMT_MSA.1/PhysP			X					
13	FMT_MTD.1/AuthData							X	
14	FMT_MTD.1/Deleg							X	
15	FIA_UID.1							X	
16	FIA_UAU.1							X	
17	FIA_UAU.4							X	
18	FIA_UAU.5							X	
19	FIA_UAU.6							X	
20	FIA_AFL.1							X	
21	FMT_MTD.1/Lock							X	
22	FIA_USB.1							X	
23	FDP_RIP.1			X					
24	FDP_ACC.1/Deleg			X					
25	FDP_ACF.1/Deleg			X					
26	FMT_MSA.1/DFT			X					
27	FMT_MSA.1/DT			X					
28	FMT_MSA.3/Deleg			X					
29	FDP_ACC.1/KeyMan			X					
30	FDP_ACF.1/KeyMan			X					

#	SFR	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8
31	FMT_MSA.1/KeyMan			X					
32	FMT_MSA.1/Kevi			X					
33	FMT_MSA.3/KeyMan			X					
34	FDP_ACC.1/MigK			X					
35	FDP_ACF.1/MigK			X					
36	FMT_MSA.1/MigK			X					
37	FMT_MTD.1/MigK			X					
38	FDP_ACC.1/M&R			X					
39	FDP_ACF.1/M&R			X					
40	FMT_MSA.3/M&R			X					
41	FCO_NRO.1/M&R			X					
42	FDP_ACC.1/NVS			X					
43	FDP_ACF.1/NVS			X					
44	FMT_MSA.3/NVS			X					
45	FDP_ACC.1/MC			X					
46	FDP_ACF.1/MC			X					
47	FMT_MSA.1/MC			X					
48	FMT_MSA.3/MC			X					
49	FPT_STM.1							X	
50	FCO_NRO.1/STS							X	
51	FDP_ACC.1/EID			X					
52	FDP_ACF.1/EID			X					
53	FMT_MSA.3/EID			X					
54	FDP_ETC.2			X					
55	FDP_ITC.2			X					
56	FDP_UCT.1/Exp			X					
57	FDP_UCT.1/Imp			X					
58	FDP_UIT.1/Data			X					
59	FDP_UIT.1/Session			X					
60	FAU_GEN.1			X				X	
61	FDP_ACC.1/DAA			X					
62	FDP_ACF.1/DAA			X					
63	FMT_MSA.1/DAA			X					
64	FMT_MSA.3/DAA			X					
65	FPR_UNL.1			X					
66	FPT_FLS.1	X	X	X					X
67	FPT_TST.1		X						
68	FPT_PHP.3				X				

8 Appendix

8.1 TCG Specification Commands Implemented in the TOE

The following table lists all commands available on the TOE. It shows which of the commands labeled as “optional” in the TCG specification are implemented in the TOE and which of them are Nuvoton-specific commands.

Table 5. – Commands Implemented in the TOE

Command					Avail	
	Optional	RQU	No Owner	Deactivated	Disabled
TPM_ORD_ActivateIdentity 122 0x0000007A					
TPM_ORD_AuthorizeMigrationKey 43 0x0000002B					
TPM_ORD_CertifyKey 50 0x00000032		X			
TPM_ORD_CertifyKey2 51 0x00000033		X			
TPM_ORD_ChangeAuth 12 0x0000000C					
TPM_ORD_ChangeAuthAsymFinish 15 0x0000000F		X			
TPM_ORD_ChangeAuthAsymStart 14 0x0000000E		X			
TPM_ORD_ChangeAuthOwner 16 0x00000010					
TPM_ORD_CMK_ApproveMA 29 0x0000001D	X				
TPM_ORD_CMK_ConvertMigration 36 0x00000024	X				
TPM_ORD_CMK_CreateBlob 27 0x0000001B	X				
TPM_ORD_CMK_CreateKey 19 0x00000013	X				
TPM_ORD_CMK_CreateTicket 18 0x00000012	X				
TPM_ORD_CMK_SetRestrictions 28 0x0000001C	X				
TPM_ORD_ContinueSelfTest 83 0x00000053		X	X	X	X
TPM_ORD_ConvertMigrationBlob 42 0x0000002A		X			
TPM_ORD_CreateCounter 220 0x000000DC					
TPM_ORD_CreateEndorsementKeyPair 120 0x00000078		X	X		
TPM_ORD_CreateMigrationBlob 40 0x00000028					
TPM_ORD_CreateWrapKey 31 0x0000001F					
TPM_ORD_DAA_Join 41 0x00000029	X				
TPM_ORD_DAA_Sign 49 0x00000031	X				
TPM_ORD_Delegate_CreateKeyDelegation 212 0x000000D4					
TPM_ORD_Delegate_CreateOwnerDelegation 213 0x000000D5					
TPM_ORD_Delegate_LoadOwnerDelegation 216 0x000000D8		X	X		
TPM_ORD_Delegate_Manage 210 0x000000D2		X	X		
TPM_ORD_Delegate_ReadTable 219 0x000000DB		X	X		
TPM_ORD_Delegate_UpdateVerification 209 0x000000D1					
TPM_ORD_Delegate_VerifyDelegation 214 0x000000D6		X			
TPM_ORD_DirRead 26 0x0000001A		X			
TPM_ORD_DirWriteAuth 25 0x00000019					
TPM_ORD_DisableForceClear 94 0x0000005E		X	X		
TPM_ORD_DisableOwnerClear 92 0x0000005C					

Command			Optional	RQU	No Owner	Avail	
						Deactivated	Disabled
TPM_ORD_DisablePubekRead	126	0x0000007E					
TPM_ORD_DSAP	17	0x00000011		X		X	X
TPM_ORD_EstablishTransport	230	0x000000E6		X			
TPM_ORD_EvictKey	34	0x00000022		X			
TPM_ORD_ExecuteTransport	231	0x000000E7					
TPM_ORD_Extend	20	0x00000014		X	X	X	X
TPM_ORD_FieldUpgrade	170	0x000000AA	X	X	X	X	X
TPM_ORD_FlushSpecific	186	0x000000BA		X	X	X	X
TPM_ORD_ForceClear	93	0x0000005D		X	X		
TPM_ORD_GetCapability	101	0x00000065		X	X	X	X
TPM_ORD_GetCapabilityOwner	102	0x00000066					
TPM_ORD_GetPubKey	33	0x00000021		X			
TPM_ORD_GetRandom	70	0x00000046		X	X		
TPM_ORD_GetTestResult	84	0x00000054		X	X	X	X
TPM_ORD_GetTicks	241	0x000000F1		X	X		
TPM_ORD_IncrementCounter	221	0x000000DD				X	X
TPM_ORD_Init	151	0x00000097		X	X	X	X
TPM_ORD_KeyControlOwner	35	0x00000023					
TPM_ORD_LoadContext	185	0x000000B9		X			
TPM_ORD_LoadKey	32	0x00000020		X			
TPM_ORD_LoadKey2	65	0x00000041		X			
TPM_ORD_MakeIdentity	121	0x00000079					
TPM_ORD_MigrateKey	37	0x00000025		X			
TPM_ORD_NV_DefineSpace	204	0x000000CC		X	X	A	A
TPM_ORD_NV_ReadValue	207	0x000000CF		X	X	A	A
TPM_ORD_NV_ReadValueAuth	208	0x000000D0					
TPM_ORD_NV_WriteValue	205	0x000000CD		X	X	A	A
TPM_ORD_NV_WriteValueAuth	206	0x000000CE					
TPM_ORD_OIAP	10	0x0000000A		X	X	X	X
TPM_ORD_OSAP	11	0x0000000B		X		X	X
TPM_ORD_OwnerClear	91	0x0000005B					
TPM_ORD_OwnerReadInternalPub	129	0x00000081					
TPM_ORD_OwnerReadPubek	125	0x0000007D					
TPM_ORD_OwnerSetDisable	110	0x0000006E				X	X
TPM_ORD_PCR_Reset	200	0x000000C8		X	X		
TPM_ORD_PcrRead	21	0x00000015		X	X		
TPM_ORD_PhysicalDisable	112	0x00000070		X	X	X	
TPM_ORD_PhysicalEnable	111	0x0000006F		X	X	X	X
TPM_ORD_PhysicalSetDeactivated	114	0x00000072		X	X	X	
TPM_ORD_Quote	22	0x00000016		X			
TPM_ORD_Quote2	62	0x0000003E	X	X			
TPM_ORD_ReadCounter	222	0x000000DE		X	X		

Command			Optional	RQU	No Owner	Avail	
						Deactivated	Disabled
TPM_ORD_ReadPubek	124	0x0000007C		X	X		
TPM_ORD_ReleaseCounter	223	0x000000DF			X		
TPM_ORD_ReleaseCounterOwner	224	0x000000E0					
TPM_ORD_ReleaseTransportsigned	232	0x000000E8					
TPM_ORD_Reset	90	0x0000005A		X	X	X	X
TPM_ORD_ResetLockValue	64	0x00000040					
TPM_ORD_SaveContext	184	0x000000B8		X			
TPM_ORD_SaveState	152	0x00000098		X	X	X	X
TPM_ORD_Seal	23	0x00000017					
TPM_ORD_Sealx	61	0x0000003D	X				
TPM_ORD_SelfTestFull	80	0x00000050		X	X	X	X
TPM_ORD_SetCapability	63	0x0000003F		X	X	X	X
TPM_ORD_SetOperatorAuth	116	0x00000074		X	X		
TPM_ORD_SetOwnerInstall	113	0x00000071		X	X		
TPM_ORD_SetOwnerPointer	117	0x00000075		X			
TPM_ORD_SetTempDeactivated	115	0x00000073		X	X		X
TPM_ORD_SHA1Complete	162	0x000000A2		X	X	X	X
TPM_ORD_SHA1CompleteExtend	163	0x000000A3		X	X	X	X
TPM_ORD_SHA1Start	160	0x000000A0		X	X	X	X
TPM_ORD_SHA1Update	161	0x000000A1		X	X	X	X
TPM_ORD_Sign	60	0x0000003C		X			
TPM_ORD_Startup	153	0x00000099		X	X	X	X
TPM_ORD_StirRandom	71	0x00000047		X	X		
TPM_ORD_TakeOwnership	13	0x0000000D			X	X	
TPM_ORD_Terminate_Handle	150	0x00000096		X	X	X	X
TPM_ORD_TickStampBlob	242	0x000000F2		X			
TPM_ORD_UnBind	30	0x0000001E		X			
TPM_ORD_Unseal	24	0x00000018					
TSC_ORD_PhysicalPresence	10	0x4000000A		X	X	X	X
TSC_ORD_ResetEstablishmentBit	11	0x4000000B		X	X	X	X

8.2 References

Nuvoton TPM

[AGD] NPCT6xx TPM1.2 Programmer's Guide, August 2015, Rev 1.2

[Datasheet] NPCT6xx Trusted Platform Module Version 1.2 (TPM1.2), January 2016, Revision 1.1

Common Criteria

[CC] Common Criteria for Information Technology Security Evaluation, version 3.1, revision 4, September 2012
Part 1: Introduction and general model, CCMB-2012-09-001,
Part 2: Security functional requirements, CCMB-2012-09-002,
Part 3: Security Assurance Requirements, CCMB-2012-09-003

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, version 3.1, revision 4, September 2012, CCMB-2012-09_004

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

Protection Profile

[PP] Trusted Computing Group Protection Profile PC Client-Specific Trusted Platform Module, TPM Family 1.2; Level 2 Revision 116, version 1.2
http://www.commoncriteriaportal.org/files/ppfiles/pp0030_ma1a_pdf.pdf

TCG

[TCG-1] TPM Main Part 1 Design Principles, Specification version 1.2, revision 116 (1 March, 2011)

[TCG-2] TPM Main Part 2 TPM Structures, Specification version 1.2, revision 116 (1 March, 2011)

[TCG-3] TPM Main Part 3 Commands, Specification version 1.2, revision 116 (1 March, 2011)

[TCG_PC] TCG PC Client-Specific TPM Interface Specification (TIS) Specification Version 1.3 (21 March 2013)
https://www.trustedcomputinggroup.org/home

Specifications

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

[FIPS 180] FIPS PUB 180-2 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, SECURE HASH STANDARD, National Institute of Standards and Technology, 2002 August 1

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

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

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

8.3 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

8.4 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-x 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 on 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 that is common to all types of platform and 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 that is common to all types of platform and that is not required to be trustworthy (and therefore does not need to be part of the TPM).

User:	An entity that uses the platform on which 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, which 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.

Headquarters

No. 4, Creation Rd. 3
Science-Based Industrial Park
Hsinchu, Taiwan, R.O.C
TEL: 886-3-5770066
FAX: 886-3-5665577
http://www.nuvoton.com.tw (Ch.)
http://www.nuvoton.com (Eng.)

Nuvoton Technology Corporation America

2727 North First Street
San Jose, CA 95134, U.S.A.
TEL: 1-408-9436666
FAX: 1-408-5441798

Nuvoton Technology (Shanghai) Ltd.

27F, 2299 Yan An W. Rd.
Shanghai, 200336 China
TEL: 86-21-62365999
FAX: 86-21-62365998

Taipei Office

1F, No.192, Jingye 1st Rd
Zhongshan District, Taipei, 104
Taiwan, R.O.C.
TEL: 886-2-2658-8066
FAX: 886-2-8751-3579

Winbond Electronics Corporation Japan

NO. 2 Ueno-Bldg., 7-18, 3-chome
Shinyokohama Kohoku-ku
Yokohama, 222-0033
TEL: 81-45-4781881
FAX: 81-45-4781800

Nuvoton Technology (H.K.) Ltd.

Unit 9-15, 22F, Millennium City 2
378 Kwun Tong Rd
Kowloon, Hong Kong
TEL: 852-27513100
FAX: 852-27552064

For Advanced PC Product Line information contact: APC.Support@nuvoton.com